Local discriminant bases*

Naoki Saito[1,2] and Ronald R. Coifman[2]

[1] Schlumberger-Doll Research
Old Quarry Road, Ridgefield, CT 06877-4108

[2] Department of Mathematics
Yale University
10 Hillhouse Avenue, New Haven, CT 06520

ABSTRACT

We describe an extension to the "best-basis" method to construct an orthonormal basis which maximizes a class separability for signal classification problems. This algorithm reduces the dimensionality of these problems by using basis functions which are well localized in time-frequency plane as feature extractors. We tested our method using two synthetic datasets: extracted features (expansion coefficients of input signals in these basis functions), supplied them to the conventional pattern classifiers, then computed the misclassification rates. These examples show the superiority of our method over the direct application of these classifiers on the input signals. As a further application, we also describe a method to extract signal component from data consisting of signal and textured background.
keywords: wavelet packets, local trigonometric transforms, classification, feature extraction, dimensionality reduction, linear discriminant analysis, classification and regression trees

1 INTRODUCTION

Extracting relevant features from signals or images is an important process for data analysis, such as classifying signals into known categories (*classification*) or predicting a response of interest based on these signals (*regression*). In this paper, we focus our attention on methods of selection of coordinate systems to enhance the performance of a few classification schemes.

More precisely, let $\mathcal{L} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$ be a *training* (or *learning*) dataset with N pairs of measurement vectors (or discrete signals) $\boldsymbol{x}_i$ and responses y_i, where $\mathcal{X} \subset \mathbb{R}^n$ is called a *signal space*, n is a dimensionality of each signal $\boldsymbol{x}_i$ (in this paper, we assume $n = 2^{n_0}$ for some n_0), and $\mathcal{Y}$ is called a *response space*. For classification problems, we set $\mathcal{Y} = \{1, \ldots, L\}$ where L is the number of known classes (for regression problems, we generally set $\mathcal{Y} = \mathbb{R}$). Let N_l be the number of signals belonging to class l, i.e., $N = N_1 + \cdots + N_L$, and let us denote a set of class l signals by $\{\boldsymbol{x}_i^{(l)}\}_{i=1}^{N_l}$. Now we want to find a map called *feature extractor* $f : \mathcal{X} \rightarrow \mathcal{F} \subset \mathbb{R}^k$, $(k \ll n)$ for extracting relevant features and reducing the dimensionality of the problem without losing important information as much as possible so that the following classification process can be improved in its accuracy and efficiency. The resulting range $\mathcal{F}$ is called a *feature space*. The final classification process now can be written as a map (generally

*in *Mathematical Imaging: Wavelet Applications in Signal and Image Processing* II, A. F. Laine and M. A. Unser, Editors, Proc. SPIE Vol. 2303, 1994

nonlinear) $g : \mathcal{F} \to \mathcal{Y}$. Preferably, the performance of the whole process should be measured by the misclassification rate using a *test* dataset $\mathcal{T}$ (which has not been used to construct the feature extractors and classifiers) as $(1/|\mathcal{T}|) \sum_{\boldsymbol{x}_i \in \mathcal{T}} \delta(y_i - g \circ f(\boldsymbol{x}_i))$, where $|\mathcal{T}|$ is a number of samples in $\mathcal{T}$, and $\delta(r \neq 0) = 1$ and $\delta(0) = 0$. If we use the training dataset for computing misclassification rates, we obviously have overly optimistic figures.

In this paper, we focus the feature extractors of the form $f = \Theta^{(k)} \circ \Psi$, where $\Theta^{(k)} : \mathcal{X} \to \mathcal{F}$ represents the selection rule (e.g., picking most important k coordinates from n coordinates), and $\Psi \in \mathrm{O}(n)$, i.e., an n-dimensional orthogonal matrix. As a classifier g, we adopt Linear Discriminant Analysis (LDA) of R. A. Fisher [6] (see also [7]) (in fact LDA itself does further feature extraction followed by a simple classification scheme) and Classification and Regression Trees (CART) [2] although other classifiers such as k-nearest neighbor (k-NN) [7], or artificial neural networks (ANN) [10] are all possible. The reader interested in comparisons of different classifiers is referred to the excellent review article of Ripley [10].

LDA first tries to find a linear map $A^T : \mathcal{X} \to \mathcal{F}$ (in this case not necessarily orthogonal matrix) which simultaneously minimizes the scatter of sample vectors (signals) within each class and maximizes the scatter of mean vectors $\{\boldsymbol{m}_l\}_{l=1}^L$ around the total mean vector $\boldsymbol{m} = \sum_{l=1}^L \pi_l \boldsymbol{m}_l$ where π_l is the prior probability of class l (which can be set to N_l/N without the knowledge on the true prior probability). The scatter of samples within each class can be measured by the within-class covariance matrix $\Sigma_w = \sum_{l=1}^L \pi_l \Sigma_l$, where Σ_l is the covariance matrix of class l. The scatter of mean vectors around the total mean can be measured by the between-class covariance matrix $\Sigma_b = \sum_{l=1}^L \pi_l (\boldsymbol{m}_l - \boldsymbol{m})(\boldsymbol{m}_l - \boldsymbol{m})^T$. Then, LDA equivalently maximizes a class separability index $J(A) = \mathrm{tr}[(A^T \Sigma_b A)^{-1} (A^T \Sigma_w A)]$ which measures how much these classes are separated in the feature space. This requires solving the so-called generalized (or pencil-type) eigenvalue problem, $\Sigma_b A = \Sigma_w A \Lambda$, where Λ is a diagonal matrix containing the eigenvalues. Once the map A is obtained (normally $k = L - 1$ for LDA), then the feature vector $A^T \boldsymbol{x}_i$ is computed for each i, and finally it is assigned to the class which has the mean vector closest to this feature vector in the Euclidean distance in this coordinate system. This is equivalent to bisecting the feature space $\mathcal{F}$ by hyperplanes. In this paper we regard LDA as a classifier although, as explained, it also includes its own feature extractor A. LDA is the optimal strategy if all classes of signals obey multivariate normal distributions with different mean vectors and an equal covariance matrix. However, in reality, it is hard to assume this condition. Moreover, since it relies on solving the eigensystem, LDA can only extract global features (or squeezes all discriminant information into a few $[L-1]$ basis vectors) so that the interpretation of the extracted features becomes difficult, it is sensitive to outliers and noise, and it requires $O(n^3)$ calculations.

