RC 21039 (94204) (1/28/97)
Computer Science/Mathematics

IBM Research Report

An Efficient Recursive Partitioning Algorithm For Classification, Using Wavelets

Vittorio Castelli

IBM Research Division
T.J. Watson Research Center
Yorktown Heights, New York 10598

Ioannis Kontoyiannis

Information Systems Laboratory
Stanford University
Stanford, CA 94305

An Efficient Recursive Partitioning Algorithm For Classification, Using Wavelets [1]

Vittorio Castelli [2] and Ioannis Kontoyiannis
IBM T.J. Watson Research Center, USA *Stanford University, USA*

Abstract

Summary

We describe and analyse 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 traditional sample-by-sample algorithms, progressive classification is computationally more efficient and at the same time, under certain conditions, more accurate.

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

Keywords: Parametric, Nonparametric Classification; Wavelet Transform; Multiresolution; Image Analysis

1. Introduction

We describe and analyse a new methodology for performing efficient classification on large two-dimensional data sets ("images"). We apply a dyadic, recursive partitioning algorithm to generate a tree-structured partition, with a rule for partitioning 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

[1]This work was funded in part by grant no. NASA/CAN NCC5-101.

an image with known pixel-values but unknown labels and assigning a label to each pixel.

The following example motivates our subsequent analysis. Consider a large digital library of satellite images of the earth's surface. A basic requirement from such a system is that it supports content-based search through the images, and classification is the main operator for such searches. The traditional approach (see, for example, Chettri *et al.* (1992), Richards (1993), Chettri and Cromp (1993), Salu and Tilton (1993), Paola and Schowengerdt (1995), and the references therein) is to examine the pixels one-by-one and assign a label to each pixel independently of the labels of the neighbouring pixels. In practice this approach suffers two defects: It is computationally very expensive, and – other than through smoothing – it does not take into account the strong correlation between the labels of neighbouring pixels that seems to occur in natural images.

We present a framework called *progressive classification* that allows us to classify transformed data quickly and accurately. The main gist of this approach is to look at a lower-resolution representation of the data, where each pixel corresponds to a $k \times k$ pixel block from the original image. 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 $k \times k$ pixel block is now represented by four pixels instead of just one, and examines each one of those four 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.[3] 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 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, nearest neighbour, learning vector quantisation, maximum likelihood and neural network classifiers.)

This methodology was first implemented by Castelli *et al.* (1996), where practical issues arising in the implementation were discussed, and extensive numerical results

[3] 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 (1995), and the special issue Daubechies *et al.* (1992) 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.

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 the subsequent analysis (presented in Sections 2 and 3) we conclude that, first, progressive classification achieves a significant speedup over traditional pixel-by-pixel classification methods, and, second, that for images with label values that are highly correlated, the progressive classifier will give more accurate results than the corresponding non-progressive classifier. The reason for the speed-up is clearly that, 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 much smaller than MN. The reason for the improved 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 our 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 (1995)). A work closer in spirit to the present paper is Donoho (1997), 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 the error rate of the classifier as well as in the classification complexity, and the partition regions produced by our algorithms are generated by a spatially adaptive blocking technique, and they depend on the choice of the wavelet basis and also on the choice of the underlying classifier.

In section 2, after some preliminaries on classification and wavelets we define the progressive framework and we present the statistical analysis of progressive classifiers applied to one-dimensional data. Section 3 contains the corresponding analysis for two-dimensional models, and section 4.3 briefly shows how the progressive classification framework can be applied to general lattice data in dimensions higher than one or two. In section 4.1 we discuss the tradeoff between classification complexity and accuracy, and in section 4.2 we discuss the choice of the wavelet basis. Appendix A contains the proofs of the results in section 2.

2. Classification in One Dimension

2.1 *Preliminaries*

First we give a brief description of the DWT; Daubechies (1992) 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 filters $\mathbf{F}$ and $\mathbf{G}$ where by a "filter" we mean a (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 (1992), Rioul and Vetterli (1991), and Rioul (1993). 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 the sake of simplicity the analysis will be based on using Haar wavelets, namely the DWT will be specified by the filters $\mathbf{G} = (1/\sqrt{2}, 1/\sqrt{2})$ and $\mathbf{F} = (1/\sqrt{2}, -1/\sqrt{2})$. In this case, the level-ℓ wavelet coefficients $\{w_i^\ell\}$ will be scaled averages of distinct subsequences from $\mathbf{x}$:

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

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 also often referred to as the "scaling function coefficients" or the "approximation coefficients.")

