ALTERNATIVE LOCAL DISCRIMINANT BASES USING EMPIRICAL EXPECTATION AND VARIANCE ESTIMATION

EIRIK FOSSGAARD

ABSTRACT. We propose alternative discriminant measures for selecting the best basis for classification purposes among a large collection of orthonormal bases organized in a binary tree structure. A generalization of the Local Discriminant Basis Algorithm of Saito and Coifman is constructed. The success of these new methods is evaluated and compared to earlier methods in experiments.

KEY WORDS: Local Feature Extraction, Discriminant Measures, Cluster Search.

1. INTRODUCTION

This paper is the result of an attempt to improve on the method applied in [Fos97] to discriminate between two distinct classes/types of signals by using expansions of the data in wavelet packet/local trigonometric bases. This method was first invented and described by N.Saito and R.Coifman. For a thorough exposition to this theme, see [Sai94] and [CS96]. A summary of the main ideas is given below.

1.1. **Definition of the problem.** The problem which concerns us in this article is to discriminate between two or more different types/classes of signals. We will do this by constructing a classification scheme by finding the best selection possible inside some limited intervals of a set of parameters governing and defining a certain algorithm, the "Dyadic Cluster Search Algorithm" (the DCS-algorithm) to be described below, such that the overall misclassification rate is minimized on a given set of training data. As a motivating example we will use the problem of dicriminating between two different configurations Q_n and Q_m of coherent electromagnetic wave sources located inside some small plane disk centered in the origin. The resulting signals S_{Q_n} and S_{Q_m} from these configurations of wavesources are simply superpositions of functions of the form $e^{-i\omega t}\phi(\mathbf{x})$, where ϕ is a elementary solution of the Helmholtz equation: $(\nabla^2 + (\frac{\omega}{c})^2)\phi(\mathbf{x}) = 0$, $\mathbf{x} \in \mathbf{R}^3$. We generate waveforms by sampling the resulting signal $S_{Q_n}(\mathbf{x}, t)$ in constant time at discrete points $\mathbf{x}$ at a large distance from the origin. The reasons for studying this particular example are that it can model several interesting practical problems, and that the different waveforms corresponding to different wave source configurations are difficult to discriminate. For details concerning this example, see [Fos97]. These waveforms are

Eirik Fossgaard is currently a lecturer at Institute of Mathematics and Statistics, University of Tromsø, 9037 Tromsø, Norway (E-mail: eirikf@math.uit.no). This work was supported by Department of Mathematics, Royal Institute of Technology, 100 44 Stockholm, Sweden. The author thanks Professor Jan Olov Strömberg, Department of Mathematics, Royal Institute of Technology, 100 44 Stockholm, Sweden, for helpful suggestions.

used to test our classification scheme in the Experiment section in this paper in addition to a more classical three-class waveform discrimination problem.

We will take as our starting point the approach of Saito and Coifman as presented in [Sai94], [CS95] and [CS96]. The main ideas in this approach are briefly described below.

Each signal belonging to a training dataset is decomposed in a time/space - frequency *dictionary*, that is a decomposition into a large collection of orthonormal bases arranged in a binary tree structure, containing either *wavelet-packet* basis functions, or *local trigonometric* basis functions. A measure of *energy-density* is then computed for each coordinate in the dictionary for each class of signals, originally in [Sai94] this is taken to be the square of the coordinate summed over all the training signals belonging to a class of signals, and then normalized by the total energy projected onto this coordinate. Then a basis called the "Local Discriminant Basis", LDB for short, is chosen from the dictionary by maximizing a certain *discrimination measure*, defined by some *additive* cost-functional, over the dictionaries of energy-densities. The coordinates where the discrimination measure takes on its largest values are called the most important *features* of the signals. These coordinates are selected from the LDB and used as input for some classifier.

This method is very powerful in many cases, but it also has its weaknesses, a serious one is that the LDB is not able to distinguish two signals both consisting exclusively of one and the same basis element, differing only in the sign. One way of dealing with this problem is described in [CS96], where one estimates the *probability-density functions*, pdf's, of the projections onto the different basis elements in the dictionary, and selects the basis which maximizes some well-chosen functional on these pdf's.

We will try to improve on the LDB-method described above, by constructing new dicrimination measures that yield more relevant features. We will also try to improve the performance of the algorithm by using *several* LDB's in sequence, and by using a classifier specially designed to fully utilize the increased degree of freedom that multiple LDB's (MLDB's) give us in selecting features which are most important to our problem.

1.2. **The original LDB method.** The problem as expressed in [Sai94] is optimizing a map: $d : \mathcal{X} \to \mathcal{Y}$, where $\cup_{y\in\mathcal{Y}}\mathcal{X}^{(y)} = \mathcal{X} \subset \mathbf{R}^n$ is the input set of signals, $\mathcal{Y} = \{1, 2, ..., N\}$ is the output set of class labels, and $\mathcal{X}^{(y)}$ is the subset of class y signals. To optimize the map d, one considers maps of the form

$$d = c \circ \mathcal{F}_K \circ \mathbf{\Psi}_{n\times n}, \tag{1}$$

where the *feature extractor* $\mathbf{\Psi}_{n\times n} \in O(n)$ is an orthogonal $n \times n$ matrix which represents the best basis, that is the most discriminating basis, available in a binary-tree dictionary of wavelet packet bases or local trigonometric bases. $\mathcal{F}_K$ is a *feature selector* which selects the $K < n$ most important basis elements from the best basis $\mathbf{\Psi}_{n\times n}$, and c is a classifier. The problem then is to choose $c, \mathcal{F}_K$ and $\mathbf{\Psi}_{n\times n}$ such that the rate of misclassification of the map d is minimized on the set $\mathcal{X}$. In [Sai94], $\mathbf{\Psi}_{n\times n}$ is taken to be

$$\mathbf{\Psi}_{n\times n} = \arg \max_{B_k\in\mathcal{D}_i\in\mathcal{L}} \lambda(B_k), \tag{2}$$

where $\mathcal{L} = \cup_i\mathcal{D}_i$ is the *library* of all dictionaries at our disposal corresponding to the different wavelet or local trigonometric basis functions under consideration,

the B_k are all bases in $\mathcal{D}_i$, and λ is a measure of performance of the basis B_k in the classification problem. Such a measure is called a discrimination measure. The search for this $\mathbf{\Psi}_{n\times n}$ is fast by the *best-basis-algorithm*, see [CW92], if the measure λ satisfies an additivity property defined as follows: Let V_1 and V_2 be any two disjoint mutually orthogonal subspaces of $\mathbf{R}^n$. Let B_1 and B_2 be any two orthogonal bases of V_1 and V_2, respectively. Then

$$\lambda(B_1 \oplus B_2) = \lambda(B_1) + \lambda(B_2). \tag{3}$$

In [Sai94] the discrimination measure λ is defined as

$$\lambda(B_k) = \sum_{\mathbf{w}_m \in B_k} \gamma(\Gamma^{(1)}(\mathbf{w}_m), ..., \Gamma^{(N)}(\mathbf{w}_m)), \tag{4}$$

where the *time-frequency energy-map* $\Gamma^{(y)}$ is defined by

$$\Gamma^{(y)}(\mathbf{w}_m) = \frac{\sum_{j=1}^{J_y} (\mathbf{w}_m \cdot \mathbf{x}_j^{(y)})^2}{\sum_{j=1}^{J_y} \|\mathbf{x}_j^{(y)}\|^2},$$

$$\mathbf{x}_j^{(y)} \in \mathcal{X}^{(y)},\ 1 \le y \le N,\ J_y = |\mathcal{X}^{(y)}|, \tag{5}$$

and γ can be some form of l^p- distance or relative entropy.

The signals $\mathbf{x}^{(y)} \in \mathcal{X}^{(y)},\ 1 \le y \le N$ are fed into each dictionary as given by (5) and the best basis is selected by the "The Local Discriminant Basis Selection Algorithm",(The LDB algorithm) constructed in [Sai94], and stated in Algorithm 1.1 below. Then the best K basis elements are selected from this basis, ordinarily by selecting the K elements $\mathbf{w}_m$ where γ in (4) takes on its K greatest values. These elements, called "the most discriminating" basis elements, are sorted in decreasing order of importance after decreasing values of γ. They are then used to construct a classifier by doing a "Linear Discriminant Analysis" (LDA) or a "Classification and Regression Trees" (CART)-analysis, or some other statistical classification technique, on the coordinates of the signals in these K best basis elements.

