Hindawi Publishing Corporation
Advances in Fuzzy Systems
Volume 2012, Article ID 920920, 7 pages
doi:10.1155/2012/920920

Research Article

Classifying High-Dimensional Patterns Using a Fuzzy Logic Discriminant Network

Nick J. Pizzi[1] and Witold Pedrycz[2]

[1] *Department of Computer Science, University of Manitoba, Winnipeg MB, Canada R3T 2N2*
[2] *Department of Electrical and Computer Engineering, University of Alberta, Edmonton AB, Canada T6R 2G7*

Correspondence should be addressed to Nick J. Pizzi, pizzi@cs.umanitoba.ca

Received 28 July 2011; Accepted 8 December 2011

Academic Editor: Maysam Abbod

Although many classification techniques exist to analyze patterns possessing straightforward characteristics, they tend to fail when the ratio of features to patterns is very large. This "curse of dimensionality" is especially prevalent in many complex, voluminous biomedical datasets acquired using the latest spectroscopic modalities. To address this pattern classification issue, we present a technique using an adaptive network of fuzzy logic connectives to combine class boundaries generated by sets of discriminant functions. We empirically evaluate the effectiveness of this classification technique by comparing it against two conventional benchmark approaches, both of which use feature averaging as a preprocessing phase.

1. Introduction

Biomedical spectroscopic modalities produce information-rich but complex, voluminous data [1]. For instance, magnetic resonance spectroscopy, which exploits the interaction between an external homogenous magnetic field and a nucleus that possesses spin, is a reliable and versatile spectroscopic modality [2, 3]. Coupled with robust multivariate discrimination methods, it is especially useful in the interpretation and classification of high-dimensional biomedical spectra (patterns) of tissues and biofluids [4]. However, the ratio of the number of features to the number of patterns for these data is typically very large; the feature space dimensionality is $O(10^3\text{-}10^4)$ while the number of patterns is O(10–100). This "curse of dimensionality" [5, 6] is a serious challenge for the classification of complex biomedical spectra: the excess degrees of freedom tend to cause overfitting, which significantly affects the reliability of the chosen classifier by diminishing its capability to determine effective generalizations.

We present a pattern classification technique, an extension to a method described in [7], that attenuates the confounding effects of the curse of dimensionality using an adaptive network of fuzzy logic connectives to combine pattern class boundaries generated by sets of discriminant functions based on sets of feature regions possessing high discriminatory power. We empirically evaluate the effectiveness of this classification technique by comparing it against two conventional benchmark approaches, both of which use feature averaging as a preprocessing phase.

Section 2 presents a brief discussion on pattern classification including pattern mapping, validation, discriminant analysis, and dimensionality reduction approaches. Details of our technique are presented in Section 3. Datasets, experiment design, and results are discussed in Section 4 followed by some concluding remarks.

2. Biomedical Pattern Classification

2.1. Mappings and Validation. We begin by defining some formal notation to precisely describe the problem of pattern classification where N is the number of patterns (samples, vectors, individuals, or cases), n is the number of features (dimensions, attributes, or measurements), and c is the number of classes (groups). Let $\mathbf{X} = \{(\mathbf{x}_k, \omega_k),\ k = 1, 2, \ldots, N\}$ be a set of N labeled patterns where $\mathbf{x}_k \in \Re^n$ and $\omega_k \in \Omega$. Typically, $\Omega = \{1, 2, \ldots, c\}$; however, it is often advantageous [8] to use 1-of-c encoding for the class labels for iterative

classifiers such as artificial neural networks [2]; namely, $\Omega = \{\gamma_1, \gamma_2, \ldots, \gamma_c\}$, where, for $\mathbf{x}_i$, $\gamma_{\omega_i} = 1$ and $\gamma_{\omega_j} = 0$ $(\forall\, \omega_i \neq \omega_j)$. A classifier is a system that determines a mapping, $f : \mathbf{X} \rightarrow \Omega$. Using f, if a classifier predicts that the class label for $\mathbf{x}_i$ is ω_p, then a correct classification occurs when $\omega_p = \omega_i$. It is considered a misclassification (a classification error) if $\omega_p \neq \omega_i$.

Unfortunately, many investigations involving pattern classification are biased as they use the entire dataset to determine the mapping. This approach leads to overly optimistic pattern classification results and do not take into account the possibility of overfitting; that is, the mapping becomes a simple table lookup between the given patterns and class labels, thereby possessing no generalized predictive power for new (unseen) patterns. To compensate for this bias, it is essential to perform some type of validation [9, 10]. For instance, patterns in $\mathbf{X}$ may be randomly allocated to a design (training) subset, $\mathbf{X}^D$ containing N^D patterns, or a validation (test) subset, $\mathbf{X}^V$ containing N^V patterns ($N^D + N^V = N$). Now, a mapping is determined using only design patterns, $f' : \mathbf{X}^D \rightarrow \Omega$, but the classification performance is measured using f' with the validation patterns.

Classification performance is measured using the $c \times c$ "confusion matrix" of the desired class labels versus the predicted class labels. If the class prediction for $\mathbf{x}_i$ is ω_p, then the element, $[\omega_p, \omega_i]$, of the confusion matrix is incremented by one (perfect accuracy is reflected by zeroes on the off-diagonal and nonzeroes on the diagonal). The conventional performance measure is the ratio of correctly classified patterns to the total number of patterns, $P_O = (\sum_i r_{ii})/N^V (i = 1, 2, \ldots, c)$, where r_{ij} is the number of class i validation patterns predicted, by the mapping f, to belong to class j. While other measures exist, such as the average class-wise accuracy, receiver operating characteristics graphs (ROC curves) [11], or the kappa score (a chance corrected measure of agreement) [12], for the sake of clarity during the discussion of the experiment results, we will use P_O.

2.2. Discriminant Functions. Linear discriminant analysis (LDA) [13] is a conventional classification approach that determines linear boundaries between c classes while taking into account inter class and intra class variances. If the error distributions for the classes are the same (identical covariance matrices), LDA constructs the optimal linear boundary between the classes. In real-world situations, this optimality is seldom achieved since different classes typically give rise to different distributions.

LDA allocates a pattern, $\mathbf{x}$, to class i for which the probability distribution, $p_i(\mathbf{x})$, is greatest. That is, $\mathbf{x}$ is allocated to class i, if $q_i p_i(\mathbf{x}) \geq q_j p_j(\mathbf{x})$ $(\forall\, j = 1, 2, \ldots, c\ [j \neq i])$, where q_i is the class' prior (or proportional) probability. The discriminant function for class i is

$$D_i(\mathbf{x}) = \log q_i + \boldsymbol{\mu}_i^{\mathrm{T}} \mathbf{W}^{-1} \left(x - \frac{1}{2} \boldsymbol{\mu}_i \right), \tag{1}$$

where $\boldsymbol{\mu}_i$ is the mean for class i and $\mathbf{W}$ is the covariance matrix of the patterns in $\mathbf{X}$. The feature space hyperplane separating class i from class j is defined by $F_{ij}(\mathbf{x}) = D_i(\mathbf{x}) - D_j(\mathbf{x}) = 0$. Figure 1 illustrates the class boundaries defined by a set of linear discriminant functions for a two-dimensional dataset with three classes ($N = 33, n = 2, c = 3$). As mentioned in Section 2.2, when LDA is used for pattern classification, it is imperative to define the discriminant functions using the design patterns, $\mathbf{X}^D$, but to validate the performance using the validation patterns, $\mathbf{X}^V$.

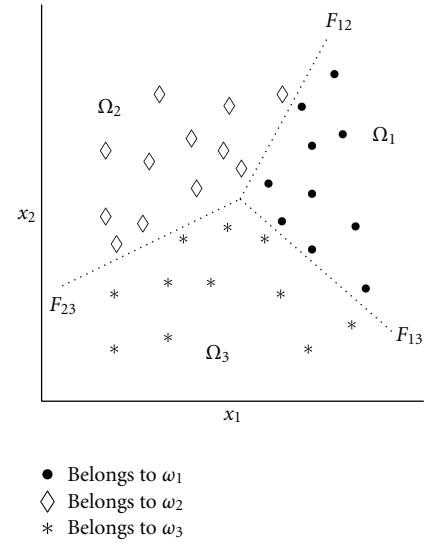

Figure 1: Class boundaries defined by linear discriminant functions for three classes of 2-dimensional patterns.

The support vector machine (SVM) [14, 15] is an important family of supervised learning algorithms that select models that maximize the error margin of a training subset. This approach has been successively used in a wide range of data classification problems [16]. Given a set of patterns that belong to one of two classes, an SVM finds the hyperplane leaving the largest possible fraction of patterns of the same class on the same side while maximizing the distance of either class from the hyperplane. The approach is usually formulated as a constrained optimization problem and solved using constrained quadratic programming. While the original approach [17] could only be used for linearly separable problems, it may be extended by employing a "kernel trick" [18] that exploits the fact that a nonlinear mapping of sufficiently high dimension can project the patterns to a new parameter space in which classes can be separated by a hyperplane. In general, it cannot be determined a priori which kernel will contribute to producing the best classification results for a given dataset, and one must rely on heuristic (trial and error) experimentation. Common kernel functions $K(\mathbf{x}, \mathbf{y})$, for patterns $\mathbf{x}$ and $\mathbf{y}$, are power, $(\mathbf{x} \cdot \mathbf{y})^d$; polynomial, $(a\mathbf{x} \cdot \mathbf{y} + b)^d$; sigmoid, $\tanh(a\mathbf{x} \cdot \mathbf{y} + b)$; Gaussian, $\exp(-0.5|\mathbf{x} - \mathbf{y}|^2/\sigma)$.

2.3. Feature Reduction. As with any pattern classifier, LDA becomes unreliable when there are a large number of features. Even when using stable methods such as singular value decomposition, the inversion of $\mathbf{W}$ in (1) becomes unstable, so it becomes imperative to preprocess the features.

A preprocessing strategy to use when n/N is very large (curse of dimensionality) is to reduce the dimensionality of the feature space of the patterns; that is, we find a mapping (transformation) $f' : \mathbf{X} \rightarrow \mathbf{Y} = \{(\mathbf{y}_k, \omega_k)\}$ where $\mathbf{y}_k \in \Re^m$ and $m \ll n$. Now, the classification mapping becomes $f : \mathbf{Y} \rightarrow \Omega$. A standard approach to feature space reduction is to take the averages of a fixed number of contiguous feature regions. Although this type of averaging may often work well in attenuating the effects of the curse of dimensionality, it also has a tendency to sometimes wash away information content. Other feature reduction approaches do not transform the original feature space but rather attempt to find those features that possess the greatest discriminatory power [19–22]. One example of this type of approach is stochastic feature selection.

2.4. Stochastic Feature Selection. Stochastic feature selection (SFS) [23] is a feature selection/reduction preprocessing strategy that may be used with any homogeneous or heterogeneous set of classifiers (e.g., LDA, artificial neural networks, support vector machines). Essentially, SFS iteratively presents, in a highly parallelized fashion, many feature regions (contiguous subsets of pattern features) to the set of classifiers retaining the best set of classifier/region pairs. While SFS has a rich set of parameters to control many different aspects of the classification process, here we present only those aspects that are relevant to this discussion and refer the reader to [23] for a thorough description of this strategy. For a pattern $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, we define a region to be a contiguous subset of its features, $\mathbf{x}^{rs} = [x_r, x_{r+1}, \ldots, x_s]$ $(1 \leq r \leq s \leq n)$. The user specifies the minimum and maximum number of regions to be selected for each classification iteration as well as the minimum and maximum length for a feature region $(s - r + 1)$. SFS exploits the quadratic combination of (disjoint or overlapping) feature regions. The intent is that if the original feature space has nonlinear boundaries between classes, the new (quadratic) parameter space may have boundaries that are more linear. Given the feature region $\mathbf{x}^{rs} = [x_r, x_{r+1}, \ldots, x_s]$, SFS has three categories of quadratic combinations: using the original feature region, $\mathbf{x}^{rs}$; squaring the feature values for $\mathbf{x}^{rs}$ $[x_r^2, x_{r+1}^2, \ldots, x_s^2]$, or using all pair-wise feature cross products from two regions, $\mathbf{x}^{rs}$ and $\mathbf{x}^{tu} = [x_t, x_{t+1}, \ldots, x_u]$ $(1 \leq t \leq u \leq n)$, producing the result $[x_r x_t, x_r x_{t+1}, \ldots, x_r x_u, \ldots, x_s x_t, x_s x_{t+1}, \ldots, x_s x_u]$. The fitness function (classification performance measure) is P_O. In this study, the only classifier that is used is LDA. When SFS is finished, it returns the best set of classifier results (the cardinality of the set is user-specified) where each result contains (i) the value of P_O, (ii) the indices (to the original features) of the set of feature regions selected, and (iii) the discriminant functions for each class as determined by LDA using the selected feature regions.

2.5. Fuzzy Adaptive Logic Network. Our approach builds upon the fuzzy adaptive logic network (cf. [24] for a thorough description). This approach, which can be used for pattern classification, combines two different subsystems within its general architecture. A neurocomputing subsystem uses a set of perceptrons to construct class boundaries to delineate patterns from different classes. Via a set of respective weights and inputs, a perceptron is defined as $P(\mathbf{x}, \mathbf{w}) = f(\sum_i w_i x_i + w_0)$ where f is a transfer function (any sigmoid function but often the logistic function), which describes an n-dimensional hyperplane. This geometric information is then presented to the logic processing subsystem that comprises a layer of fuzzy conjunctions ("and" elements) and another layer of fuzzy disjunctions ("or" elements). The intent is to use these fuzzy logic connectives to combine the hyperplanes from the neurocomputing subsystem to form convex hull-like topologies. For instance, a convex region delineated by p perceptrons may be represented by the compound logic predicate, $Q = P_1(\mathbf{x}, \mathbf{w}_1), P_2(\mathbf{x}, \mathbf{w}_2), \ldots, P_p(\mathbf{x}, \mathbf{w}_p)$, which produces values close to one (meaning it becomes true) when all contributing predicates are *true* (i.e., the respective perceptrons produce high outputs). To capture the geometric notion of disjoint regions, one may take a union (in the set theoretic sense) of the individual regions described by the Q's: $V = Q_1$ or Q_2 or $\ldots$ or Q_q. To implement these fuzzy predicates, one uses t-norms to model the *and* logic connectives and s-norms to model the *or* logic connectives. A t-norm, $\wedge$, is a function $[0,1]^2 \rightarrow [0,1]$ that is commutative, symmetric, monotonic, and satisfies the boundary conditions $x \,\mathrm{t}\, 0 = 0$ and $x \,\mathrm{t}\, 1 = x$, while the boundary conditions for the s-norm, $\vee$, are $x \,\mathrm{s}\, 0 = x$ and $x \,\mathrm{s}\, 1 = 1$. The fuzzy *or* and *and* connectives may now be defined as

$$\begin{aligned} OR(\mathbf{x};\ \mathbf{w}) &= \wedge_i (w_i \vee x_i), \\ AND(\mathbf{x};\ \mathbf{w}) &= \vee_i (w_i \wedge x_i), \end{aligned} \tag{2}$$

where $\mathbf{x}$ is the input and $\mathbf{w}$ are the corresponding adjustable weights (connections) confined to the unit interval. In the case of $OR(\mathbf{x}; \mathbf{w})$, the greater the weight value the more relevant the respective input (if all weights are 1, it becomes a standard *or* gate). In the case of $AND(\mathbf{x}; \mathbf{w})$, the greater the weight value, the less relevant the respective input (if all weights are 0, it becomes a standard *and* gate). If we restrict ourselves to differentiable t- and s-norms, a gradient descent strategy can be used to train a fuzzy adaptive logic network (cf. [24] for details).

3. Fuzzy Logic Network with Linear Discriminants

Building upon the concepts described in Section 2, we now describe our pattern classification algorithm, FLND (fuzzy logic network with discriminants). There are four major steps to the FLND algorithm: (i) use SFS to find the best κ sets of feature regions using the patterns from the design subset, $\mathbf{X}^D$; (ii) for each set of feature regions, compute the linear discriminant function for each class and then compute the discriminant values for each design pattern; (iii) use a genetic algorithm to determine the optimal weights for the fuzzy logic network given the design pattern discriminant values found in (ii); (iv) use the patterns from the validation subset, $\mathbf{X}^V$, to assess the classification performance, P_O, using the selected feature regions and discriminant function values. Figure 2 illustrates the architecture of the FLND system.

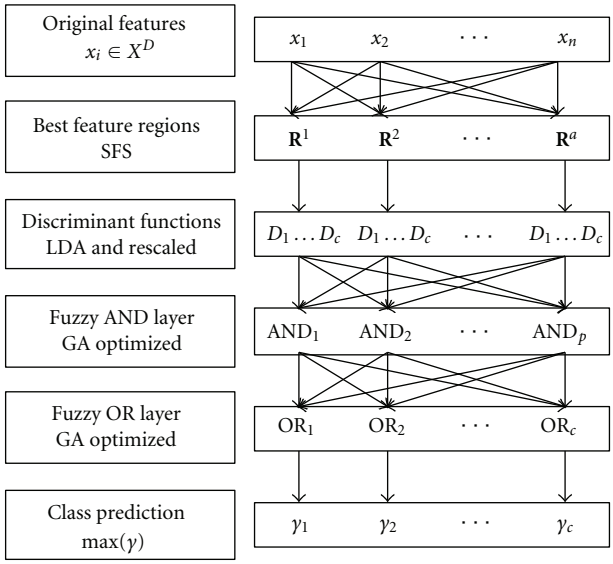

Figure 2: FLND architecture.

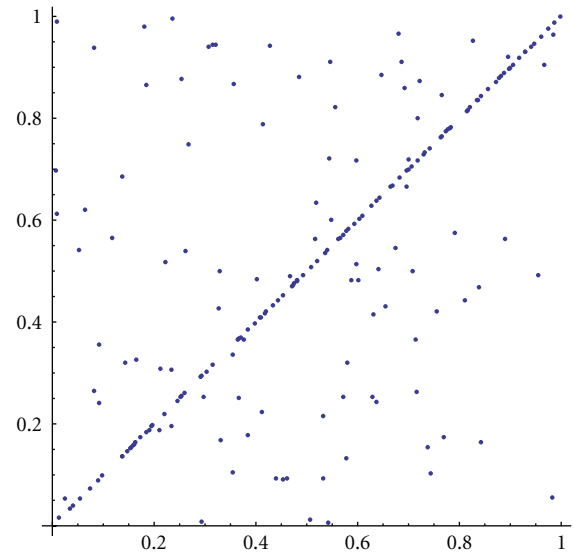

Figure 3: Plot of the first two features of the second synthetic dataset.

Let us now look at each algorithmic step in more detail. In the experiments described in Section 4, SFS uses LDA as the sole classifier and P_O is the performance measure. After a set number of iterations, SFS returns κ sets of feature regions, $\mathbf{R}^\alpha$ ($\alpha = 1, 2, \ldots, \kappa$), and the respective discriminant functions for each class, D_i^α ($i = 1, 2, \ldots, c$), using the feature regions (feature regions are sorted by P_O). The set of feature regions is of the form $\mathbf{R}^\alpha = \{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_{\beta_\alpha}\}$ where β_α is the total number of regions for set α and $\mathbf{r}$ is a single contiguous feature region as described in Section 2.4. The discriminant functions are computed using $\mathbf{R}$ rather than all n features. The input space is now no longer the original n features but rather the respective values of $D(\mathbf{R})$ for each class and each feature region set, which is a significant reduction in the dimensionality of the input space ($c \times \alpha \ll n$).

The fuzzy logic network component of FLND uses the product ($x_1 \times x_2$) and probabilistic sum ($x_1 + x_2 - x_1 \times x_2$) for the t- and s-norms, respectively, with p (user selected) *AND* connectives and *cOR* connectives. There are two deficiencies with this component that does not exist with the fuzzy adaptive logic network described in Section 2.5. First, while perceptron output maps onto the unit interval (due to the sigmoidal nature of its transfer function), which is necessary for input into a fuzzy logic *AND* connective, values from linear discriminant functions map onto $\Re$. This can be easily dealt with by rescaling the linear discriminant values prior to presentation to the fuzzy logic network ($(x - \min) \div (\max - \min)$, where min and max are the respective minimum and maximum for all discriminant function values).

The second, more serious, issue is that a gradient descent strategy cannot be used to minimize the network error (i.e., optimize the weights) since the weight adjustments are now based on discrete sets of discriminant functions rather than differentiable perceptron output. We deal with this issue by using a straightforward implementation of a genetic algorithm (GA) [9, 25, 26] to perform the structural optimization of the network. While much slower than a gradient descent approach, it still provides more than adequate computational performance. We implemented a conventional genetic algorithm as described in [27], but other more sophisticated GA variants could certainly be explored. The crossover rate was set to 0.10, and the mutation rate was set to 0.007 for all experiments listed in Section 4.

Finally, all performance results, using P_O, are based on the class predictions of FLND using the patterns from the validation subset. Further, the results are also benchmarked against conventional applications of LDA and SVM.

4. Experiments and Discussion

4.1. Synthetic Datasets. We begin our experiments with the two-dimensional exclusive-or problem ($n = 2, c = 2, N = 4$). Intuitively, one expects that LDA would perform poorly in this case as no hyperplane can act as a class boundary to perfectly separate the two classes of patterns ($\{\{0, 0\}, \{1, 1\}\}$ versus $\{\{0, 1\}, \{1, 0\}\}$). Using LDA, this is actually the case with $P_O = 0.5$ (one pattern misclassification for each class). As this is a strictly pedagogical experiment, we skip the validation exercise and do not bother with SFS and move directly to the fuzzy logic network. Setting the initial GA population to 200, the number of iterations to 100, and the number of *AND* connectives to 2, we now get perfect accuracy, $P_O = 1.0$. The weights for the two *AND* connectives are $\{0.09, 0.04\}$ and $\{0.0, 1.0\}$. The weights for the subsequent two *OR* connectives are $\{0.37, 0.04\}$ and $\{0.08, 0.25\}$.

This synthetic dataset is a variant of the exclusive-or dataset described above ($n = 10$, $c = 2$, $N = 500$, $N^D = N^V = 250$, $\mathbf{x} \in [0, 1]^n$). A pattern belongs to the first class, if all of its features are identical; otherwise, it belongs to the second class. Figure 3 is a plot of the first two features of this dataset. The initial GA population is 800, the number

Table 1: Confusion matrices for LDA and FLND using $\mathbf{X}^V$.

Desired versus predicted	LDA (P_O=0.5)		FLND (P_O = 0.8)	
	Class 1	Class 2	Class 1	Class 2
Class 1	65	60	88	37
Class 2	63	62	14	111

Table 2: FLND confusion matrices for patterns $\mathbf{X}^D$ (N^D = 80) and $\mathbf{X}^V$ (N^V = 70).

Desired versus predicted	Design patterns (P_O = 0.84)		Validation patterns (P_O = 0.83)	
	Normal	Abnormal	Normal	Abnormal
Normal	33	7	40	9
Abnormal	6	34	3	18

of iterations is 100, and the number of *AND* connectives is 10 (as with the previous experiment, we do not use SFS). In this case, LDA again performed extremely poorly, $P_O = 0.51$, while FLND produced a significantly superior classification accuracy, $P_O = 0.80$. Table 1 lists the confusion matrices for LDA and FLND using this dataset. For completeness, we also list the weights for the *AND* connectives, $\{0.06, 0.34\}$, $\{0.04, 0.15\}$, $\{0.11, 0.07\}$, $\{0.05, 0.08\}$, $\{0.04, 0.08\}$, $\{0.09, 0.09\}$, $\{0.04, 0.08\}$, $\{0.05, 0.20\}$, $\{0.04, 0.05\}$, $\{0.12, 0.20\}$, and the *OR* connectives, $\{0.07, 0.04, 0.08, 0.05, 0.50, 0.04, 0.04, 0.05, 0.03, 0.25\}$, $\{0.04, 0.05, 0.15, 0.20, 0.04, 0.04, 0.05, 0.40, 0.25, 0.04\}$.

4.2. Magnetic Resonance Spectra. Magnetic resonance spectra (patterns) of a biofluid ($n = 4255$) were acquired and used to measure the effectiveness of FLND for the classification of a complex, voluminous, "real world" biomedical dataset. In this case, $N = 150$ with 89 spectra belonging to class 1 ("normal") and 61 spectra belonging to class 2 ("abnormal"). These spectra were randomly allocated to the design subset ($N^D = 80$ with 40 normal spectra and 40 abnormal spectra) or the validation ($N^V = 70$ with the remaining 49 normal spectra and 21 abnormal spectra) subset.

For this dataset, the following SFS parameters were used with FLND: the range for the number of feature regions, 2–5; the range for the number of features within a region, 2–20; $\kappa = 10$; 10^4 iterations. The fuzzy logic network parameters were $p = 7$; crossover rate, 0.10; mutation rate, 0.008; size of GA population, 1200; 50 GA iterations.

Table 2 lists the confusion matrices for FLND with the design patterns and validation patterns. For the design patterns, $P_O = 0.84$, while $P_O = 0.83$ for the validation patterns. Moreover, 82% of the normal (class 1) validation patterns were correctly classified and 86% of the abnormal (class 2) validation patterns were correctly classified. The latter result is especially advantageous as, for many confirmatory biomedical data analysis problems, it is important to have a low false positive rate (i.e., predictions for abnormal conditions should be as accurate as possible).

Table 3 lists the $\kappa = 10$ best sets of discriminatory feature regions, $\mathbf{R}$, found by FLND. For each entry, we list the specific regions selected, how those regions were combined, and the total number of individual features used. Interestingly, over half of the selected discriminatory regions fell in the approximate range 3050–3850, which likely indicates that the biological metabolites represented by this spectral region are particularly germane in distinguishing between normal and abnormal states for the underlying biofluid being investigated. Also important to note is that most of the entries used quadratic combinations of the corresponding feature regions, with the top three results using the pair-wise cross products of the respective regions. Finally, the dimensionality of the feature space is only 4% that of the original space (180 quadratically combined features versus 4255 original spectra features).

4.3. Benchmark Comparisons. We now compare the FLND results from Section 4.2 with two classifier benchmarks, SVM and LDA. First, we use SVM and LDA to construct mappings using all $n = 4255$ features. Subsequently, for each classifier, feature averaging is used as a preprocessing technique, which is a typical strategy for voluminous biomedical spectra, in order to reduce the complexity of the classification problem [28–31]. By reducing the dimensionality of the feature space, we hope to address the curse of dimensionality. Furthermore, averaging has a tendency to attenuate noise signatures. In our specific case, the original features are contiguously averaged using varying window sizes (with no overlap) to produce six sets of averaged features of size 851, 185, 115, 37, 23, and 5, respectively. We use proportional class probabilities for LDA and all SVM kernels listed in Section 2.2. For clarity, in the case of SVM, we report only the best results for each averaged feature set. Table 4 lists the validation subset classification results (confusion matrices and P_O) using the benchmarks with feature averaging. In no case did the benchmarks outperform FLND. Using all of the original features, both benchmarks performed poorly: $P_O = 0.64$ for SVM and $P_O = 0.60$ for LDA. For each benchmark, the best results occurred with 185 averaged features: $P_O = 0.77$ for SVM and $P_O = 0.74$ for LDA. We also note that classification results begin to degrade as the window size increases (i.e., the number of averaged features decreases). This is not uncommon as feature averaging can cause a washing away of information content present in biomedical spectra.

Table 3: Discriminatory feature regions selected by FLND.

κ	Feature regions, $\mathbf{R}^1$–$\mathbf{R}^{10}$	No. of Features	Combination
1	$[x_{1948} \cdots x_{1962}][x_{3642} \cdots x_{3653}]$	180	Cross product
2	$[x_{1207} \cdots x_{1223}][x_{3058} \cdots x_{3073}]$	272	Cross product
3	$[x_{987} \cdots x_{1005}][x_{3544} \cdots x_{3559}]$	304	Cross product
4	$[x_{3817} \cdots x_{3835}]$	19	Square
5	$[x_{3198} \cdots x_{3216}]$	19	Square
6	$[x_{2175} \cdots x_{2193}][x_{3233} \cdots x_{3252}][x_{3849} \cdots x_{3868}]$	59	Original
7	$[x_{3408} \cdots x_{3424}][x_{3441} \cdots x_{3459}]$	323	Cross product
8	$[x_{2349} \cdots x_{2364}][x_{2993} \cdots x_{3004}][x_{3836} \cdots x_{3854}]$	47	Original
9	$[x_{3635} \cdots x_{3649}][x_{3912} \cdots x_{3928}]$	255	Cross product
10	$[x_{3107} \cdots x_{3125}][x_{3782} \cdots x_{3799}][x_{3849} \cdots x_{3868}]$	57	Original

Table 4: Validation patterns ($\mathbf{X}^V$) confusion matrices for benchmark classifiers using averaged features.

No. Features	Desired versus Predicted	SVM			LDA		
		Normal	Abnormal	P_O	Normal	Abnormal	P_O
4255	Normal	33	16	0.64	30	19	0.60
	Abnormal	9	12		9	12	
851	Normal	35	14	0.70	35	14	0.71
	Abnormal	7	14		7	14	
185	Normal	38	11	0.77	37	12	0.74
	Abnormal	5	16		6	15	
115	Normal	37	12	0.76	37	12	0.73
	Abnormal	5	16		7	14	
37	Normal	35	14	0.70	36	13	0.69
	Abnormal	7	14		9	12	
23	Normal	35	14	0.70	35	14	0.71
	Abnormal	7	14		7	14	
5	Normal	30	19	0.60	30	19	0.60
	Abnormal	9	12		9	12	

5. Conclusion

We have empirically demonstrated the effectiveness of a classification technique that uses an adaptive network of fuzzy logic connectives to combine class boundaries generated by sets of discriminant functions based on collections of feature regions possessing high discriminatory power. Using a complex, voluminous "real world" biomedical dataset, FLND outperformed all classifier benchmarks in the classification of patterns from a validation subset. It achieved an 8% improvement in classification accuracy compared against the best benchmark result (0.83 versus 0.77 for SVM using feature averaging with a window size of 23). This increase in classification accuracy is achieved by taking the class boundaries described by the discriminant functions and using layers of fuzzy logic connectives to combine these boundaries into convex, nonlinear boundaries. This new method also significantly reduces the dimensionality of the input space as the original set of spectral features is replaced by a much smaller set of class discriminant values. This is a particularly useful characteristic when dealing with the curse of dimensionality (large feature to sample ratio), which is a prevalent property of many complex biomedical datasets acquired using current spectroscopic modalities.

While this classification technique has demonstrated the utility of merging fuzzy logic connectives with multivariate statistical discrimination, the investigation has also led to the identification of future areas of research to potentially improve its overall effectiveness and computational performance. First, rather than setting the number of fuzzy *and* connectives by the user *a priori*, it would be worthwhile to investigate a cascade approach to determining an optimal number of *and* connections that would be completely data-driven. Second, alternate structural optimizations to the fuzzy logic network need to be examined beginning with more sophisticated evolutionary computational approaches or exploiting recent advances in stochastic optimization techniques. Finally, a more intelligent rescaling strategy for the discriminant function values needs to be investigated. For instance, this may include a fuzzified (weighted) distance measure based on the proximity (belongingness) of a sample to all class boundaries.

Acknowledgments

Conrad Wiebe and Aleksander Demko are gratefully acknowledged for the implementation of the stochastic feature selection algorithm. The aythors also thank the Natural Sciences and Engineering Research Council (NSERC) for its support of this investigation.

References

[1] D. L. Pavia, G. M. Lampman, G. S. Kriz, and J. A. Vyvyan, *Introduction to Spectroscopy*, Harcourt Brace College, Fort Worth, Tex, USA, 2008.

[2] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, Cambridge, UK, 2009.

[3] W. Pedrycz, D. J. Lee, and N. J. Pizzi, "Representation and classification of high-dimensional biomedical spectral data," *Pattern Analysis & Applications*, vol. 13, no. 4, pp. 423–436, 2010.

[4] N. J. Pizzi and W. Pedrycz, "Aggregating multiple classification results using fuzzy integration and stochastic feature selection," *International Journal of Approximate Reasoning*, vol. 51, no. 8, pp. 883–894, 2010.

[5] F. Y. Kuo and I. H. Sloan, "Lifting the curse of dimensionality," *Notices of the American Mathematical Society*, vol. 52, no. 11, pp. 1320–1328, 2005.

[6] I. V. Oseledets and E. E. Tyrtyshnikov, "Breaking the curse of dimensionality, or how to use SVD in many dimensions," *SIAM Journal on Scientific Computing*, vol. 31, no. 5, pp. 3744–3759, 2009.

[7] N. J. Pizzi and W. Pedrycz, "A fuzzy logic network for pattern classification," in *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society*, pp. 53–58, Cincinnati, Ohio, USA, June 2009,.

[8] N. Pizzi, L. P. Choo, J. Mansfield et al., "Neural network classification of infrared spectra of control and Alzheimer's diseased tissue," *Artificial Intelligence in Medicine*, vol. 7, no. 1, pp. 67–79, 1995.

[9] U. M. Braga-Neto and E. R. Dougherty, "Is cross-validation valid for small-sample microarray classification?" *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.

[10] B. Efron and G. Gong, "A leisurely look at the bootstrap, the jackknife, and cross-validation," *The American Scientist*, vol. 37, no. 1, pp. 36–48, 1983.

[11] J. A. Swets, "Measuring the accuracy of diagnostic systems," *Science*, vol. 240, no. 4857, pp. 1285–1293, 1988.

[12] B. S. Everitt, "Moments of the statistics kappa and weighted kappa," *The British Journal of Mathematical and Statistical Psychology*, vol. 21, pp. 97–103, 1968.

[13] G. A. F. Seber, *Multivariate Observations*, John Wiley & Sons, Hoboken, NJ, USA, 2004.

[14] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, Cambridge, UK, 2002.

[15] V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[16] L. Wang, *Support Vector Machines: Theory and Applications*, Springer, Berlin, Germany, 2005.

[17] V. Vapnik and A. Lerner, "Pattern recognition using generalized portrait method," *Automation and Remote Control*, vol. 24, pp. 774–780, 1963.

[18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 3rd edition, 2007.

[19] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1–3, pp. 389–422, 2002.

[20] N. K. Kasabov and Q. Song, "DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 144–154, 2002.

[21] Q. Liu, A. H. Sung, Z. Chen, and J. Xu, "Feature mining and pattern classification for steganalysis of LSB matching steganography in grayscale images," *Pattern Recognition*, vol. 41, no. 1, pp. 56–66, 2008.

[22] E. K. Tang, P. N. Suganthan, and X. Yao, "Gene selection algorithms for microarray data based on least squares support vector machine," *BMC Bioinformatics*, vol. 7, article 95, 2006.

[23] N. J. Pizzi, "Classification of biomedical spectra using stochastic feature selection," *Neural Network World*, vol. 15, no. 3, pp. 257–268, 2005.

[24] W. Pedrycz, A. Breuer, and N. J. Pizzi, "Fuzzy adaptive logic networks as hybrid models of quantitative software engineering," *Intelligent Automation and Soft Computing*, vol. 12, no. 2, pp. 189–209, 2006.

[25] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

[26] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*, John Wiley & Sons, Hoboken, NJ, USA, 2004.

[27] C. Jacob, *Illustrating Evolutionary Computation with Mathematica*, Academic Press, San Diego, Calif, USA, 2001.

[28] R. M. Rangayyan, *Biomedical Signal Analysis: A Case-Study Approach*, Wiley-IEEE Press, New York, NY, USA, 2001.

[29] R. L. Somorjai, M. E. Alexander, R. Baumgartner et al., "A data-driven, flexible machine learning strategy for the classification of biomedical data," in *Artificial Intelligence Methods and Tools for Systems Biology*, W. Dubitzky and F. Azuaje, Eds., pp. 67–85, Springer, Dordrecht, The Netherlands, 2004.

[30] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd, Springer, New York, NY, USA, 2009.

[31] B. C. Wheeler and W. J. Heetderks, "A comparison of techniques for classification of multiple neural signals," *IEEE Transactions on Biomedical Engineering*, vol. 29, no. 12, pp. 752–759, 1982.