Next a few words about classification. Let $\mathbf{x} = \{x_i\}$ be the collection of data to be classified. The observations x_i take values in $\mathbf{R}$ or $\mathbf{R}^d$, and with each x_i there is associated a label $\theta_i \in \{0, 1\}$ denoting its class. A pair (x_i, θ_i) is called a labeled sample. Assume that the observations and labels are generated by $(\mathbf{X}, \mathbf{\Theta}) = \{(X_i, \Theta_i)\}$ where $\mathbf{\Theta}$ is a stationary process, and that, given the labels θ_i, the observations X_i are conditionally independent and distributed according to the densities f_{θ_i}, where f_0 and f_1 are some fixed densities. The problem of classification consists of observing the values $\{x_i\}$ and assigning labels $\{\hat{\theta}_i\}$ to them, in such a way as to minimise the probability of incorrect classification, or *error rate*, $\Pr\{\hat{\theta}_i \neq \theta_i\}$. 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.*

We denote random variables by upper-case letters, and their realisations by the corresponding lower-case letters, so that, for example, taking the Haar transform (1) of the sequence $\{x_i\}$ up ℓ levels produces a sequence of wavelet coefficients $\{w_i^\ell\}$ that are distributed according to the random variables $W_i^\ell = 2^{\ell/2} \sum_{j=0}^{2^\ell-1} X_{2^\ell i+j}$.

2.2 *The Progressive Classifier*

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

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

(i) B, the *baseline classifier*, is a classification rule that operates on the full-resolution data, and it is the product classifier characterised by a fixed partition $\Pi^0 = \{\Pi^0_0, \Pi^0_1\}$ of the observation space.
(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 regions $\Pi^\ell_{\rm prog}, \Pi^\ell_0, \Pi^\ell_1$.
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 $\Pi^L_{\rm prog}$ 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 analyses the corresponding coefficients at level $(L-1)$ independently, using the partition Π^{L-1}. The same process is repeated iteratively.

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 4.2, the progressive classification framework 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.

1. ML: The baseline classifier is the product maximum likelihood classifier.
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. Gaussian: The distributions of samples of classes 0 and 1 are Gaussian with unit variance and means $+1$ and -1, respectively.

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 (θ_1, θ_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 (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 $\phi(x)$ 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; 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 $-T \leq w_i^1 \leq T$ 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.
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 $p < p_T$ the error rate of the level-1 progressive classifier is smaller than the error rate of the baseline classifier.

Next we drop the Gaussian assumption, and extend Theorem 1 to general (not necessarily maximum likelihood) level-1 progressive classifiers. We partition the observation space (x_1, x_2) into disjoint regions $A_{i,j,k}$ and A_0, for $i, j, k \in \{0, 1\}$, as follows. The point (x_1, x_2) belongs to region $A_{i,j,k}$ if the baseline classifier decides $\hat{\theta}_1 = i, \hat{\theta}_2 = j$ and the level-1 progressive classifier decides $\hat{\theta}_1 = \hat{\theta}_2 = k$. The point (x_1, x_2) belongs to region A_0 if the progressive classifier takes the progressive step. Write $\bar{i} = 1 - i$.

Theorem 2. Under the Markov assumption, if

$$P_1 = \sum_{m=0,1} \sum_{i,j=0,1} \left\{ (1 - 2\mathbb{I}_i(m))\mu_{m,m}(A_{i,j,\bar{i}}) + (1 - 2\mathbb{I}_j(m))\mu_{m,m}(A_{i,j,\bar{j}}) \right\} > 0, \qquad (2)$$

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

Although the assumption $P_1 > 0$ may at first seem like 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) to change P_1 to $-P_1$).
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 $\Pr\{W_1^1 \in \Pi^1_{\text{prog}}\} < 1/2$.

For example, let assumptions *1*, *2* and *3* hold, consider the ML 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 generalised as follows.

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

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

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

Note that Theorem 2 and Corollary 1 together imply that when the label process is highly correlated, namely whenever the transition probability p is small enough, then a well-designed level-1 progressive classifier will be both faster and more accurate than the corresponding baseline classifier.

2.4 *Results for 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 A.
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 $W_0^{\ell-1}$ is

$$f_{W_0^{\ell-1}}(w \mid \theta_0, \ldots, \theta_{2^{\ell-1}-1}) = f_{W_0^{\ell-1}}(w \mid S_0^{\ell-1} = k) = \phi\left(w + 2^{-(\ell-1)/2}(2^{\ell-1} - 2k)\right),$$

and its unconditional density is given by the mixture

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

where the mixing coefficients $\Pr\{S_0^{\ell-1} = k\}$ can be calculated from the model for the label process $\boldsymbol{\Theta}$. 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

$$\sum_{i=0}^{2^\ell-1} \sum_{j=0}^{2^\ell-1} \Pr\{S_0^{\ell-1} = i, S_1^{\ell-1} = j\} f_{W_0^{\ell-1}}(w_0 \mid S_0^{\ell-1} = i) f_{W_1^{\ell-1}}(w_1 \mid S_1^{\ell-1} = j). \tag{3}$$

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 (1996)). Figure 2 shows an example of the mixture density (3) and its components for level $\ell = 4$ under the Gaussian assumption, for the Markov model with $p = 0.08$.