To state the LDB-algorithm of Saito and Coifman, we need some notation. Denote $\mathbf{R}^n$ by $\Omega_{0,0}$ and let $\{\Omega_{j,k}\}_{1\le j\le L, 0\le k\le 2^j-1}$ be a collection of mutually orthogonal subspaces of $\Omega_{0,0}$ with the property that

$$\Omega_{j,k} = \Omega_{j+1,2k} \oplus \Omega_{j+1,2k+1}$$
$$\text{for } j = 0, 1, ..., L \le \log_2 n,\ k = 0, 1, ...2^j - 1. \tag{6}$$

The number L is the depth of the decomposition. It is easy to see that this definition leads to a binary tree structured decomposition of $\mathbf{R}^n$ into mutually orthogonal subspaces $\Omega_{j,k}$ which is fulfilled by both orthonormal wavelet packet bases and local cosine bases. Let $B_{j,k}$ be a set of orthonormal basis vectors for the subspace $\Omega_{j,k}$ and let $A_{j,k}$ denote the restriction of the Local Discriminant Basis to the span of $B_{j,k}$. $\Delta_{j,k}$ is a dummy variable for book-keeping of the maximum value of the discrimination measures of the several possible choices of $B_{j,k}$.

Algorithm 1.1. *The Local Discriminant Basis Selection Algorithm, (the LDB-algorithm). Given a training dataset τ consisting of N classes of signals $\{\{\mathbf{x}_i^{(y)}\}_{i=1}^{J_y}\}_{y=1}^{N}$,*

Step 0:
Choose a dictionary $\mathcal{D}$ of orthonormal wavelet packets or local sine or cosine packets and specify the maximum depth L of decomposition and a measure γ in (4).

Step 1:
Construct time-frequency energy maps $\Gamma^{(y)}$ for y = 1,...,N.

Step 2:
Set $A_{L,k} = B_{L,k}$ and $\Delta_{L,k} = \lambda(B_{L,k})$ for $k = 0, ..., 2^L - 1$, with $\lambda(B_{j,k})$ as defined in (4).

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

Set $\Delta_{j,k} = \lambda(B_{j,k})$.

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.

Step 5:
Use $k \leq n$ most discriminating basis functions for constructing classifiers.

Proposition 1.1. *The basis obtained by the LDB-algorithm maximizes the additive discriminant measure λ on the time-frequency energy maps $\{\Gamma^{(y)}\}_{y=1}^N$ among all the bases in $\mathcal{D}$ obtainable by the LDB-algorithm.*

We refer to [Sai94] for the proof of this proposition.

2. A generalized LDB method

2.1. **New Discrimination Measures.** Using the notation from the previous section, for each basis vector $\mathbf{w}_m$ in some basis B_k, let $Z_{y,m}$ be random variable on the space $\mathcal{X}^{(y)}$ of input signals of class y defined by

$$Z_{y,m} : \mathbf{x} \in \mathcal{X}^{(y)} \to [-1, 1],\ Z_{y,m}(\mathbf{x}) = \mathbf{w}_m \cdot \frac{\mathbf{x}}{\|\mathbf{x}\|_2}. \tag{7}$$

In [CS96] one estimates the empirical probability density function (pdf) p of $Z_{y,m}$. These estimates are then used to find the most discriminating basis. But getting good estimates of the pdf's is hard and computationally demanding. We will take a different approach and work on the *a priori* assumption that p is the uniform distribution. For each fixed $\mathbf{w}_m \in B_k$, we can then compute the *empirical expectation*

$E[Z_{y,m}]$ of the basis coordinate $\mathbf{w}_m \cdot \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$ for class y signals as

$$\begin{aligned} E[Z_{y,m}] &= \sum_{\mathcal{X}^{(y)}} p(Z_{y,m}|Y=y) Z_{y,m} \\ &= \sum_{\mathbf{x}\in\mathcal{X}^{(y)}} \frac{1}{|\mathcal{X}^{(y)}|} \left(\mathbf{w}_m \cdot \frac{\mathbf{x}}{\|\mathbf{x}\|_2}\right). \end{aligned} \tag{8}$$

If $\|\mathbf{x}\|_2 = 1, \forall \mathbf{x} \in \mathcal{X}$, then in this probabilistic setting, (5) is equivalent to $\Gamma^{(y)}(\mathbf{w}_m) = E[Z^2_{y,m}]$. We will first consider two-class problems: $\mathcal{Y} = \{1, 2\}$, and deal with N-class problems later. Choosing $\gamma = \ell^2 - \textit{distance squared}$, (4) becomes

$$\lambda(B_k) = \sum_{m:\mathbf{w}_m \in B_k} \left(E[Z^2_{1,m}] - E[Z^2_{2,m}]\right)^2. \tag{9}$$

We see that with this λ, the best basis given by (2) is the basis maximizing the sum of the euclidean distances between the expected values of all the basis coordinates for the two classes. Now, we observe that the measure of performance (9) of the basis B_k does not consider how the data is distributed around the expected values. For example, if $I_{1,m}$ and $I_{2,m}$ are intervals defined by

$$\begin{aligned} I_{1,m} &= \left[E[Z^2_{1,m}] - \sqrt{Var[Z^2_{1,m}]}, E[Z^2_{1,m}] + \sqrt{Var[Z^2_{1,m}]}\right] \\ I_{2,m} &= \left[E[Z^2_{2,m}] - \sqrt{Var[Z^2_{2,m}]}, E[Z^2_{2,m}] + \sqrt{Var[Z^2_{2,m}]}\right], \end{aligned}$$

where $Var[Z^2_{y,m}]$ is empirical variance of $Z^2_{y,m}$, then it may well happen that $I_{1,m} \cap I_{2,m} \neq \emptyset$, even if $\mathbf{w}_m \in \arg\max_{B_k \in \mathcal{D}_i \in \mathcal{L}} \lambda(B_k)$.

Ideally, we want a basis B where the overlap $\mathcal{O}(B)$ given by

$$\mathcal{O}(B) = \sum_{m:\mathbf{w}_m \in B} |I_{1,m} \cap I_{2,m}|$$

is as small as possible. That is a basis which simultanously is discriminating *between* classes and has the opposite property *inside* classes. This motivates the following definition of a new discrimination measure λ' by

$$\lambda'(B_k) = \sum_{m:\mathbf{w}_m \in B_k} \left[\frac{E[Z^2_{1,m}] - E[Z^2_{2,m}]}{(Var[Z^2_{1,m}] + Var[Z^2_{2,m}])^{1/2}}\right]^2. \tag{10}$$

Note how the performance measure in (10) derives from the measure in (9). We see that the numerator in (10) measures the separability of datapoints *between* the classes $1, 2$, and the denominator measures the dispersion of the datapoints *inside* each of the classes $1, 2$. Neither of the measures λ', λ captures differences between classes in sign in the basis coordinates. To improve on this fact, we define the measure λ'' by

$$\begin{aligned} \lambda''(B_k) = \sum_{m:\mathbf{w}_m \in B_k} & \Big[E[(Z_{1,m} - Z_{2,m})^2]^{1/2} / \\ & \Big(E[(Z_{1,m}(\mathbf{x}) - Z_{1,m}(\mathbf{x}'))^2]^{1/2} + \\ & E[(Z_{2,m}(\mathbf{x}'') - Z_{2,m}(\mathbf{x}'''))^2]^{1/2}\Big)\Big], \\ & \mathbf{x} \neq \mathbf{x}' \in \mathcal{X}^{(1)},\ \mathbf{x}'' \neq \mathbf{x}''' \in \mathcal{X}^{(2)}. \end{aligned} \tag{11}$$

We see that the numerator in (11) measures the separability of *signed* datapoints between the classes $1, 2$, and the denominator measures the dispersion of *signed* datapoints *inside* these classes.

2.2. **Construction of an Oracle Classifier Using Multiple LDB's.** The construction is due to the following observation: Having chosen a best basis $\mathbf{\Psi}_{n\times n}$, where $\mathbf{\Psi}_{n\times n} = \arg\max_{B_k\in\mathcal{D}_i\in\mathcal{L}} \zeta(B_k)$, and ζ is some discrimination measure, there are subsets $\mathcal{S}_j$ of the set $\mathcal{X}$ of input signals on which $\mathbf{\Psi}_{n\times n}$ works better than other subsets. That is, the signals in disjoint sets $\mathcal{S}_j$ have significant differences in how they distribute their energy among the different elements in the basis $\mathbf{\Psi}_{n\times n}$. More precisely: Let $\mathcal{W}_K$ be the *feature space* of dimension $K < n$ spanned by the K most important elements in the best basis $\mathbf{\Psi}_{n\times n}$, sorted in decreasing order of importance, and $P_{\mathcal{W}_K}$ be the orthogonal projection onto $\mathcal{W}_K$.

Now, consider the sets $A^{(k)}$ and $B^{(k)}$ of points in k-dimensional euclidean space given by: $A^{(k)} = \{P_{\mathcal{W}_k}\mathbf{x}_j\}_{\mathbf{x}_j\in\mathcal{X}^{(1)}} \subset [-1,1]^k$, $B^{(k)} = \{P_{\mathcal{W}_k}\mathbf{x}_j\}_{\mathbf{x}_j\in\mathcal{X}^{(2)}} \subset [-1,1]^k$. It is clear by the definition of $\mathcal{W}_k$, that the two point-clouds $A^{(k)}$ and $B^{(k)}$ should be concentrated in more or less disjoint regions in $[-1,1]^k$ if the two classes are separable by our method. That is we should observe clustering when plotting the points of $A^{(k)}$ and $B^{(k)}$ in $[-1,1]^k$ and labeling each point after its class. We sort out clusters by the DCS-algorithm described below.