Another popular classifier, CART, is a nonparametric method which recursively splits the input signal space *along* the coordinate axes and generates a partition of the input signal space into disjoint blocks so that the process can be conveniently described as a binary tree where nodes represent these splits. At each node, the split which best classifies the signals in the left and right branches is selected. Splitting is continued until nodes become "pure", i.e., they contain only one class of signals, or become "sparse", i.e., they contain only a few signals. Then the class label is assigned for each terminal node usually by majority vote of the samples belonging to that node. The pruning process to eliminate unimportant branches is usually applied after growing the initial tree to avoid the "overtraining." We refer the reader to [2] for the details of splitting, stopping, and pruning rules. Although CART does not assume any parametric model for the data distributions, we still face the difficulty of dealing with too many parameters in the original signal space and with too many computations if we consider linear combinations of the coordinates to generate a tree.

In order to fully utilize these classifiers, we must supply them *good* features (preferably just a few) and throw out useless part of the data. This improves both accuracy and speed of these classifiers. In this paper, we address how to construct good feature extractors. In particular, we use the "best-basis" paradigm [4] which permits a rapid [e.g., $O(n \log n)$] search among a large collection of orthogonal bases for the problem at hand; we select basis functions which are well localized in the time-frequency (or space-wave number) plane and most discriminate given classes, and then the coordinates (expansion coefficients) of these basis functions are fed into LDA or CART.

The organization of this paper is as follows. In Section 2, we propose a fast algorithm for constructing such a local basis for classification problems after reviewing the "best-basis" algorithm for signal compression. This is immediately followed by examples in Section 3. Then we discuss a method of signal/"background" separation as a further application of such a basis in Section 4 and finally conclude in Section 5.

We note that the concise version of this paper was announced earlier in [3] which also contains an algorithm for constructing a local basis for regression problems. The details of all these algorithms, their applications to regression problems, and examples using both synthetic and real datasets can be found in [12].

2 CONSTRUCTION OF LOCAL DISCRIMINANT BASIS

In this section, we describe a fast algorithm to construct an adaptive orthonormal basis which is localized in the time-frequency plane and which discriminates given signal classes.

2.1 Review of the best-basis algorithm

The *best-basis* algorithm of Coifman and Wickerhauser [4] was developed mainly for signal compression. This method first expands a given signal or a given collection of signals into a *library of orthonormal bases*, i.e., a redundant set of wavelet packet bases or local trigonometric bases having a binary tree structure where the nodes of the tree represent subspaces with different time-frequency localization characteristics. Then a complete basis called a *best basis* which minimizes a certain information cost functional (e.g., entropy) is searched in this binary tree using the divide-and-conquer algorithm. More precisely,

Definition 1 A *library of orthonormal bases* is a binary tree if it satisfies:

(a) Subsets of basis vectors can be identified with subintervals of $I = [0, 1[$ of the form $I_{j,k} = [2^{-j}k, 2^{-j}(k+1)[$, for $j = 0, 1, \ldots, J,\ k = 0, 1, \ldots, 2^j - 1$, where $J \leq n_0$.

(b) Each basis in the library corresponds to a disjoint cover of I by intervals $I_{j,k}$.

(c) If $\Omega_{j,k}$ is the subspace identified with $I_{j,k}$, then $\Omega_{j,k} = \Omega_{j+1,2k} \oplus \Omega_{j+1,2k+1}$.

Let $B_{j,k} = (\boldsymbol{w}_{j,k,0}, \ldots, \boldsymbol{w}_{j,k,2^{n_0-j}-1})^T$ be a set (matrix) of basis vectors belonging to the subspace $\Omega_{j,k}$. Note that each basis vector $\boldsymbol{w}_{j,k,m}$ is specified by the triplet (j, k, m) representing scale, frequency band, and time position respectively for wavelet packet bases, or scale, time window index, frequency respectively for local trigonometric bases. Also, we note that $B_{0,k}$ is the standard Euclidean system for wavelet packet libraries and is the usual discrete cosine (or sine) basis for local cosine (or sine) basis library. Now let $A_{j,k}$ be the best basis for the signal $\boldsymbol{x}$ restricted to the span of $B_{j,k}$ and let $\mathcal{J}$ be an information cost functional measuring the goodness of nodes (subspaces) for compression. Then, the best-basis algorithm works as:

Algorithm 1 (The Best-Basis Algorithm [4]) *Given a vector* $\boldsymbol{x}$,

Step 0: *Choose a time-frequency decomposition method [i.e., specify wavelet packet transform (i.e., a pair of quadrature mirror filters), local cosine transform, or local sine transform].*

Step 1: *Expand* $\boldsymbol{x}$ *into the library of orthonormal bases and obtain coefficients* $\{B_{j,k}\boldsymbol{x}\}_{0\le j\le J,\ 0\le k\le 2^j-1}$.

Step 2: *Set* $A_{J,k} = B_{J,k}$ *for* $k = 0, \ldots, 2^J - 1$.