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 $W_0^\ell \in \Pi_{\text{prog}}$ then the level-ℓ 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 $\{A_{i,j,k}\}$, $k = 0, 1$, $i, j \in \{0, 1, \text{prog}\}$. 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. Figure 3 shows an example of the partition of the plane for level-2 and level-3 progressive ML classifiers, for the Markov model.

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

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

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:

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

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

Observe that the condition in Theorem 5 is no longer necessary. In the Gaussian case with an ML classifier and a Markov model, Theorem 5 simplifies to:

Corollary 2. Suppose that assumptions *1*, *2* and *3* hold. 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 row-by-row and the resulting matrix is filtered column-by-column. The result is arranged as a matrix composed of four blocks (subbands), as shown in Figure 4. 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 $\{x_i, \theta_i\}$ generated by the stationary random field $(\mathbf{X}, \mathbf{\Theta}) = \{(X_u, \Theta_u) \ ; \ u \in \mathbf{Z}^2\}$, 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 X_u are conditionally independent, and distributed according to the fixed densities f_0 and f_1. 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, and the Markovian assumption will not be explicitly used in the derivations of our results.
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 generalise 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 let $\mu_{i,j,h,k}^\ell(w_{0,0}, w_{0,1}, w_{1,0}, w_{1,1})$ be the induced conditional measure. 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 denoted by $\pi^\ell(i) = \Pr\{S_u^\ell = i\}$ 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 $\Pi_0^{\ell}, \Pi_1^{\ell}, \Pi_{\text{prog}}^{\ell}$ of the level-ℓ classifier satisfies

$$\sum_{i,j,h,k} \sum_{n=1}^{4^{\ell}-1} \sum_{\substack{\{n_1,n_2,n_3,n_4\} \\ \Sigma n_i = n}} \pi^{\ell}(n_1,n_2,n_3,n_4) \Big[\mu_{n_1,n_2,n_3,n_4}(A_{i,j,h,k,1}) \Big\{ e_{i,j,h,k,1} + \sum n_i - 4^{\ell} \Big\}$$
$$+ \mu_{n_1,n_2,n_3,n_4}(A_{i,j,h,k,0}) \Big\{ e_{i,j,h,k,0} - \sum n_i \Big\} \Big] > 0,$$

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 $\pi^{\ell}(i,j,h,k)$. These can be calculated numerically if, for example, we assume that $\mathbf{\Theta}$ is a stationary Markov random field, and use Monte Carlo simulation to estimate the relevant model parameters. In Castelli and Kontoyiannis (1996) a a simpler, 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 specialised to any set of additional assumptions. Under the Gaussian assumption, for instance, if the progressive classifier is defined by a threshold (as is the case for maximum likelihood classifiers), 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 $T^{\text{opt}} < \infty$ 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 $\Pr\{W_u^{\ell} \in \Pi_{\text{prog}}^{\ell}\} < 3/4$ 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 (1995) 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, minimising 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 5 shows a simulation example for J.
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 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 *General Wavelet Bases*

Although the Haar wavelet basis was used to simplify the theoretical analysis, in practice, the progressive classification framework can be applied using any orthonormal or biorthogonal family of wavelets, with only minor modifications. As reported in Castelli *et al.* (1996), 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 since 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. Although they may not always provide computationally simple ways to define the optimum decision regions in the parametric case, they justify the use of the progressive approach in the most relevant practical cases, namely in conjunction with nonparametric classifiers. In this case they can also provide guidelines on how to evaluate numerically the partitions produced by nonparametric methods.

4.3 *Higher-Dimensional Data*

The progressive methodology, 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.

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.