The DSC-algorithm carries out a classification on the signals in the input signal space $\mathcal{X} = \mathcal{X}^{(1)} \cup \mathcal{X}^{(2)}$ by dividing the set $\mathcal{X}$ into disjoint subsets $\mathcal{S}_j$ and performing a classification on each of these subsets represented in a basis $\mathbf{\Psi}_j$. Each $\mathcal{S}_j$ consists in some sense, determined by restrictions on the minimal size of the sets $\mathcal{S}_j$, of the signals $\mathbf{x}_i \in \mathcal{X}$ on which the most discriminating basis $\mathbf{\Psi}_j$ selected by (2) performs best. Having computed a best basis $\mathbf{\Psi}_1$, the set $\mathcal{S}_1$ is selected first, the signals in $\mathcal{S}_1$ are assigned class names and then $\mathcal{S}_1$ is deleted from the set $\mathcal{X}$. Then a new best basis $\mathbf{\Psi}_2$ for the new $\mathcal{X}$ is computed by the formula (2), the set $\mathcal{S}_2$ is selected, and so on. The algorithm terminates when the set $\mathcal{X}$ has become sparse. Thus, we see that by adapting the parameters we can prevent the algorithm from trying to classify the part of the training dataset which it finds most difficult to classify, and so we gain a smaller overall training-error-rate. But this adjusting of parameters has to be done carefully, so that the algorithm does not fail to catch important features of the signals, neither should it find features that are too adapted to the specific training dataset.

The DCS-algorithm selects the subsets $\mathcal{S}_j$ using as few features as possible, starting with the most discriminating basis-element only. Then, given some upper limit on the rate of error allowed in the clusters, if no clean clustering is observed in the feature space of this single feature element, the algorithm adds information by taking into consideration also the second best feature element and looks for clustering in the feature space spanned by the two best feature elements and so on. If no sufficiently clean clustering is observed using all K best feature elements, the upper error limit is increased and the feature space of the one most important feature element is again searched for clusters, and so on. Using as few features as possible reduces the risk of overtraining of the algorithm, that is the algorithm selecting features that are too adapted to the specific set $\mathcal{X}$ of training data.

On the other hand, we see that the DCS-algorithm is flexible in its selection of relevant features in that it constructs a sequence of feature extractors $\{\mathbf{\Psi}_j\}$ where each $\mathbf{\Psi}_j$ is specially adapted to some part $\mathcal{S}_j$ of the dataset $\mathcal{X}$. The output of the

algorithm is a sequence $\mathcal{C} = \{C_i\}_{i=1}^{L}$ of dyadic hypercubes $C_i \subset [-1,1]^K$ of possibly different dimensions $k_i, 1 \leq k_i \leq K,\ 1 \leq i \leq L$, where to each cube C_i corresponds a specific feature space $\mathcal{W}_{k_i}$ as defined above, and a class name y_{C_i} which equals the name of the majority class of the set $\left(\{P_{\mathcal{W}_{k_i}}\mathbf{x}_j\}_{\mathbf{x}_j \in \mathcal{X}}\right) \cap C_i = \{P_{\mathcal{W}_{k_i}}\mathbf{x}_j\}_{\mathbf{x}_j \in \mathcal{S}_i}$ of datapoints in $\mathcal{W}_{k_i}$ that C_i contains. We will call $\mathcal{C}$ a simple two-class *oracle classifier*, or simply oracle, for the two-class problem $d : \mathcal{X}^{(1)} \cup \mathcal{X}^{(2)} \to \mathcal{Y} = \{1,2\}$.

The formal definition of the DCS-algorithm is given below. Here is a description of the input parameters $K, \delta, \eta, \mu, \nu$ governing the search and selection process in this algorithm:

K is the maximum number of independent features allowed, that is the maximum dimension of the feature space. δ is the stepsize in the search grid of minimum error rates. η is the fraction of the total training set that is allowed to remain unclassified. ν is the minimal allowed fraction of the *total* set of training signals that each hyper cube must contain in order to be recognized as significant. μ is the fraction of the set of unclassified training signals that remains in any step of the DCS-algorithm that each hypercube must contain in order to be recognized as significant in this step of the DCS-algorithm.

We note that in the definition of the DCS-algorithm below, the sets $\mathcal{S}_i$ are given by: $\mathcal{S}_i \equiv \left(P_{\mathcal{W}_{k_j}} A^{(k_j)} \ \cup P_{\mathcal{W}_{k_j}} B^{(k_j)}\right) \cap C_i$, where k_j is the best number of features at the j-th iteration and C_i is some dyadic subcube of $[-1,1]^{k_j}$.

Algorithm 2.1. *The Dyadic Cluster Search Algorithm (the DCS-algorithm). Given appropriately chosen numbers:* $n \geq K \geq 1,\ 1 > \delta > 0,\ 1 > \eta \geq 0,\ 1 > \mu \geq \nu > 0$.

Step 0:
Choose a performance measure λ *as in (9),(10) or in (11), or some other favourite measure. Set* $\beta = \lceil \nu \left|\mathcal{X}\right| \rceil,\ \gamma_A = \lceil \eta \left|\mathcal{X}^{(1)}\right| \rceil,\ \gamma_B = \lceil \eta \left|\mathcal{X}^{(2)}\right| \rceil,\ I = [-1,1]$.

Step 1:
Select the best basis (best basis means the "most discriminating" basis) by the formula (2) and select the feature spaces $\mathcal{W}_K$ *by truncating to the* $K < n$ *most important basis elements in the best basis. Compute the sets* $A^{(K)},\ B^{(K)}$ *as defined above. Set* $\Delta = 0.0,\ k = 1,\ C^{(k)} = I^k,\ C^{(k)}_{next} = I^k,\ FoundCluster = 0$.

Step 2:
Set $A = A^{(k)},\ B = B^{(k)},\ C = C^{(k)},\ C_{next} = C^{(k)}_{next}$.

If $|A| \leq \gamma_A$ *and* $|B| \leq \gamma_B$, *terminate the algorithm.*

Else, *compute* $\alpha = \lceil \mu(|A| + |B|) \rceil,\ N_A(C) = |A \cap C|\,,\ N_B(C) = |B \cap C|$.

Step 3:
If $N_A + N_B \geq \max(\alpha, \beta)$, *compute the error rate* $\epsilon = \min(N_A, N_B)/(N_A + N_B)$ *and proceed to* **Step 4**.

Else

If $C_{next} \neq C$*: Set* $C = C_{next}$ *and jump to* **Step 2**.

Else if $C_{next} = C$*:*

If $FoundCluster = 1$*: Jump to* **Step 1**.

Else if $FoundCluster = 0$*:*

If $k < K$*: Set* $k = k + 1$ *and jump to* **Step 2**.

Else if $k = K$*: Set* $k = 1$, $\Delta = \Delta + \delta$ *and jump to* **Step 2**.

Step 4:
If $\epsilon \leq \Delta$*, store the location of the cube* C *together with the numbers* N_A, N_B *and identification of the* k *basis elements defining the space* $\mathcal{W}_k$*. Then, for each index* $i \in \{1, 2, ..., K\}$*, set* $A^{(i)} = A^{(i)} - \left(\bigcup_{\mathbf{x}_j \in A^{(K)}} (P_{\mathcal{W}_i} \mathbf{x}_j)\right) \cap C$, $B^{(i)} = B^{(i)} - \left(\bigcup_{\mathbf{x}_j \in B^{(K)}} (P_{\mathcal{W}_i} \mathbf{x}_j)\right) \cap C$*, and for each* $\mathbf{x}_{j_i} \in \mathcal{X}^{(i)}$ *such that* $P_{\mathcal{W}_k} \mathbf{x}_{j_i} \in C$*, set* $\mathcal{X}^{(i)} = \mathcal{X}^{(i)} - \mathbf{x}_{j_i}$, $i = 1, 2$*. Set* $FoundCluster = 1$, $\Delta = 0.0$, $k = 1$*, and jump to* **Step 2**.

Else, *divide* C *into* 2^k *subcubes* $C_1, ..., C_{2^k}$ *by splitting each of the sidelengths of* C *into two sides of equal length, and for each index* $i = 1, 2, ..., 2^k$*, jump to* **Step 2** *with* $C = C_i$, $C_{next} = C_{i+1}, 1 \leq i < 2^k$, $C_{next} = C_i, i = 2^k$.