Step 3: *Determine the best subspace* $A_{j,k}$ *for* $j = J-1, \ldots, 0,\ k = 0, \ldots, 2^j - 1$ *by*

$$(1) \qquad A_{j,k} = \begin{cases} B_{j,k} & \text{if } \mathcal{J}(B_{j,k}\boldsymbol{x}) \le \mathcal{J}(A_{j+1,2k}\boldsymbol{x} \cup A_{j+1,2k+1}\boldsymbol{x}), \\ A_{j+1,2k} \oplus A_{j+1,2k+1} & \text{otherwise.} \end{cases}$$

To make this algorithm fast, the cost functional $\mathcal{J}$ needs to be *additive*:

Definition 2 A map $\mathcal{J}$ from sequences $\{x_i\}$ to $\mathbb{R}$ is said to be *additive* if $\mathcal{J}(0) = 0$ and $\mathcal{J}(\{x_i\}) = \sum_i \mathcal{J}(x_i)$.

Thus, if $\mathcal{J}$ is additive, then in (1) we have $\mathcal{J}(A_{j+1,2k}\boldsymbol{x} \cup A_{j+1,2k+1}\boldsymbol{x}) = \mathcal{J}(A_{j+1,2k}\boldsymbol{x}) + \mathcal{J}(A_{j+1,2k+1}\boldsymbol{x})$, i.e., a simple addition suffices instead of computing the cost of union of the nodes. A popular measure as $\mathcal{J}$ is an *entropy* of a nonnegative sequence $\boldsymbol{p}$ with $\sum p_i = 1$:

$$(2) \qquad H(\boldsymbol{p}) = -\sum_i p_i \log p_i,$$

with the convention $0 \cdot \log 0 = 0$. For a general sequence or signal $\boldsymbol{x}$, we set $p_i = (|x_i|/\|\boldsymbol{x}\|)^2$ where $\|\cdot\|$ is the ℓ^2 norm and define $H_2(\boldsymbol{x}) \triangleq H(\boldsymbol{x}/\|\boldsymbol{x}\|)$. Although (2) is additive with respect to $\boldsymbol{p}$, but $H_2(\boldsymbol{x})$ is not additive with respect to $\boldsymbol{x}$. However, it is easy to show that minimizing the additive measure $h(\boldsymbol{x}) = -\sum_i |x_i|^2 \log |x_i|^2$ implies minimizing $H_2(\boldsymbol{x})$.

2.2 Discriminant measures

The cost functional $\mathcal{J}$ such as (2) measures the flatness of the energy distribution of the signal so that minimizing this leads to an efficient representation (or coordinate system) for the signal. Because of this cost functional, the best-basis algorithm is good for signal compression but is not necessarily good for classification problems; for classification, we need a measure to evaluate the power of discrimination of the nodes in the tree-structured bases. Once the discriminant measure (or discriminant information functional) is specified, we can compare the goodness of each node for the classification problem to that of union of the two children nodes and can judge whether we should keep the children nodes or not, in the same manner as the best-basis search algorithm.

There are many choices for the discriminant measure (see e.g., [1]); all of them essentially measure "statistical distances" among classes. For simplicity, let us first consider the two-class case. Let $\boldsymbol{p} = \{p_i\}_{i=1}^n$, $\boldsymbol{q} = \{q_i\}_{i=1}^n$ be two nonnegative sequences with $\sum p_i = \sum q_i = 1$ (which can be viewed as normalized energy distributions of signals belonging to class 1 and class 2 respectively in a coordinate system). The discriminant information functional $\mathcal{D}(\boldsymbol{p}, \boldsymbol{q})$ between these two sequences should measure how differently $\boldsymbol{p}$ and $\boldsymbol{q}$ are distributed. One natural choice for $\mathcal{D}$ is the so-called *relative entropy* (also known as *cross entropy*, *Kullback-Leibler distance*, or *I-divergence*) [9]:

$$(3) \qquad I(\boldsymbol{p}, \boldsymbol{q}) \triangleq \sum_{i=1}^n p_i \log \frac{p_i}{q_i},$$

with the convention, $\log 0 = -\infty$, $\log(x/0) = +\infty$ for $x \gneqq 0$, $0 \cdot (\pm\infty) = 0$. It is clear that $I(\boldsymbol{p}, \boldsymbol{q}) \geq 0$ and equality holds iff $\boldsymbol{p} \equiv \boldsymbol{q}$. This quantity is not a metric since it is not symmetric and does not satisfy the triangle inequality. But it measures the discrepancy of $\boldsymbol{p}$ from $\boldsymbol{q}$. Note that if $q_i = 1/n$ for all i, i.e., q_i are distributed uniformly, then $I(\boldsymbol{p}, \boldsymbol{q}) = -H(\boldsymbol{p})$, the negative of the entropy of the sequence $\boldsymbol{p}$ itself.

The relative entropy (3) is asymmetric in $\boldsymbol{p}$ and $\boldsymbol{q}$. For certain applications the asymmetry is preferred (see e.g., Section 4). If, however, a symmetric quantity is preferred, one should use the *J-divergence* between $\boldsymbol{p}$ and $\boldsymbol{q}$ [9]:

$$J(\boldsymbol{p}, \boldsymbol{q}) \triangleq I(\boldsymbol{p}, \boldsymbol{q}) + I(\boldsymbol{q}, \boldsymbol{p}). \tag{4}$$

Another possibility of the measure $\mathcal{D}$ is a ℓ^2 analogue of $I(\boldsymbol{p}, \boldsymbol{q})$ [14]:

$$W(\boldsymbol{p}, \boldsymbol{q}) = \|\boldsymbol{p} - \boldsymbol{q}\|^2 = \sum_{i=1}^{n} (p_i - q_i)^2. \tag{5}$$

Clearly, ℓ^p $(p \geq 1)$ versions of this measure are all possible.