Appendix A

A.1. *Proof of Theorem 1*

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

$$
\begin{aligned}
R_{\rm prog} &= 4/2\,\{0 \cdot \pi(1,1)\mu_{1,1}(A) + 1 \cdot \pi(1,0)\mu_{1,0}(A) \\
&\qquad + 1 \cdot \pi(0,1)\mu_{0,1}(A) + 2 \cdot \pi(0,0)\mu_{0,0}(A)\} \\
R_{\rm bas} &= 2\,\{1 \cdot \pi(1,1)\mu_{1,1}(A) + 2 \cdot \pi(1,0)\mu_{1,0}(A)
\end{aligned}
$$

$$+ 0 \cdot \pi(0,1)\mu_{0,1}(A) + 1 \cdot \pi(0,0)\mu_{0,0}(A)\},$$

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. □

A.2. *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

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

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

$$\begin{aligned}
\Delta R &= \frac{1}{2}\sum_{m,n}\sum_{i,j}\sum_{k}\left(\mathbb{I}_k(m) + \mathbb{I}_k(n) - \mathbb{I}_i(m) - \mathbb{I}_j(n)\right)\pi(m,n)\mu_{m,n}(A_{i,j,k}) \\
&= \frac{1}{2}\sum_{m,n}\pi(m,n)\left[\sum_i\left\{(1 - 2\mathbb{I}_i(m))\sum_j \mu_{m,n}(A_{i,j,\bar{i}})\right\}\right. \\
&\qquad \left. + \sum_j\left\{(1 - 2\mathbb{I}_j(n))\sum_i \mu_{m,n}(A_{i,j,\bar{j}})\right\}\right] \\
&= \frac{1-p}{4}\sum_m\left[\sum_i\left\{(1 - 2\mathbb{I}_i(m))\sum_j \mu_{m,m}(A_{i,j,\bar{i}})\right\}\right. \\
&+ \qquad \left. \sum_j\left\{(1 - 2\mathbb{I}_j(m))\sum_i \mu_{m,m}(A_{i,j,\bar{j}})\right\}\right]
\end{aligned}$$

$$
\begin{aligned}
&-\frac{p}{4}\sum_m\left[\sum_i\left\{(2\mathbb{I}_i(m)-1)\sum_j \mu_{m,\overline{m}}(A_{i,j,\overline{i}})\right\}\right.\\
&\qquad\left.+\sum_j\left\{(2\mathbb{I}_j(\overline{m})-1)\sum_i \mu_{m,\overline{m}}(A_{i,j,\overline{j}})\right\}\right]\\
\triangleq\ & \frac{1-p}{4}P_1-\frac{p}{4}P_2.
\end{aligned}
$$

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. □

A.3. *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

$$
\begin{aligned}
&(1-\Pr\{W_1^1\in\Pi^1_{\text{prog}}\}+3\Pr\{W_1^1\in\Pi^1_{\text{prog}}\})\\
&\qquad\qquad < \quad 2(1-\Pr\{W_1^1\in\Pi^1_{\text{prog}}\}+2\Pr\{W_1^1\in\Pi^1_{\text{prog}}\}),
\end{aligned}
$$

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

A.4. *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

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

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

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

A.5. *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:

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

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

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

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

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

and the theorem follows. □

A.6. *Proof of Theorem 5*

The proof is similar to the proof of Theorem 3. If $W_1^\ell \notin \Pi^\ell_{\text{prog}}$, 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 $W_1^\ell \in \Pi^\ell_{\text{prog}}$, 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 $(1 - \Pr\{\Pi^\ell_{\text{prog}}\}) + (3 + n_1)\Pr\{\Pi^\ell_{\text{prog}}\})$ is smaller than $(2 + n_2)(1 - \Pr\{\Pi^\ell_{\text{prog}}\}) + (2 + n_1)\Pr\{\Pi^\ell_{\text{prog}}\})$, or, equivalently, if

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

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

A.7. *Proof of Corollary 2*

We just give a sketch of the proof. Note that $\mu_{2^{\ell}/2,2^{\ell}/2}(\Pi_{\text{prog}}) \geq \mu_{i,2^n-j}(\Pi_{\text{prog}})$ for each i, 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

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

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

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

References

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

Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984) *Classification and Regression Trees*, Belmont, CA: Wadsworth & Brooks/Cole.

Castelli, V. and Kontoyiannis, I. (1996) Progressive classification: a multiresolution approach. IBM Research Report RC-20475, June 1996.

Castelli, V., Li, C.-S., Turek, J. J. and Kontoyiannis, I. (1996) Progressive classification in the compressed domain for large EOS satellite databases. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP '96)*, vol. IV, pp. 2199–2202.

Chettri, S. R. and Cromp, R. F. (1993) Probabilistic neural network architecture for high-speed classification of remotely sensed imagery. *Telematics and Informatics*, **10**, 187–198.

Chettri, S. R., Cromp, R. F. and Birmingham, M. (1992) Design of neural networks for classification of remotely sensed imagery. *Telematics and Informatics*, **9**, 145–156.

Daubechies, I. (1992) *Ten Lectures on Wavelets*, Philadelphia, PA: Society for Industrial and Applied Mathematics.

Daubechies, I., Mallat, S. and Willsky, A. S. (editors) (1992) *Special Issue on Wavelet Transforms, IEEE Trans. Inform. Theory*, **38**, no. 2, Part II.