2.3. **On Using and Choosing Oracle Classifiers.** Given a two-class problem $d : \mathcal{X}^{(1)} \cup \mathcal{X}^{(2)} \to \mathcal{Y} = \{1, 2\}$, we compute $\mathcal{C} = \{C_j\}_{j=1}^{L}$ by the DCS-algorithm. Then, given a sample $\mathbf{x} \in \mathcal{T}$, where $\mathcal{T}$ is a test dataset, we assign $\mathbf{x}$ to a class by the following procedure: We check if: $P_{\mathcal{W}_{k_j}} \mathbf{x} \in C_j$, starting with index $j = 1$ and continue until we, if possible, get a positive answer for some index $j' \leq L$. We then assign a weighted class $y_{C_{j'}}$-vote to $\mathbf{x}$ by computing the product of $1 - \epsilon_{j'}$, where $\epsilon_{j'}$ is the error rate of $C_{j'}$, and its statistical frequency $(N_A(C_{j'}) + N_B(C_{j'}))/|\mathcal{X}|$. If $P_{\mathcal{W}_{k_j}} \mathbf{x} \notin C_j$, $\forall C_j \in \mathcal{C}$, we consider the class of $\mathbf{x}$ undetermined.

Different choices of discrimination measure or different settings of the parameters in the DCS-algorithm result in different classifiers. For a two-class problem, we can construct several classifiers by using different performance measures/parameters, and let the weighted majority vote of the classifiers decide whether a sample $\mathbf{x} \in \mathcal{T}$ is of class 1 or class 2. For a N-class problem, $N > 2$, we will apply the method of splitting the N-class problem into N two-class problems: $d : \mathcal{X} \to \{i, 0\}$, $1 \leq i \leq N$, as proposed in [CS96], by splitting the training data set into two sets of class i and not i. One then constructs oracles for each two-class problem. To classify an unknown sample $\mathbf{x} \in \mathcal{T}$, we compute weighted class votes as explained above for the set of oracles and assign $\mathbf{x}$ to the majority vote class.

3. Experimental Results

In some of the calls to the DCS-algorithm in the experiments described below we allowed the algorithm to select a best basis only once, we call this method a LDB-method (Local Discriminant Basis-method). In the cases were we allowed the

algorithm to select multiple different best bases in sequence, we call the method a MLDB-method (Multiple Local Discriminant Basis-method). In the cases where we organized the classifiers resulting from different calls (calls with different discrimination measures) to the DCS-algorithm into a classifier by taking the majority vote over these classifiers, we call the method a *superposition* LDB or MLDB-method, denoted SLDB or SMLDB-method, respectively. In all the three examples below we generated 10 independent realizations of both the training dataset and the test dataset. The results shown in Table 1, Table 2, Table 3 are the mean over the 10 simulations corresponding to the 10 independent realizations of the datasets. The classification rate is defined as the ratio between the number of signals that the DCSA was able to assign a class label and the number of signals that the DCSA was unable to classify. The absolute quality of the results we achieve in Experiment 1 and Experiment 2, is difficult to estimate since we have not been able to calculate the Bayes errors for these classification problems. But these examples will, as we will see below, serve us well in comparing the performance of the different discrimination measures λ, λ' and λ''.

3.1. **Experiment 1.** We consider a two class waveform classification problem as presented in [Fos97]. We generated sets of 100 training signals and 1000 test signals of length 1024 for each class by the formula

$$Q_m(R,\theta,t) = \\ C(R,t)\sum_{j=1}^{m} A_m(j)e^{ik(\frac{r_j^2}{2R}-r_j\cos(\theta-\theta_j))}, \tag{12}$$

where we have:

$C(R,t) = \dfrac{e^{-ik(ct-R)}}{R}$ is considered constant $= 1$
for simplicity.
$R = 10^4$.
$k = 100$.
$A_m(j) = \dfrac{1}{m}$.
r_j is random variable uniformly distributed on $[1, 10]$.
θ_j is random variable uniformly distributed on
$[2\pi\dfrac{j}{m}, 2\pi\dfrac{j}{m} + \dfrac{\pi}{4}]$.

For each m-tiple of realizations $\{r_j, \theta_j\}_{j=1}^m$ of the pair of random variables r_j, θ_j, we generate a discrete signal $S_m(\theta)$ by uniformly sampling the real part of $Q_m(R,\theta,t)$ 1024 times in the variable θ with sampling density $2\pi/16 \cdot k = 2\pi/1600$. We generated data sets by extracting realizations of $S_m/\|S_m\|_2$ *smoothly* from a fixed sampling interval. In this problem we used $m = 3,\ m = 4$ in (12) to define two classes of signals and the coiflet with filterlength 18 as dictionary. All calls to the DCS-algorithm in this experiment were made with $K = 5,\ \delta = 0.01,\ \eta = 0.05,\ \mu = 0.10,\ \nu = 0.05$. The results are shown in Table 1.

Method	Classification rate (%)				Error rate (%)			
	Training data		Test data		Training data		Test data	
	Total	σ	Total	σ	Total	σ	Total	σ
$\text{LDB}_{\lambda'}$	99.7	0.7	99.5	0.9	19.9	3.0	29.8	2.8
$\text{MLDB}_{\lambda'}$	98.5	1.4	98.4	1.4	8.6	1.1	23.5	2.7
$\text{LDB}_{\lambda''}$	97.6	2.0	96.3	3.6	16.0	4.5	23.5	4.4
$\text{MLDB}_{\lambda''}$	95.2	1.9	93.8	2.3	12.9	3.1	23.7	3.4
LDB_{λ}	98.9	1.7	98.8	2.1	16.6	4.6	24.6	4.9
MLDB_{λ}	98.4	1.1	99.5	0.6	13.7	3.1	24.4	1.9
SLDB	100	0.0	100	0.0	14.7	3.9	22.2	1.5
SMLDB	100	0.0	100	0.0	9.1	3.3	**20.4**	2.0

TABLE 1. The average classification rates and the corresponding error rates over 10 simulations from Experiment 1. The index λ in LDB_{λ} and MLDB_{λ} defines the discriminant measure used in the DCS-algorithm to select the LDB or MLDB. SLDB is the superposition of methods $\text{LDB}_{\lambda'}$, $\text{LDB}_{\lambda''}$, LDB_{λ}. SMLDB is the superposition of the methods $\text{MLDB}_{\lambda'}$, $\text{MLDB}_{\lambda''}$, MLDB_{λ}. σ is the square root of the sample variance.

3.2. **Comments on Experiment 1.** In this experiment we achieved the best result by the superposition method using multiple LDB's, denoted SMLDB. We see that the generalized methods MLDB1, MLDB2, MLDB3 are almost indistinguishable in this example. We conclude that our new measures λ', λ'' hardly yield a significantly better classification than the original measure λ, the positive effect is in any case small. Furthermore, for the measure λ' we do get better results by the generalized method, whereas for the measures λ'' and λ the positive effect of generalizing is more doubtful. All in all, it seems we are a little better off with either measure λ', λ'' than the original λ. It is difficult to decide the absolute quality of these results, since we have not been able to calculate the Bayes error for this problem.

3.3. **Experiment 2.** This example is identical to Experiment 1 except that we used $m = 4,\ m = 5$ in (12) to define the two signal classes. We used the coiflet with filterlength 18 as dictionary. All calls to the DCS-algorithm in this experiment were made with $K = 5,\ \delta = 0.01,\ \eta = 0.05,\ \mu = 0.10,\ \nu = 0.05$. The results are shown in Table 2.

3.4. **Comments on Experiment 2.** In this experiment we achieved the best result with the method LDB2. We see that both discrimination measures λ', λ'' clearly outperform the original measure λ in this problem. As in Experiment 1, the measures λ' and λ'' yield about the same results with MLDB-methods. When not taking superpositions of several classifiers, the generalised MLDB-method does not yield any improvements in results on test data, rather it seems that this method adapts too much to training data in this example. Furthermore, due to the poor performance of the measure λ in this example, we get worse results with superposition methods in this example than when using the best single classifier. But we could expect to further lower the best error rate on test data by combining classifiers from the measures λ', λ'' only.

Method	Classification rate (%)				Error rate (%)			
	Training data		Test data		Training data		Test data	
	Total	σ	Total	σ	Total	σ	Total	σ
$\text{LDB}_{\lambda'}$	99.3	1.4	98.7	2.3	11.9	3.5	19.9	5.4
$\text{MLDB}_{\lambda'}$	98.0	1.4	97.3	2.3	6.4	2.9	20.5	4.3
$\text{LDB}_{\lambda''}$	96.9	2.6	96.9	2.5	10.5	2.7	**17.5**	3.2
$\text{MLDB}_{\lambda''}$	96.0	2.2	94.1	3.6	9.5	1.9	19.0	2.6
LDB_{λ}	99.1	1.7	99.6	0.9	24.2	5.6	32.6	7.9
MLDB_{λ}	98.5	1.4	99.5	0.5	21.6	3.6	35.8	4.2
SLDB	100	0.0	100	0.0	15.0	4.9	21.8	4.5
SMLDB	100	0.0	100	0.0	8.4	3.7	20.1	3.2

TABLE 2. The average classification rates and the corresponding error rates over 10 simulations from Experiment 2. The index λ in LDB_{λ} and MLDB_{λ} defines the discriminant measure used in the DCS-algorithm to select the LDB or MLDB. SLDB is the superposition of methods $\text{LDB}_{\lambda'}$, $\text{LDB}_{\lambda''}$, LDB_{λ}. SMLDB is the superposition of the methods $\text{MLDB}_{\lambda'}$, $\text{MLDB}_{\lambda''}$, MLDB_{λ}. σ is the square root of the sample variance.

Why the measure λ performs so badly in this experiment compared to the other two measures λ', λ'', we are not able to fully explain yet. But it seems like the considerations we made that led us from the measure λ to the construction of the measures λ' and λ'' come into play for this particular problem. That is, not only the euclidean distance between the Expected values of a particular basis coordinate for two different classes of signals is important to discriminate these classes well in this particular coordinate, but also *how* the data is distributed around the expected values of this coordinate. And as we have seen, the measure λ only looks for coordinates which maximizes the euclidean distance between expectations, while the measures λ' and λ'' try to find the coordinate which maximize the distance between expectations while minimizing the variance of these expectations.

However, we believe that being able to explain the qualitative difference between results in Experiment 1 and Experiment 2 is important in order to understand better the workings of the DCS-algorithm. This will be a future project.

3.5. **Experiment 3.** We consider a three class waveform classification problem as presented in [Sai94]. We generated sets of 100 training signals and 1000 test signals of length 32 for each class by first extracting signal samples by the formulas

$$\begin{aligned} f_1(i) &= uh_1(i) + (1-u)h_2(i) + \epsilon(i) \text{ for Class 1} \\ f_2(i) &= uh_1(i) + (1-u)h_3(i) + \epsilon(i) \text{ for Class 2} \\ f_3(i) &= uh_2(i) + (1-u)h_3(i) + \epsilon(i) \text{ for Class 3,} \end{aligned}$$

where $i = 1, ..., 32$, $h_1(i) = \max(6 - |i-7|, 0)$, $h_2(i) = h_1(i-8)$, $h_3(i) = h_1(i-4)$, u is a uniform random variable on the interval $(0, 1)$, and $\epsilon(i)$ are the standard normal variates. We then normalized the signals in the energy norm by setting $f_1(i) = f_1(i)/\|f_1\|_2$, $f_2(i) = f_2(i)/\|f_2\|_2$, $f_3(i) = f_3(i)/\|f_3\|_2$, $i = 1, ..., 32$. We used the coiflet with filterlength 6 as a dictionary for this problem. All calls to

the DCSA in this experiment were made with $K = 5,\ \delta = 0.01,\ \eta = 0.05,\ \mu = 0.20,\ \nu = 0.05$. The results are shown in Table 3.

Method	Classification rate (%)				Error rate (%)			
	Training data		Test data		Training data		Test data	
	Total	σ	Total	σ	Total	σ	Total	σ
$\text{LDB}_{\lambda'}$	100	0.0	100	0.0	23.7	1.9	28.2	0.8
$\text{MLDB}_{\lambda'}$	100	0.0	100	0.0	22.9	2.1	28.5	2.7
$\text{LDB}_{\lambda''}$	100	0.0	100	0.0	23.6	2.4	27.9	1.9
$\text{MLDB}_{\lambda''}$	100	0.0	100	0.0	20.7	2.7	27.7	1.5
LDB_{λ}	100	0.0	100	0.0	25.0	2.6	29.1	1.8
MLDB_{λ}	100	0.0	100	0.0	23.0	2.6	26.2	2.6
SLDB	100	0.0	100	0.0	18.7	1.9	22.7	0.9
SMLDB	100	0.0	100	0.0	15.7	1.9	**20.5**	1.0

TABLE 3. The average classification rates and the corresponding error rates over 10 simulations from Experiment 3. The index λ in LDB_{λ} and MLDB_{λ} defines the discriminant measure used in the DCS-algorithm to select the LDB or MLDB. SLDB is the superposition of methods $\text{LDB}_{\lambda'}$, $\text{LDB}_{\lambda''}$, LDB_{λ}. SMLDB is the superposition of the methods $\text{MLDB}_{\lambda'}$, $\text{MLDB}_{\lambda''}$, MLDB_{λ}. σ is the square root of the sample variance.

3.6. **Comments on Experiment 3.** In this experiment we achieved the best result by the method SMLDB, and we see that superposition methods are clearly favourable in this case. However, it seems to make little difference which measure we are using when not taking superpositions of several classifiers. We remark that both the measures $\lambda',\ \lambda''$ select the standard basis as the most discriminating basis in the first iterations in the DCS-algorithm, whereas λ does not choose this basis at any iteration. Saito and Coifman in [CS96] get a best error rate of 15.9% which is considerably closer to the estimated Bayes error of $\approx 14\%$ for this problem, than our best result of 20.5%. We believe the reason for this lies in our choice of classifier, which is a special case of the classification tree (CT) classifier, in that we restrict to only dyadic partitions of the coordinate axes. In [CS96], the best result of 15.9% is achieved with a linear discriminant analysis-classifier (LDA), while the best result achieved with a CT classifier was 20.8%, which is close to our best result of 20.5%. We would like to test the performance of our proposed discriminant measures λ', λ'' on this particular triangualr waveform classification problem with a LDA-classifier in the future.

3.7. **Conclusion.** We have shown that constructing classifiers by estimating expectations and variances directly from the expansion coefficients of the datasets in the binary-tree structured dictionary of bases may yield better classifiers than classifers constructed from the energy-density dictionaries of bases. Also, we have shown that comparing/combining different discrimination measures in classification problems may lead to significant improvements in the success of the classification methods.

Appendix A. Software and Hardware

All algorithms and transforms used in the numerical experiments, except some of the random number generators described below, were implemented in the computer language C++ and compiled with the GNU project C++ compiler on a HP K260 machine with a PA 8000 processor. We used double-precision in all calculations.

In the experiments we used the Fortran NAG-routines G05DAF, G05FAF for generating random numbers with uniform distribution, and G05FDF for generating random numbers with standard normal distribution.

Appendix B. Plots of data from Experiment 1

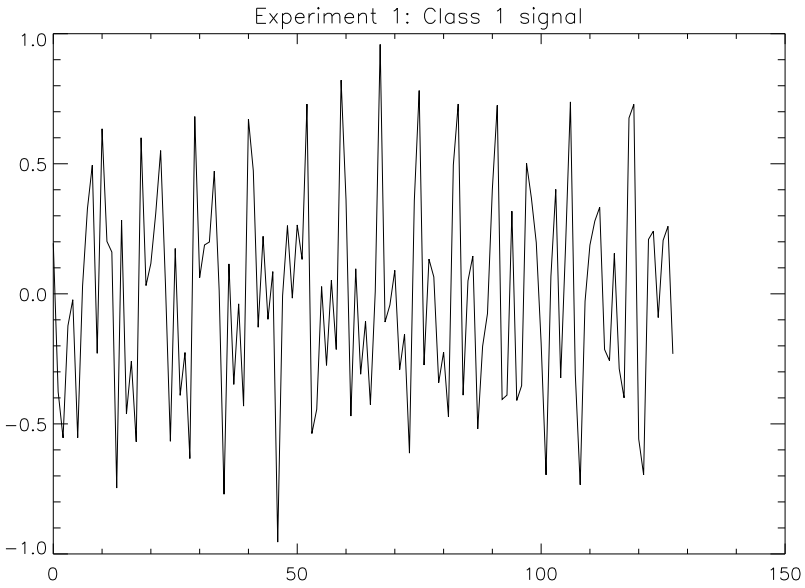

Figure 1. Plot of the first 128 samples of a $S_3(\theta)$ signal in Experiment 1.

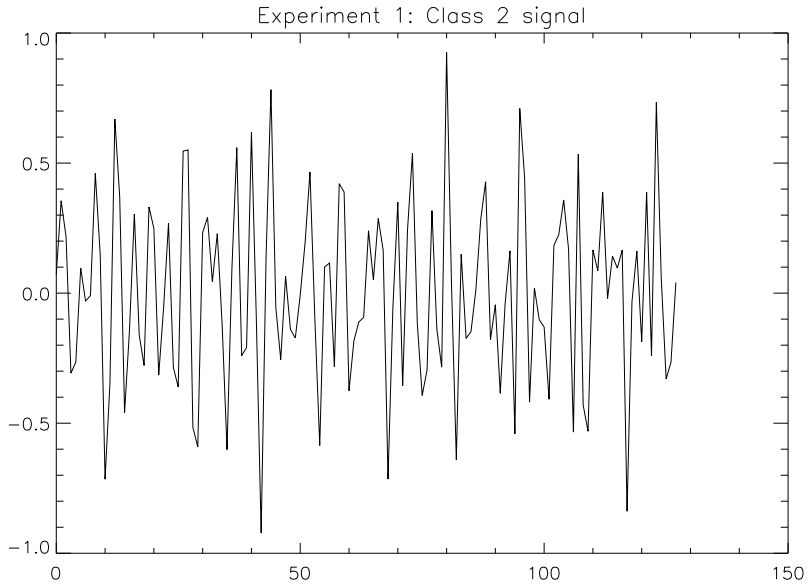

FIGURE 2. Plot of the first 128 samples of a $S_4(\theta)$ signal in Experiment 1.

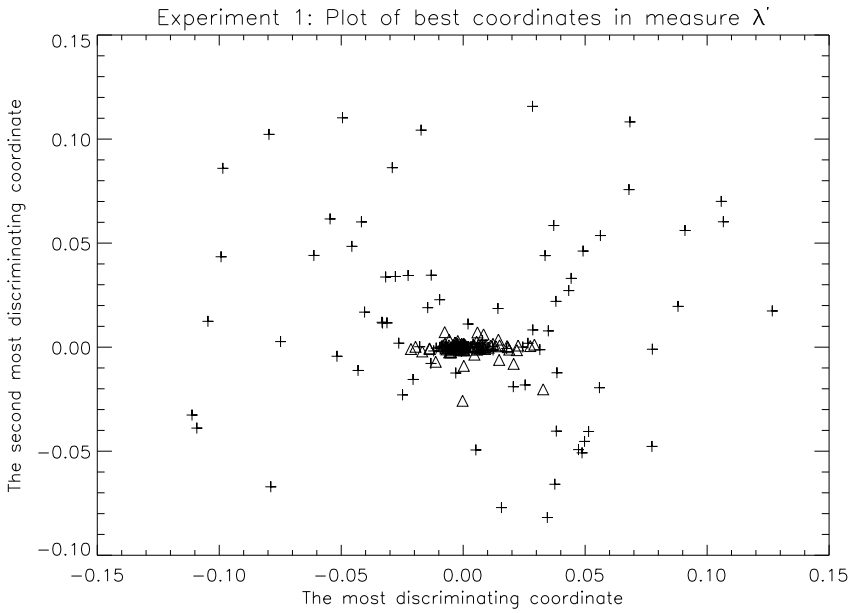

FIGURE 3. A cluster plot of a training dataset in Experiment 1. This plot shows the two coordinates of each signal in the training dataset corresponding to the two most discriminating basis elements in the most discriminating basis *first selected* by the DCSA when using the measure λ'. Crosses and triangles denote datapoints corresponding to S_3 signals and S_4 signals, respectively.

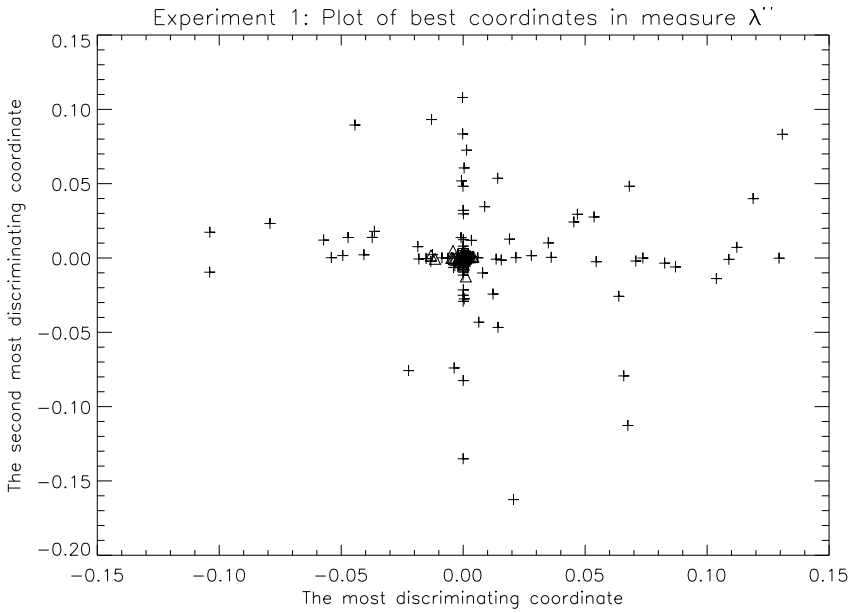

FIGURE 4. A cluster plot of a training dataset in Experiment 1. This plot shows the two coordinates of each signal in the training dataset corresponding to the two most discriminating basis elements in the most discriminating basis *first selected* by the DCSA when using the measure λ''. Crosses and triangles denote datapoints corresponding to S_3 signals and S_4 signals, respectively.

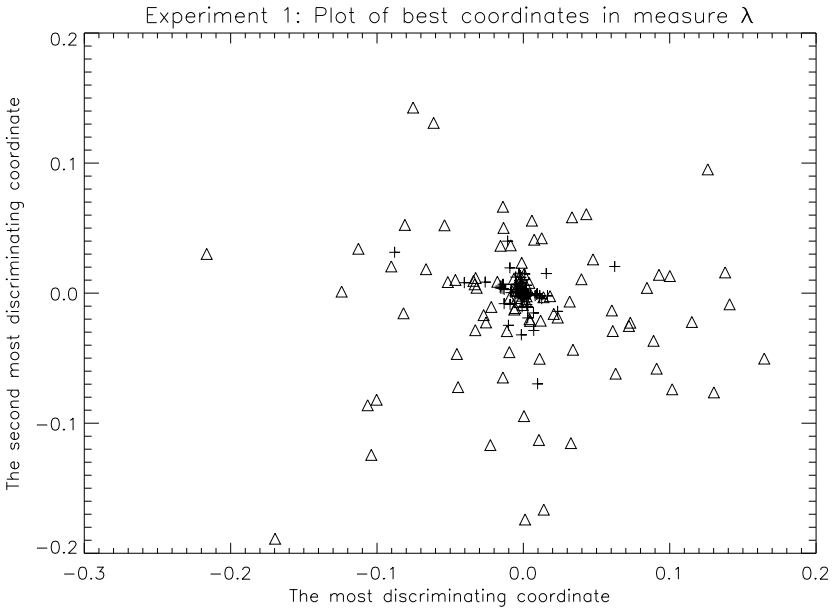

FIGURE 5. A cluster plot of a training dataset in Experiment 1. This plot shows the two coordinates of each signal in the training dataset corresponding to the two most discriminating basis elements in the most discriminating basis *first selected* by the DCSA when using the measure λ. Crosses and triangles denote datapoints corresponding to S_3 signals and S_4 signals, respectively.

Appendix C. Plots of data from Experiment 2

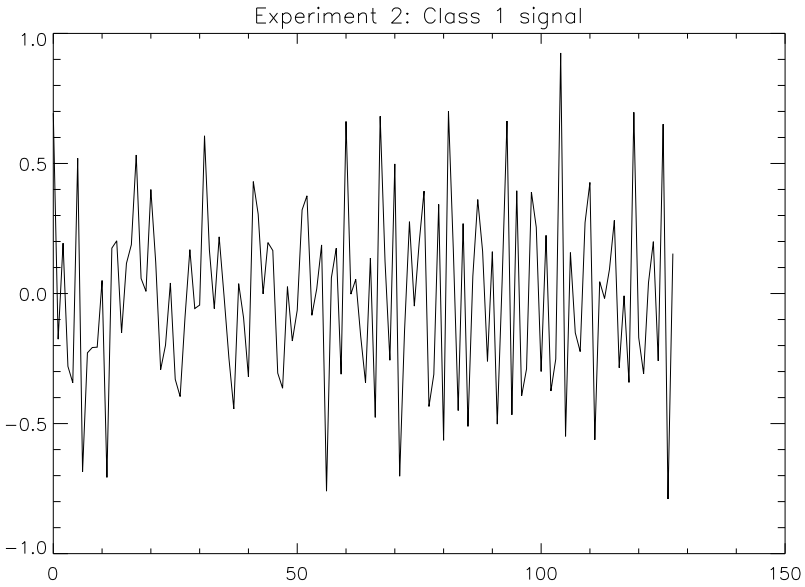

Figure 6. Plot of the first 128 samples of a $S_4(\theta)$ signal in Experiment 2.

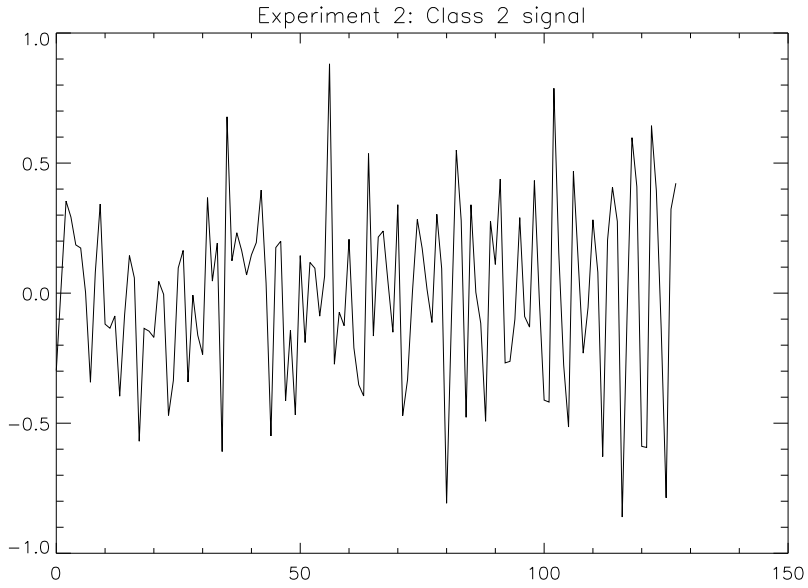

FIGURE 7. Plot of the first 128 samples of a $S_5(\theta)$ signal in Experiment 2.

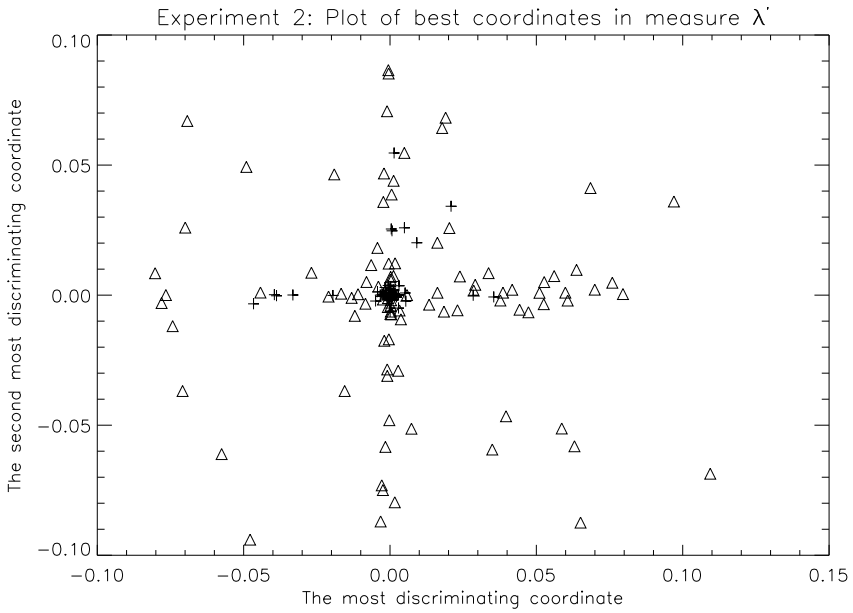

FIGURE 8. A cluster plot of a training dataset in Experiment 2. This plot shows the two coordinates of each signal in the training dataset corresponding to the two most discriminating basis elements in the most discriminating basis *first selected* by the DCSA when using the measure λ'. Crosses and triangles denote datapoints corresponding to S_4 signals and S_5 signals, respectively.

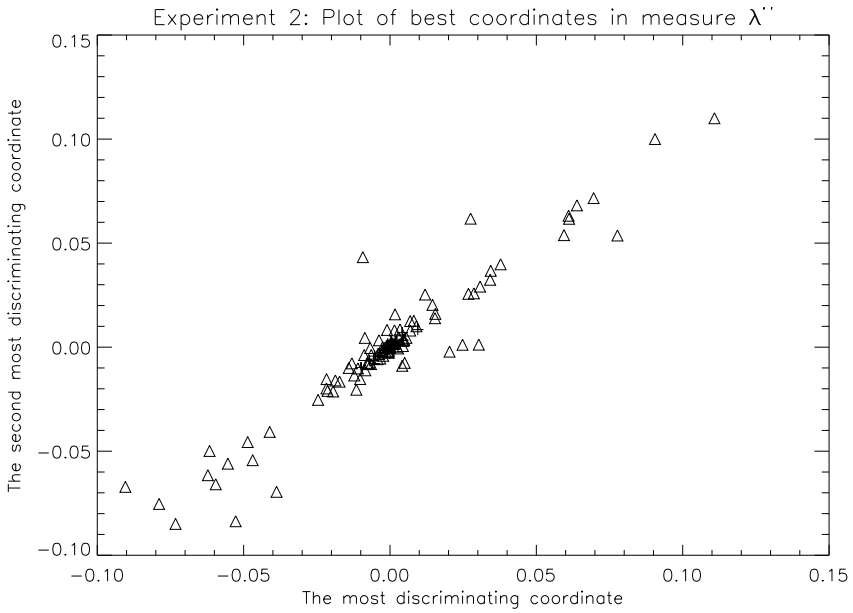

FIGURE 9. A cluster plot of a training dataset in Experiment 2. This plot shows the two coordinates of each signal in the training dataset corresponding to the two most discriminating basis elements in the most discriminating basis *first selected* by the DCSA when using the measure λ''. Crosses and triangles denote datapoints corresponding to S_4 signals and S_5 signals, respectively.

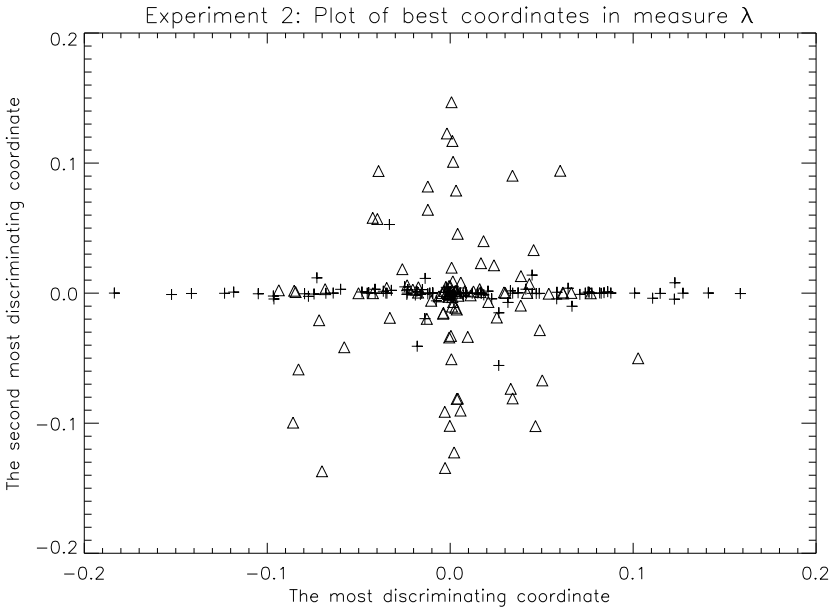

FIGURE 10. A cluster plot of a training dataset in Experiment 2. This plot shows the two coordinates of each signal in the training dataset corresponding to the two most discriminating basis elements in the most discriminating basis *first selected* by the DCSA when using the measure λ. Crosses and triangles denote datapoints corresponding to S_4 signals and S_5 signals, respectively.

Appendix D. Plots of data from Experiment 3

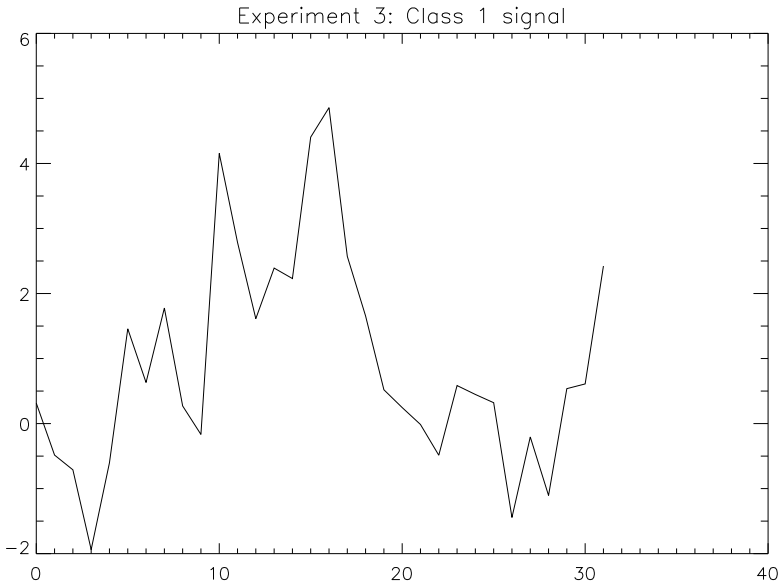

FIGURE 11. Plot of a $f_1(i)$ signal.

FIGURE 12. Plot of a $f_2(i)$ signal.

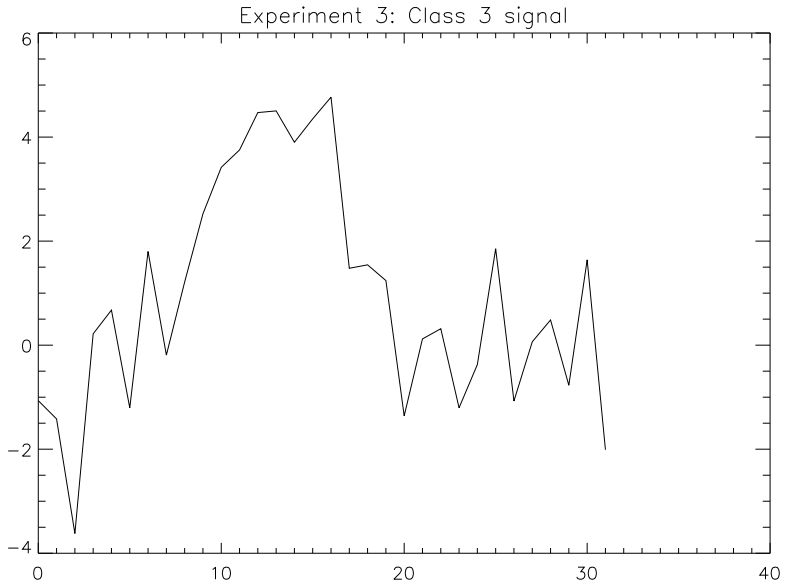

FIGURE 13. Plot of a $f_3(i)$ signal.

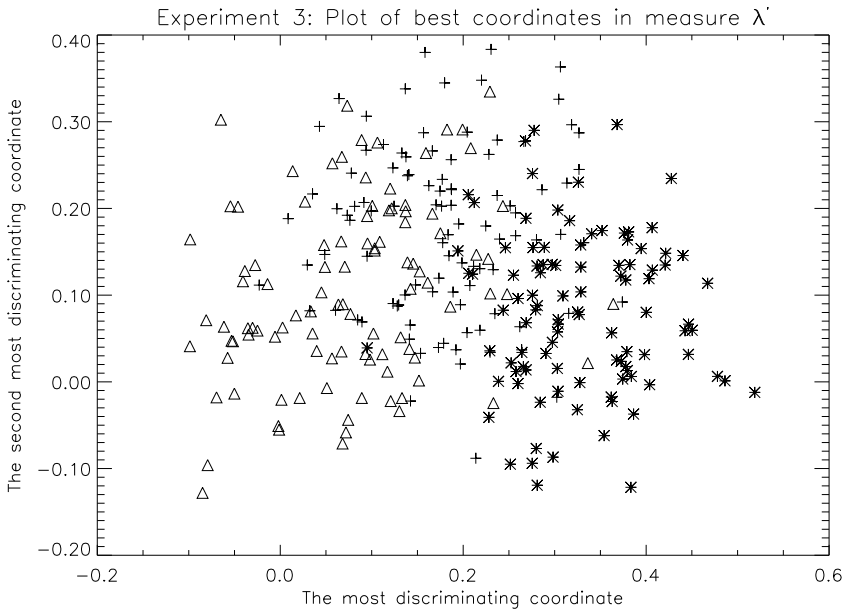

FIGURE 14. A cluster plot of a training dataset in Experiment 3. This plot shows the two coordinates of each signal in the training dataset corresponding to the two most discriminating basis elements in the most discriminating basis *first selected* by the DCSA when using the measure λ'. Crosses, triangles and stars denote datapoints corresponding to f_1 signals, f_2 signals and f_3 signals, respectively.

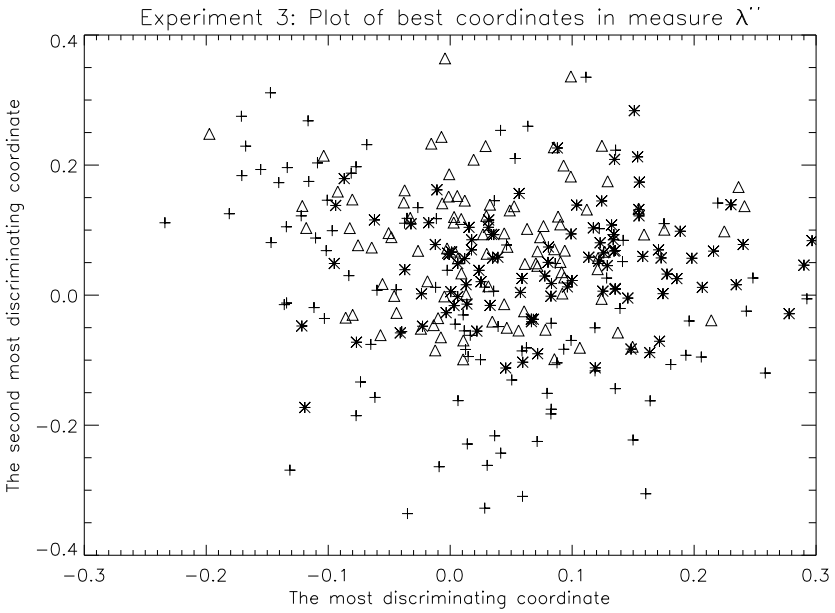

FIGURE 15. A cluster plot of a training dataset in Experiment 3. This plot shows the two coordinates of each signal in the training dataset corresponding to the two most discriminating basis elements in the most discriminating basis *first selected* by the DCSA when using the measure λ''. Crosses, triangles and stars denote datapoints corresponding to f_1 signals, f_2 signals and f_3 signals, respectively.

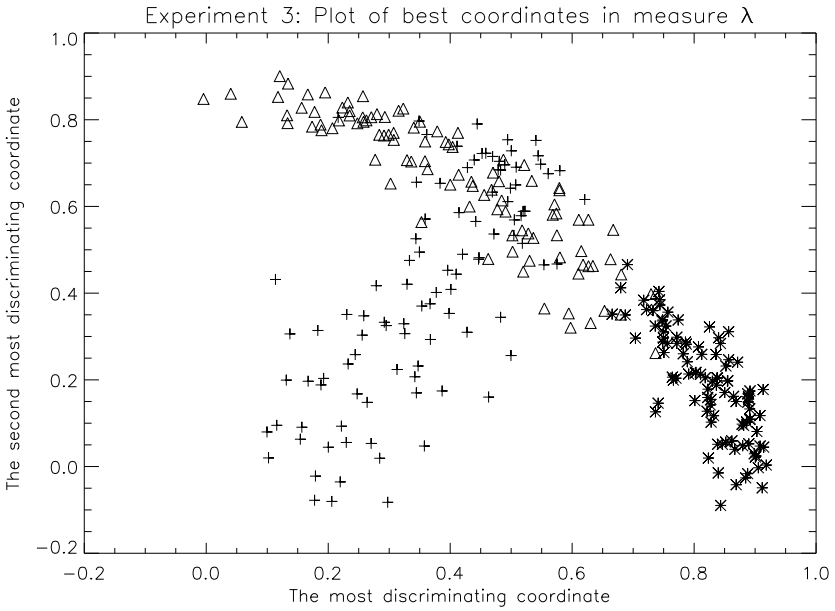

FIGURE 16. A cluster plot of a training dataset in Experiment 3. This plot shows the two coordinates of each signal in the training dataset corresponding to the two most discriminating basis elements in the most discriminating basis *first selected* by the DCSA when using the measure λ. Crosses, triangles and stars denote datapoints corresponding to f_1 signals, f_2 signals and f_3 signals, respectively.

REFERENCES

[CS95] Ronald R. Coifman and Naoki Saito. Local Discriminant Bases and Their Applications . *Journal of Mathematical Imaging and Vision*, 5:713–719, 1995.

[CS96] Ronald R. Coifman and Naoki Saito. Improved Local Discriminant Bases Using Empirical Probability Density Estimation . *Proceedings of Statistical Computing*, 1996.

[CW92] Ronald R. Coifman and Mladen V. Wickerhauser. Entropy-based Algorithms For Best Basis Selection . *IEEE Trans. Inform. Theory*, 38(2):713–719, 1992.

[Fos97] Eirik Fossgaard. Fast Computational Algorithms for the Discrete Wavelet Transform and Applications of Localized Orthonormal Bases in Signal Classification . Technical report, Department of Mathematics, University of Tromsø, 9037 Tromsø, 1997. Available online at `http://xxx.lanl.gov/` under cs.MS/9901008, ACM-class: F.2.1,G.4,I.5.4, and at `http://www.math.uit.no/~eirikf/`.

[Sai94] Naoki Saito. *Local Feature Extraction and Its Applications Using a Library of Bases* . PhD thesis, Yale University, Department of Mathematics, 10 Hillhouse Avenue, P.O. Box 208283 New Haven, CT 06520-8283, 1994. Available online at `http://www.math.yale.edu/pub/papers/`.