To obtain a fast computational algorithm, the measure $\mathcal{D}$ should be *additive* similarly to $\mathcal{J}$:

Definition 3 The discriminant measure $\mathcal{D}(\boldsymbol{p}, \boldsymbol{q})$ is said to be *additive* if

$$\mathcal{D}(\{p_i\}_{i=1}^n, \{q_i\}_{i=1}^n) = \sum_{i=1}^{n} \mathcal{D}(p_i, q_i) \tag{6}$$

The measures (3) (subsequently (4) as well) and (5) are both additive.

For measuring discrepancies among L distributions, $\boldsymbol{p}^{(1)}, \dots, \boldsymbol{p}^{(L)}$, one may take $\binom{L}{2}$ pairwise combinations of $\mathcal{D}$:

$$\mathcal{D}(\{\boldsymbol{p}^{(l)}\}_{l=1}^L) \triangleq \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} \mathcal{D}(\boldsymbol{p}^{(i)}, \boldsymbol{p}^{(j)}). \tag{7}$$

2.3 The local discriminant basis algorithm

Given an additive discriminant measure $\mathcal{D}$, what quantity should be supplied to $\mathcal{D}$ to measure the discrimination power of each node in the library? In order to fully utilize the time-frequency localization characteristics of our libraries of bases, we compute the following *time-frequency energy map* for each class and supply them to $\mathcal{D}$:

Definition 4 Let $\{\boldsymbol{x}_i^{(l)}\}_{i=1}^{N_l}$ be a set of training signals belonging to class l. Then the *time-frequency energy map* of class l, denoted by Γ_l, is a table of real values specified by the triplet (j, k, m) as

$$\Gamma_l(j,k,m) \triangleq \sum_{i=1}^{N_l} (\boldsymbol{w}_{j,k,m}^T \boldsymbol{x}_i^{(l)})^2 / \sum_{i=1}^{N_l} \|\boldsymbol{x}_i^{(l)}\|^2, \tag{8}$$

for $j = 0, \dots, J,\ k = 0, \dots, 2^j - 1,\ m = 0, \dots, 2^{n_0 - j} - 1.$

In other words, Γ_l is computed by accumulating the squares of expansion coefficients of the signals at each position in the table followed by the normalization by the total energy of the signals belonging to class l. (This normalization is important especially if there is significant differences in number of samples among classes.) In the following, we use the notation:

$$\mathcal{D}(\{\Gamma_l(j,k,\cdot)\}_{l=1}^L) = \sum_{m=0}^{2^{n_0-j}-1} \mathcal{D}(\Gamma_1(j,k,m),\ldots,\Gamma_L(j,k,m)).$$

Here is an algorithm to select a local orthonormal basis (from the library) which best discriminates the given classes in terms of their time-frequency energy distributions. We call this a *local discriminant basis* (LDB). Similarly to the best-basis algorithm, let $A_{j,k}$ represent the LDB restricted to the span of $B_{j,k}$ which is a set of basis vectors at (j,k) node. Also, let $\Delta_{j,k}$ be a work array containing the discriminant measure of the node (j,k). We assume the additive discriminant measure $\mathcal{D}$ here.

Algorithm 2 (The Local Discriminant Basis Algorithm) *Given a training dataset $\mathcal{L}$ consisting of L classes of signals $\{\{\boldsymbol{x}_i^{(l)}\}_{i=1}^{N_l}\}_{l=1}^L$,*

Step 0: *Choose a time-frequency decomposition method [i.e., specify wavelet packet transform (i.e., a pair of quadrature mirror filters), local cosine transform, or local sine transform].*

Step 1: *Construct time-frequency energy maps Γ_l for $l = 1,\ldots,L$.*

Step 2: *Set $A_{J,k} = B_{J,k}$ and $\Delta_{J,k} = \mathcal{D}(\{\Gamma_l(J,k,\cdot)\}_{l=1}^L)$ for $k = 0,\ldots,2^J-1$.*

Step 3: *Determine the best subspace $A_{j,k}$ for $j = J-1,\ldots,0,\ k = 0,\ldots,2^j-1$ by the following rule:*

Set $\Delta_{j,k} = \mathcal{D}(\{\Gamma_l(j,k,\cdot)\}_{l=1}^L)$.
If $\Delta_{j,k} \geq \Delta_{j+1,2k} + \Delta_{j+1,2k+1}$,
then $A_{j,k} = B_{j,k}$,
else $A_{j,k} = A_{j+1,2k} \oplus A_{j+1,2k+1}$ *and set* $\Delta_{j,k} = \Delta_{j+1,2k} + \Delta_{j+1,2k+1}$.

Step 4: *Order the basis functions by their power of discrimination (see below).*

Step 5: *Use $k\,(\ll n)$ most discriminant basis functions for constructing classifiers.*

The selection (or pruning) process in Step 3 is fast, i.e., $O(n)$ since the measure $\mathcal{D}$ is additive. After this step, we have a complete orthonormal basis LDB. Once the LDB is selected, we can use all expansion coefficients of signals in this basis as features; however, if we want to reduce the dimensionality of the problem, the following two steps are still necessary.

In Step 4, there are several choices as a measure of discriminant power of an individual basis function. For simplicity in notation, let $\lambda = (j,k,m) \in \mathbb{Z}^3$ be a triplet specifying the LDB selected in Step 3, and let $\alpha_{\lambda,i}^{(l)} = \boldsymbol{w}_\lambda^T \boldsymbol{x}_i^{(l)}$, i.e., an expansion coefficient of $\boldsymbol{x}_i^{(l)}$ in the basis vector $\boldsymbol{w}_\lambda$.

(a) the discriminant measure of a single basis function $\boldsymbol{w}_\lambda$:

$$\mathcal{D}(\Gamma_1(\lambda),\ldots,\Gamma_L(\lambda)). \tag{9}$$

(b) the Fisher's class separability of the expansion coefficients onto the basis function $\boldsymbol{w}_\lambda$:

$$\frac{\sum_{l=1}^L \pi_l(\mathrm{mean}_i(\alpha_{\lambda,i}^{(l)}) - \mathrm{mean}_l(\mathrm{mean}_i(\alpha_{\lambda,i}^{(l)})))^2}{\sum_{l=1}^L \pi_l\,\mathrm{var}_i(\alpha_{\lambda,i}^{(l)})}, \tag{10}$$

where $\mathrm{mean}_i(\cdot)$ and $\mathrm{var}_i(\cdot)$ are operations to take the sample mean and variance with respect to the samples indexed by i, respectively.

(c) the robust version of (b):

$$\frac{\sum_{l=1}^{L} \pi_l |\operatorname{med}_i(\alpha_{\lambda,i}^{(l)}) - \operatorname{med}_l(\operatorname{med}_i(\alpha_{\lambda,i}^{(l)}))|}{\sum_{l=1}^{L} \pi_l \operatorname{mad}_i(\alpha_{\lambda,i}^{(l)})}, \tag{11}$$

where $\operatorname{med}_i(\cdot)$ and $\operatorname{mad}_i(\cdot)$ are operations to take the sample median and median absolute deviation with respect to the samples indexed by i, respectively.

See [1, 8] for more examples. We note that this step can also be viewed as a restricted version of the projection pursuit algorithm [8].

Step 5 reduces the dimensionality of the problem from n to k without losing the discriminant information in terms of time-frequency energy distributions among classes. Thus many interesting statistical techniques which are usually computationally too expensive for n dimensional problems become feasible. How to select the best k is a tough interesting question. One possibility is to use model selection methods such as the minimum description length (MDL) criterion [11, 13].

3 EXAMPLES

To demonstrate the capability of the local discriminant basis, we conducted two classification experiments using synthetic signals. In both cases, we specified three classes of signals by analytic formulas. For each class, we generated 100 training signals and 1000 test signals. We first applied LDA and Classification Tree (CT) to the training signals of the original coordinate (i.e., standard Euclidean) system, and obtained the classification rules. Then the test signals were fed into these classifiers and the misclassification rates were computed. Next we computed the LDB (using the relative entropy as $\mathcal{D}$) from the training signals, selected a small number of most discriminant basis functions [in terms of the component-wise relative entropy (9)], and applied LDA and CT to the resulting coefficients. Finally the test signals were projected onto the LDB functions and fed into these classifiers; then the misclassification rates were computed. For each method, we also computed the misclassification rate on the training dataset.

Example 1 *Triangular waveform classification.* This is an example for classification originally examined in [2]. The dimensionality of the signal was extended from 21 in [2] to 32 for the dyadic dimensionality requirement of the bases under consideration. Three classes of signals were generated by the following formulas:

$$\begin{aligned} x^{(1)}(i) &= uh_1(i) + (1-u)h_2(i) + \epsilon(i) \quad \text{for Class 1,} \\ x^{(2)}(i) &= uh_1(i) + (1-u)h_3(i) + \epsilon(i) \quad \text{for Class 2,} \\ x^{(3)}(i) &= uh_2(i) + (1-u)h_3(i) + \epsilon(i) \quad \text{for Class 3,} \end{aligned}$$

where $i = 1, \ldots, 32$, $h_1(i) = \max(6 - |i-7|, 0)$, $h_2(i) = h_1(i-8)$, $h_3(i) = h_1(i-4)$, u is a uniform random variable on the interval $(0,1)$, and $\epsilon(i)$ are standard normal variates. Figure 1 shows five sample waveforms from each class. The LDB was computed from the wavelet packet coefficients with the 6-tap coiflet filter [5]. Then the five most discriminant coordinates were selected. In Figure 2, we compare the top five vectors from LDA and LDB. Only top two vectors were useful in LDA in this case. The top five LDB vectors look similar to the functions h_j or their derivatives whereas it is difficult to interpret the LDA vectors. The misclassification rates are given in the table:

Method	Error Training	rate (%) Test
LDA on the standard coordinate system	13.33	20.90
CT on the standard coordinate system	6.33	29.87
LDA to Top 5 LDB coordinates	14.33	15.90
CT to Top 5 LDB coordinates	7.00	21.37

The best result so far was obtained applying LDA to the LDB coordinates. We would like to note that according to Breiman et al. [2], the Bayes error of this example is about 14 %.

Example 2 *Signal shape classification.* The second example is a signal shape classification problem. In this example, we try to classify synthetic noisy signals with various shapes, amplitudes, lengths, and positions into three possible classes. More precisely, sample signals of the three classes were generated by:

$$\begin{aligned} c(i) &= (6+\eta)\cdot\chi_{[a,b]}(i)+\epsilon(i) && \text{for “cylinder” class,}\\ b(i) &= (6+\eta)\cdot\chi_{[a,b]}(i)\cdot(i-a)/(b-a)+\epsilon(i) && \text{for “bell” class,}\\ f(i) &= (6+\eta)\cdot\chi_{[a,b]}(i)\cdot(b-i)/(b-a)+\epsilon(i) && \text{for “funnel” class,} \end{aligned}$$

where $i = 1, \ldots, 128$, a is an integer-valued uniform random variable on the interval $[16, 32]$, $b-a$ also obeys an integer-valued uniform distribution on $[32, 96]$, η and $\epsilon(i)$ are standard normal variates, and $\chi_{[a,b]}(i)$ is the characteristic function on the interval $[a, b]$. Figure 3 shows five sample waveforms from each class. If there is no noise, we can characterize "cylinder" signals by two step edges and constant values around the center, "bell" signals by one ramp and one step edge in this order and positive slopes around the center, and "funnel" signals by one step edge and one ramp in this order and negative slopes around the center.

The 12-tap coiflet filter [5] was used for the LDB selection. Then the 10 most important coordinates were selected. In Figure 4, we compare the top 10 LDA and LDB vectors. Again, only the top two vectors were used for classification in LDA case. These LDA vectors are very noisy and it is difficult to interpret what information they captured. On the other hand, we can observe that the top 10 LDB vectors are located around the edges the centers of the signals. Also note that some of the vectors work as a smoother (low pass filter) and the others work as a edge detector (band pass filter), so that the resulting expansion coefficients carry the information on the edge positions and types. The misclassification rates in this case are:

Method	Error Training	rate (%) Test
LDA on the standard coordinate system	0.33	13.17
CT on the standard coordinate system	3.00	13.37
LDA to Top 10 LDB coordinates	3.67	6.20
CT to Top 10 LDB coordinates	3.00	3.83

As expected, LDA applied to the original coordinate system was almost perfect with respect to the training data, but it adapted too much to the training data, so it lost flexibility; when applied to the new test dataset, it did not work well. The best result was obtained using the CT on the LDB coordinates in this case. In this case, the misclassification rates of the training data and test data are very close; that is, the algorithm really "learned" the structures of signals.

From these examples, we can see that it is more important to select the good features than to select the best possible classifier without supplying the good features; each classifier has its advantages and disadvantages [10], i.e., the best classifier heavily depends on the problem (e.g., LDA was better than CART in Example 1 whereas the situation was opposite in Example 2.) By supplying a handful of good features, we can greatly enhance the performance of classifiers.

4 SIGNAL/BACKGROUND SEPARATION BY LDB

LDB vectors can also be used as a tool for extracting signal component from the data obscured by some unwanted noise or "background" (which may not be random). Let class 1 consist of a signal plus noise or a signal plus "background" and let class 2 consist of a pure noise or "background" Then, by selecting the LDB maximizing $\mathcal{D}$ between class 1 and class2, we can construct the best basis for denoising arbitrary noise or pulling a signal out of a textured background. In this application, the asymmetric relative entropy (3) makes more sense than the symmetric version (4).

We show one example here. As "background" (class 2), we generated 100 synthetic sinusoid with random phase as $b(k) = \sin(\pi(k/32 + u))$, where $k = 1, \ldots, 128$, and u is a uniform random variable on $(0, 1)$. As class 1 samples, we again generated 100 "backgrounds", and added a small spike (as a "signal" component) for each sample vector randomly between $20 \leq k \leq 60$, i.e., $x(k) = \sin(\pi(k/32 + u)) + 0.01\delta_{k,r}$, where $\delta_{k,r}$ is the Kronecker delta and r is an integer-valued uniform random variable on the interval $[20, 60]$. Figure 5 shows how these "backgrounds" were removed. Figure 5 (a) shows 10 sample vectors of class 1. We can hardly see the spikes. Then we transformed both class 1 and 2 samples by the discrete sine transform (DST) into "frequency" domain. Figure 5 (b) shows the transformed version of Figure 5 (a). Then these DST coefficients of both classes were supplied to the LDB algorithm of Section 2 using the local sine basis library (which essentially does segmentation in frequency domain). After the LDB was found, the basis vectors were sorted by (9). The top 20 LDB vectors are displayed in Figure 5 (c). We can clearly see that the top eight basis vectors are concentrated around low frequency region and other vectors are located in higher frequency region. We regard the subspace spanned by these eight LDB vectors as "background" using the *a priori* knowledge that the "background" component consists of only low frequency component. The reason why these vectors have large values in (9) is that the "background" parts of class 1 samples are different from class 2 samples in phase, and the DST is not a shift-invariant transform. After removing the component belonging to this "background" subspace, we reconstructed the "signal" component of class 1 samples by inverse DST which are shown in Figure 5 (d). We can clearly see the spikes now.

5 CONCLUSION

We have described an algorithm to construct an adaptive local orthonormal basis (LDB) for classification problems. The basis functions generated by this algorithm can capture relevant local features (in both time and frequency) in data. LDB provides us with better insight and understanding of relationships between the essential features of the input signals and the corresponding outputs (class names), and enhances the performance of classifiers. We have demonstrated that LDB can also be used for pulling out signal component from the data consisting of signals plus "backgrounds."

References

[1] M. Basseville, "Distance measures for signal processing and pattern recognition", Signal Processing, Vol. 8, no. 4, pp. 349–369, 1989.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Chapman and Hall, Inc., New York, 1993, previously published by Wadsworth and Brooks/Cole in 1984.

[3] R. R. Coifman and N. Saito, "Constructions of local orthonormal bases for classification and regression", Comptes Rendus Acad. Sci. Paris, Série I, Vol. 319, 1994, to appear.

[4] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection", IEEE Trans. Inform. Theory, Vol. 38, no. 2, pp. 713–719, 1992.

[5] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 61, SIAM, Philadelphia, 1992.

[6] R. A. Fisher, "The use of multiple meaurements in taxonomic problems", Ann. Eugenics, Vol. 7, pp. 179–188, 1936.

[7] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, San Diego, 1990.

[8] P. J. Huber, "Projection pursuit (with discussions)", Ann. Statist., Vol. 13, no. 2, pp. 435–525, 1985.

[9] S. Kullback and R. A. Leibler, "On information and sufficiency", Ann. Math. Statist., Vol. 22, pp. 79–86, 1951.

[10] B. D. Ripley, "Statistical aspects of neural networks", *Networks and Chaos: Statistical and Probabilistic Aspects*, O. E. Barndorff-Nielsen, J. L. Jensen, D. R. Cox, and W. S. Kendall, eds., pp. 40–123, Chapman and Hall, Inc., New York, 1993.

[11] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific, Singapore, 1989.

[12] N. Saito, *Local Feature Extraction and Its Applications Using a Library of Bases*, Ph.D. thesis, Dept. of Mathematics, Yale University, New Haven, CT 06520 USA, 1994, in preparation.

[13] ———, "Simultaneous noise suppression and signal compression using a library of orthonormal bases and the minimum description length criterion", *Wavelets in Geophysics*, E. Foufoula-Georgiou and P. Kumar, eds., pp. 299–324, Academic Press, San Diego, 1994.

[14] S. Watanabe, *Pattern Recognition: Human and Mechanical*, John Wiley and Sons, New York, 1985.

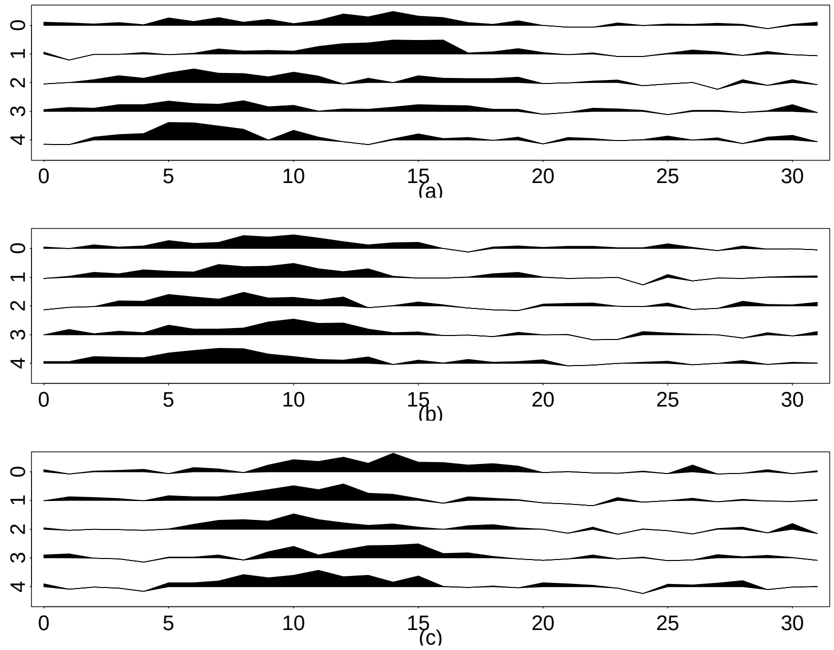

Figure 1: Five sample waveforms from (a) Class 1, (b) Class 2, and (c) Class3.

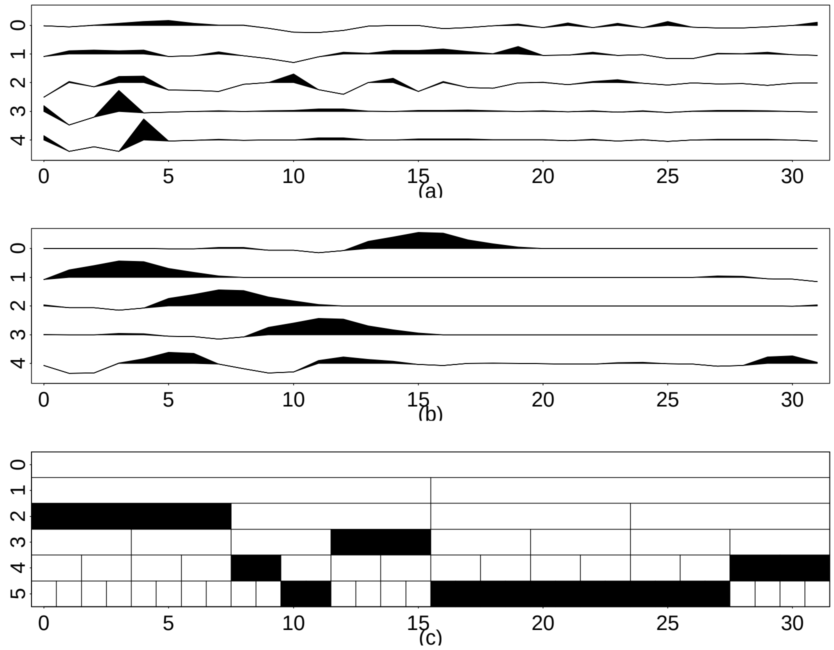

Figure 2: Plots from the analysis of Example 1: (a) Top five LDA vectors. (b) Top 5 LDB vectors. (c) The nodes selected as LDB.

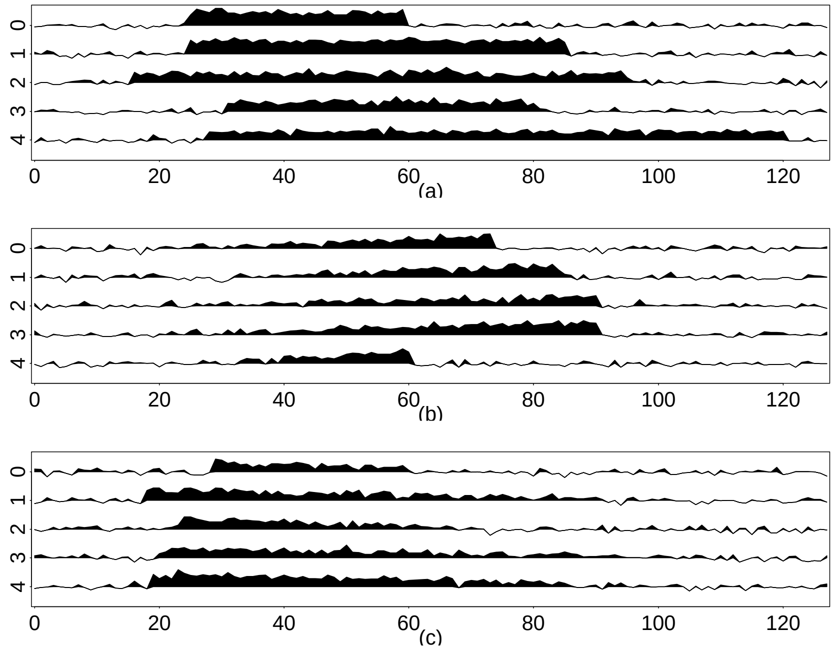

Figure 3: Five sample waveforms from (a) "cylinder" class, (b) "bell" class, and (c) "funnel" class.

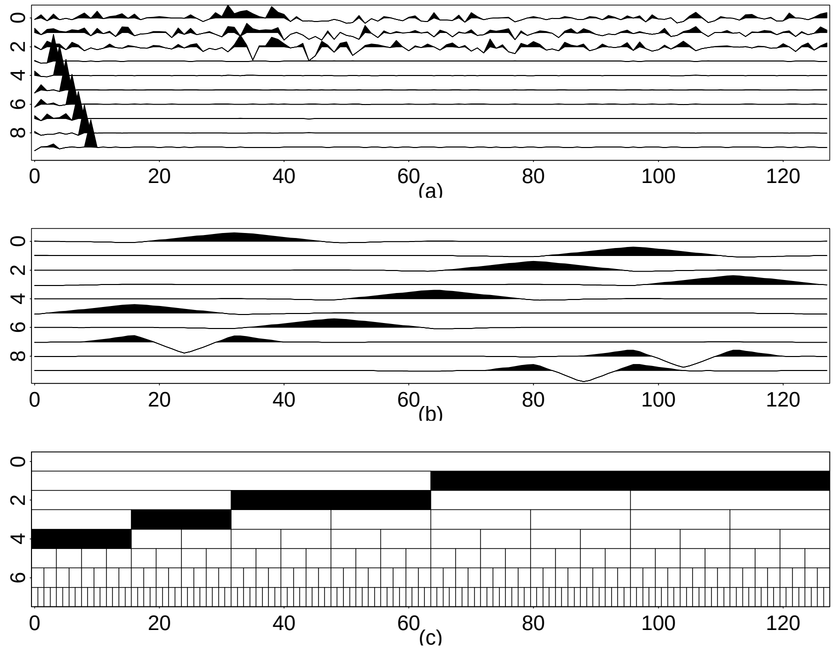

Figure 4: Plots from the analysis of Example 2: (a) Top 10 LDA vectors. (b) Top 10 LDB vectors. (c) The nodes selected as LDB.

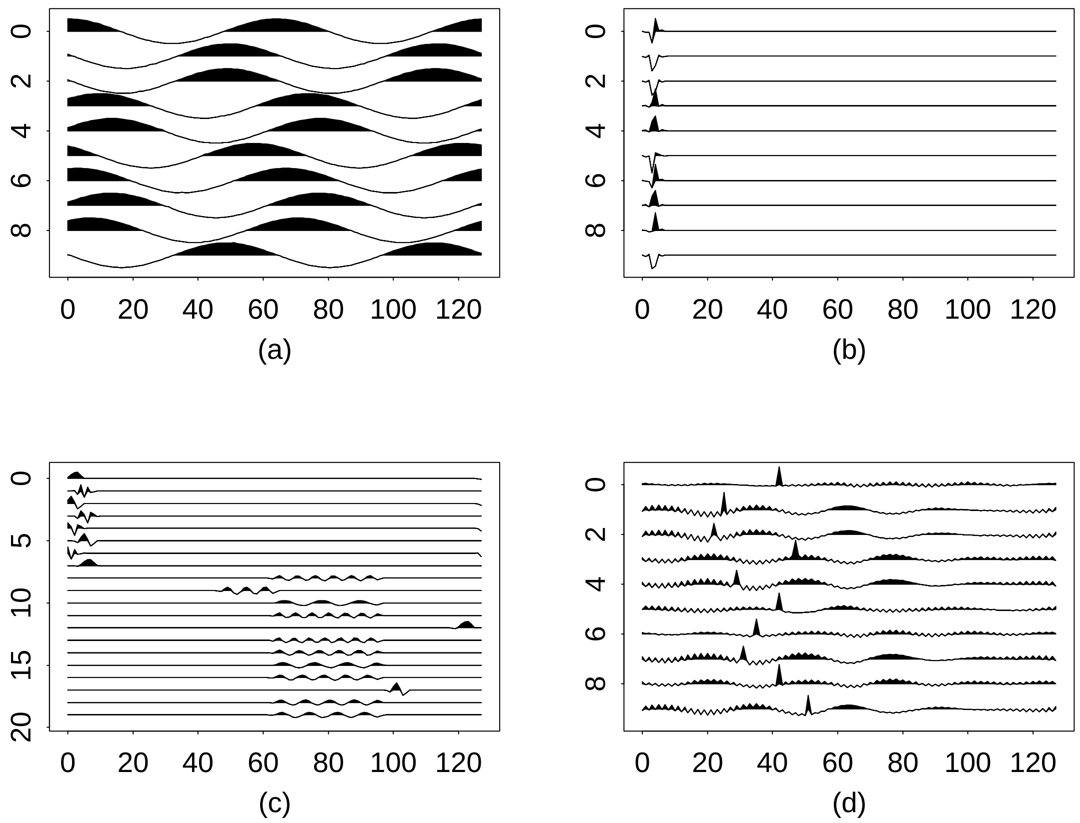

Figure 5: (a) Ten samples of Class 1 vectors, i.e., sinusoids plus spikes. (b) DST coefficients of vectors in (a). (c) Top 20 LDB vectors using the local sine library on the frequency domain. (d) Reconstructed spikes after removing the "background".