Donoho, D. L. (1997) CART and Best-Ortho-Basis: a connection. *Ann. Stat.*, **25**, 1870–1911.

Guyon, X. (1995) *Random Fields on a Network*, New York: Springer-Verlag.

Richards, J. A. (1993) *Remote Sensing Digital Image Analysis.* Berlin: Springer Verlag.

Rioul, O. (1993) Discrete-time multiresolution theory. *IEEE Trans. Signal Processing*, **41**, 2591–2606.

Rioul, O. and Vetterli, M. (1991) Wavelets and signal processing. *IEEE Signal Processing Mag.*, **8**, 14–38.

Tate, R., Watson, D. and Eglen, S. (1995) Using wavelets for classifying human in vivo magnetic resonance spectra. In *Wavelets and Statistics* (eds. A. Antoniadis and G. Oppenheim), pp. 377–383. New York: Springer-Verlag Lecture Notes in Statistics.

Paola, J. D. and Schowengerdt, R. A. (1995) A detailed comparison of backpropagation neural networks and maximum-likelihood classifiers for urban land use classification. *IEEE Trans. Geosci. Remote Sensing*, **33**, 981–996.

Salu, Y. and Tilton, J. (1993) Classification of multispectral image data by the binary diamond neural network and by nonparametric, pixel-by-pixel methods. *IEEE Trans. Geosci. Remote Sensing*, **31**, 606–617.

Figures

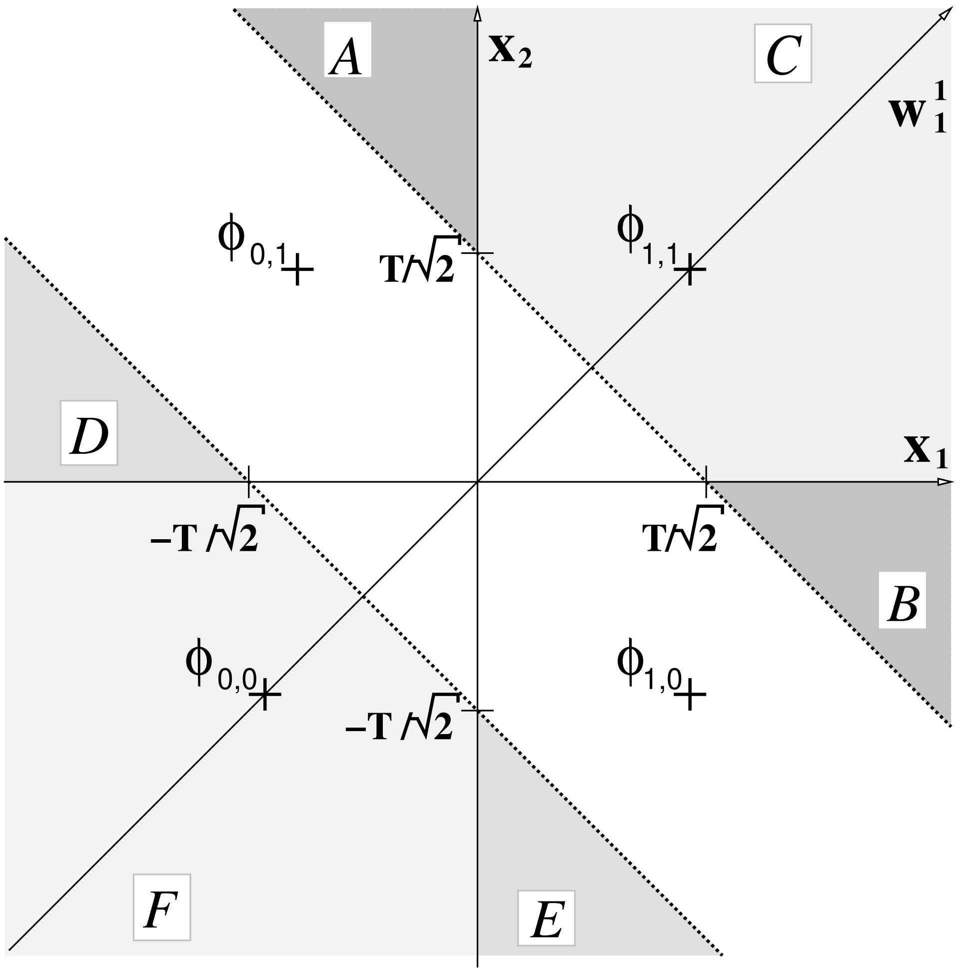


Figure 1:

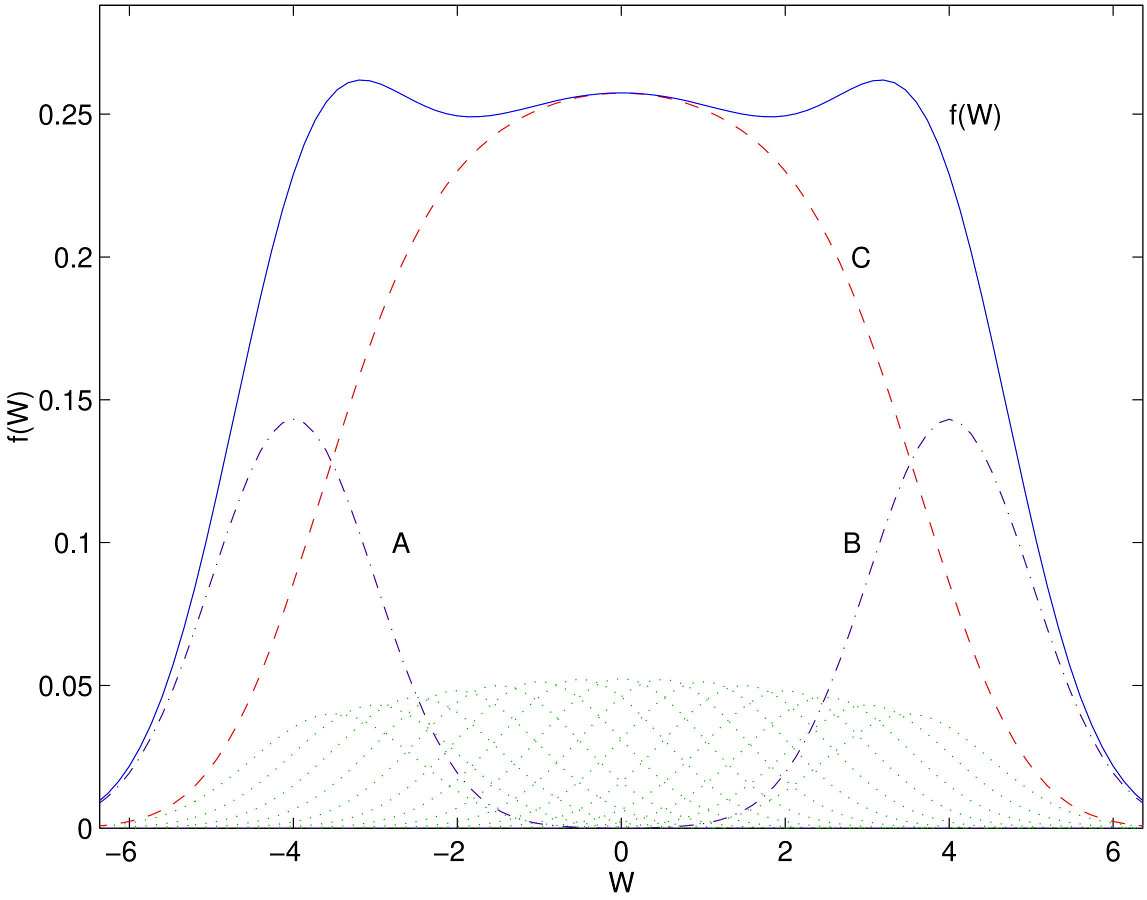
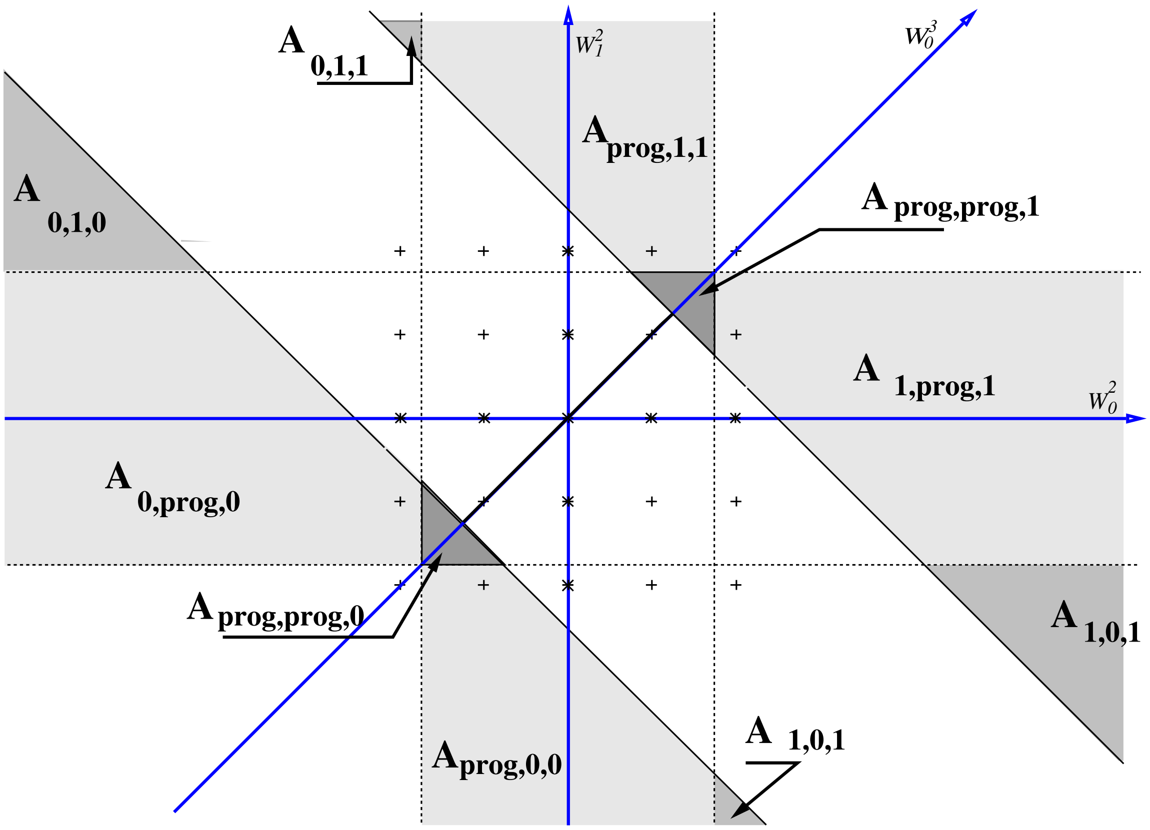


Figure 2:



W_1^2
W_0^3
W_0^2
$A_{0,1,1}$
$A_{prog,1,1}$
$A_{0,1,0}$
$A_{prog,prog,1}$
$A_{1,prog,1}$
$A_{0,prog,0}$
$A_{prog,prog,0}$
$A_{1,0,1}$
$A_{prog,0,0}$
$A_{1,0,1}$

Figure 3:

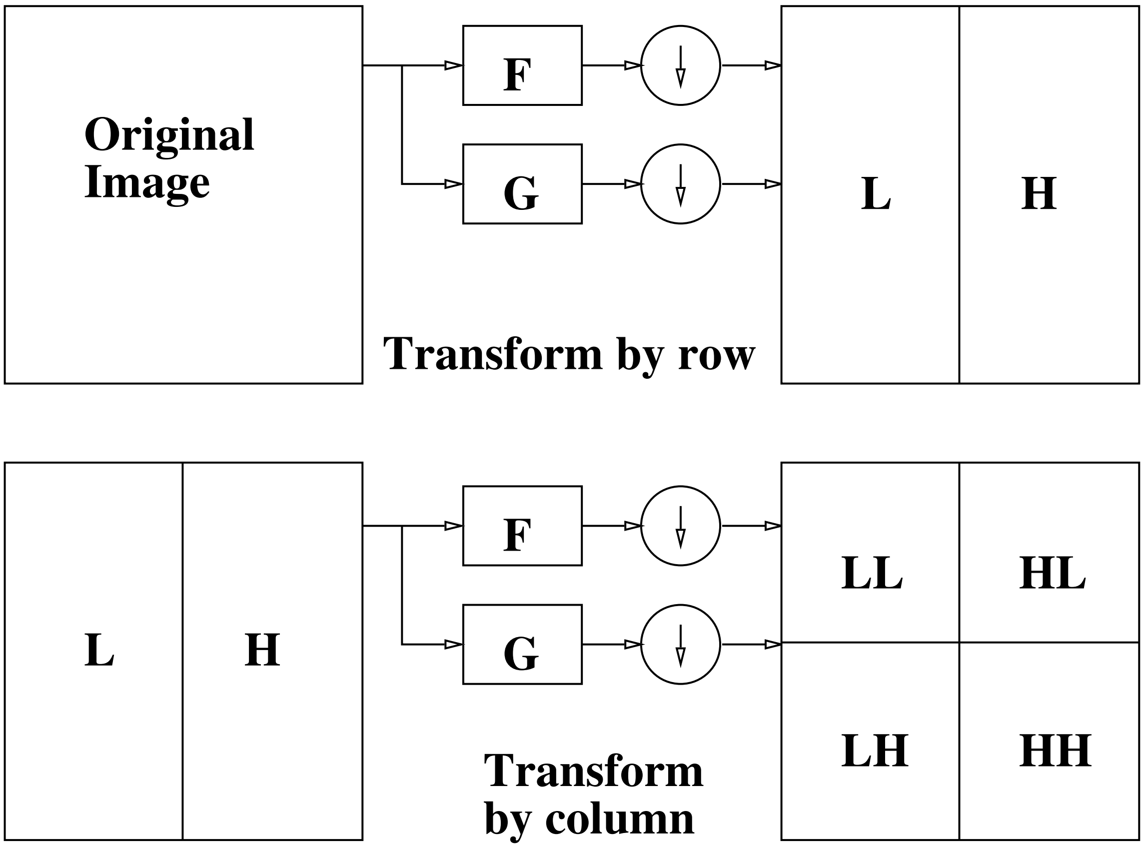

Original
Image
F
G
L
H
Transform by row
L
H
F
G
LL
HL
LH
HH
Transform
by column

Figure 4:

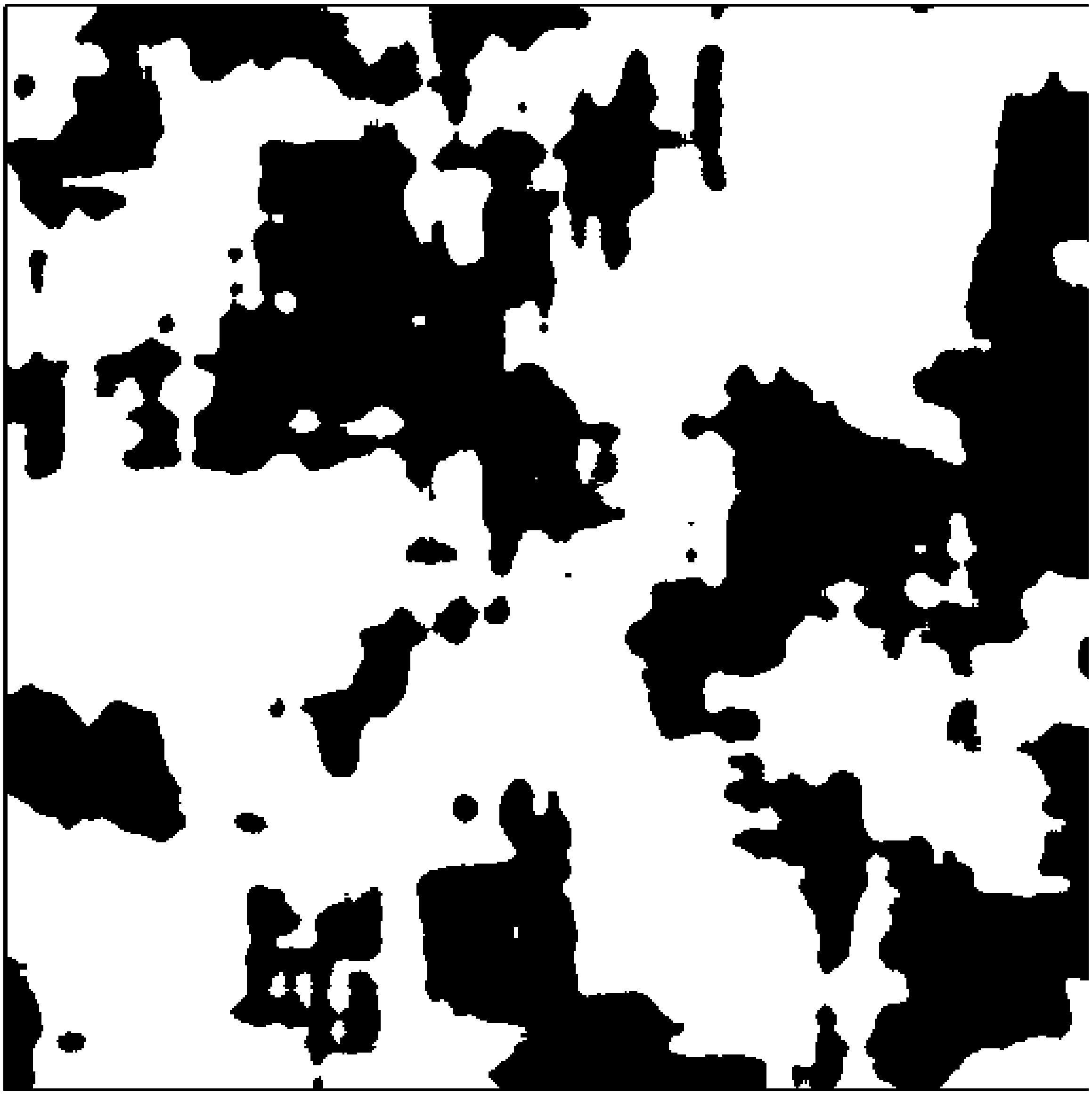


Figure 5: