Discriminatory Components for Pattern Classification

Nick J. Pizzi[1,2,3] Witold Pedrycz[4]

1 Institute for Bidiagnostics, National Research Council
Winnipeg, Canada
2 Department of Computer Science, University of Manitoba
Winnipeg, Canada
3 Department of Applied Computer Science, University of Winnipeg
Winnipeg, Canada
4 Electrical and Computer Engineering, University of Alberta
Edmonton, Canada
Email: pizzi@cs.umanitoba.ca, pedrycz@ece.ualberta.ca

Abstract—*The pattern recognition literature is replete with the use of principal component analysis in the interpretation and analysis of data. However, in the specific case of classification, especially of biomedical patterns, this pre-processing method, which transforms possibly correlated features into a new set of uncorrelated variables, must be used with caution since a principal component, which may account for significant variance in the data, is not necessarily discriminatory. To compensate for this deficiency, we present a novel classification method using an adaptive network of fuzzy logic connectives to select the most discriminatory principal components. We empirically demonstrate the effectiveness of this method using a benchmark combination of a conventional classifier and principal component analysis.*

Keywords— fuzzy logic network; principal component analysis; biomedical data; pattern classification.

1 Introduction

Today's biomedical instrumentation provides the acquisition of complex data rich in information content; however, its analysis and interpretation is often difficult and requires the latest pattern analysis methodologies [1,2,3]. This is particularly true for the domain of biomedical pattern classification, the prediction by a classifier of the class (for example, normal or abnormal) to which a pattern (for instance, an infrared spectrum of a biofluid) belongs. This prediction is validated against a gold standard, an external reference test such as a pathologist's expert assessment of the same biofluid. Although many classifier methods exist [4,5,6], the most successful approaches combine them with well designed pre-processing techniques, which simplify, in some sense, the feature space prior to presentation to a classifier, and a sound validation protocol to ensure realistic and clinically useful results.

A standard approach to biomedical pattern classification is to use simple linear classifiers coupled with pre-processing transformations that create new features (coordinates, parameters) ordered by the cumulative variance of the original features. A common pairing is linear discriminant analysis coupled with principal component analysis. The ordering allows for the reduction of the feature space by using only the first few components that account for the bulk of the data variance. While this combination is often successful when used in the classification of patterns, its success lies more in the feature reduction aspect rather than the exploitation of the data variance. This is because the feature variance is not concomitant with the discriminatory power of individual features.

We present a new classification method that uses an adaptive network of fuzzy logic connectives that operate on the entire set of principal components. This method exploits the discriminatory power of any principal component regardless of the amount of variance for which it accounts. We empirically demonstrate that this method produces superior classification performance compared to a benchmark using the conventional approach of linear discriminant analysis operating on the first few principal components of the original feature space. Section 2 presents a general discussion on classification including classifier validation, principal component analysis, linear discriminant analysis, and a recent fuzzy adaptive logic network on which the current method is based. Details of our novel approach are presented in Section 3. The synthetic and biomedical datasets, experiment design, and results are discussed in Section 4 followed by some concluding remarks and areas of future investigations.

2 Pattern classification

2.1 Classifier Validation

We begin by defining formal pattern classification notation: N is the number of patterns (samples, vectors, individuals, or cases); n is the number of features (dimensions, attributes, coordinates, or measurements); c is the number of classes (groups); and $\boldsymbol{X}=\{(\boldsymbol{x}_k,\omega_k),\ k=1,2,\ldots,N\}$ is a set of N labeled patterns where $\boldsymbol{x}_k\in\Re^n$ and $\omega_k\in\Omega=\{1,2,\ldots,c\}$. A classifier is a system that determines a mapping $f:\boldsymbol{X}\rightarrow\Omega$. If a classifier predicts that the class label for pattern $\boldsymbol{x}_i$ is ω_p then: a correct classification occurs if $\omega_p=\omega_i$; otherwise it is considered a misclassification.

Many investigations involving data classification are biased as they use the entire dataset to determine the mapping. This approach leads to unrealistic classification results that do not take into account the possibility of overfitting, wherein the

ISBN: 978-989-95079-6-8

mapping becomes a simple table lookup, between the patterns and class labels (that is, it possesses no generalized predictive power for new patterns). To compensate for this bias, it is essential to perform some type of validation. For instance, patterns in $\boldsymbol{X}$ may be randomly allocated to a design subset, $\boldsymbol{X}^D$ containing N^D patterns, or a validation subset, $\boldsymbol{X}^V$ containing N^V patterns ($N^D+N^V=N$). Now, a mapping is determined using only design patterns, $f':\boldsymbol{X}^D\rightarrow\Omega$, but the classification performance is measured using f' with the validation patterns.

Performance of a classification system is measured using the $c\times c$ confusion matrix, $\boldsymbol{C}$, of the desired class labels (as determined by an external reference test or "gold standard") versus the predicted class labels (as generated by the classifier). If the class prediction for $\boldsymbol{x}_i$ is ω_p then element $[\omega_p,\omega_i]$ of $\boldsymbol{C}$ is incremented by one. The conventional performance measure is the ratio of correctly classified patterns to the total number of patterns, $P_O=(\sum_i n_{ii})/N^V$ (i=1, 2,…, c), where n_{ij} is the number of class i validation patterns that are predicted to belong to class j.

A significant problem with P_O occurs when there is a large disparity between class sizes. For instance, if a validation set has many more patterns in one class than another, the classification accuracy relating to the former will outweigh any effects from the latter. For instance, a high value for P_O may occur due to few misclassifications with the larger class even though many misclassifications occur with the smaller class. This is especially problematic for biomedical patterns where it is relatively easy to acquire patterns from "normal" samples (tissues, biofluids, and so on) but clinically difficult to obtain "abnormal" samples. As a result, it is often better to forgo the use of P_O for these types of datasets and use the class-wise average accuracy, P_A

$$P_A = c^{-1}\sum_i^N \frac{n_{ii}}{\sum_j^N n_{ij}} (i,j=1,\ldots,c) \qquad (1)$$

2.2 Principal Component Analysis

The motivation behind principal component analysis, first described by Pearson [7] (with a practical computing method described by Hotelling [8]), is to find a set of directions (coordinates) that explain as much of the variability of the original data as possible. In other words, the principal components are a new set of orthogonal linear coordinates such that the variances of the original features with respect to these derived coordinates are in decreasing orders of magnitude [9]. As a result, each principal component is uncorrelated with all other principal components (in a normal distribution, they are statistically independent). Moreover, it can be shown [10] that no other set of m coordinates can account for more of the variability in the original data than the first m principal components.

The first principal component, $\boldsymbol{Y}_1$, of the original features x_1, x_2, …, x_n, is the linear combination

$$\boldsymbol{Y}_1 = \sum_i^n a_{1i}x_i, \sum_i^n a_{1i}^2 = 1 \qquad (2)$$

The constraint on the coefficients is necessary; otherwise the variance of $\boldsymbol{Y}_1$ can be increased simply by increasing the value of any coefficient. The second principal component, $\boldsymbol{Y}_2$, would be computed in a similar fashion to (1). Fig. 1 is a plot of some bivariate data and their two principal components. It is clear from this figure that an additional constraint, orthogonality to the first principal component, is required to compute the second principal component, otherwise it would simply be driven to the first principal component. Orthogonality is ensured by restricting the variables of the second principal component to those that are uncorrelated with the first principal component. As a result of this orthogonality constraint, if there are n features then there can be up to n principal components [11]. In fact, if the original features are completely uncorrelated, then all n principal components must be used to take into account the variance in the original features. In this case, principal component analysis serves little purpose, with respect to pattern classification, since the motivation behind the technique is to reduce the dimensionality of the original input (feature) space. However, in "real-world" high-dimensional biomedical patterns, the converse is usually true; features are highly correlated and hence only $1\leq m<<n$ principal components are required to account for all (or nearly all) of the variation.

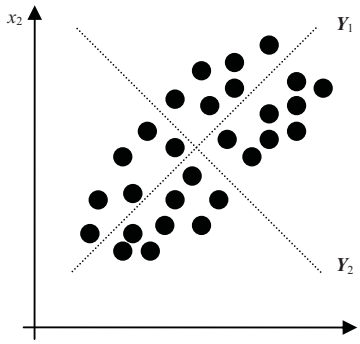

Figure 1: Principal components, $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$, for some bivariate data.

Determining the principal components is a straightforward process involving the computation of the eigensystem of the original data's covariance matrix, $\boldsymbol{V}$, whose element v_{lm} is the sample covariance between features x_l and x_m

$$v_{lm} = \frac{1}{N-1}\sum\nolimits_N \left(x_{il}-\mu_l\right)\left(x_{im}-\mu_m\right) \qquad (3)$$

where μ_j is the mean for feature x_j (cf. [12] for a derivation of the proof). The variances of the principal components are the eigenvalues of $\boldsymbol{V}$, $\lambda_1\geq\lambda_2\geq\ldots\geq\lambda_n\geq 0$ (the covariance matrix is quadratic and hence admits no negative eigenvalues). The variance of a principal component, $\boldsymbol{Y}_i$, is λ_i and its constants a_{i1}, a_{i2}, …, a_{in} are the elements of the corresponding eigenvector.

A standard strategy employed in biomedical pattern classification is to take the first m principal components whose cumulative variance exceeds some pre-defined threshold. This reduction is often significant ($m<<n$): for instance in some high-dimensional infrared spectra more than 80% of the cumulative variance may be accounted for by only the first one or two principal components [13].

ISBN: 978-989-95079-6-8

While this pre-processing strategy may be effective, it must be noted that the components are ordered by maximal variance. Unfortunately, this does not necessarily translate into maximal discriminatory power [14]. For instance, if the method used to acquire values for a particular feature is extremely prone to measurement error, then this feature will have a high variance. Now, assuming this variance is greater than other features, the first principal component will be approximately equal to this suspect feature, and hence, this principal component will be useless in discriminating between classes. Conversely, a highly discriminatory feature may have an extremely small variance and hence will not contribute to the first few principal components. In summary, maximal discriminatory power is not equivalent to maximal variance.

2.3 Linear Discriminant Analysis

Linear discriminant analysis (LDA) [15] is a standard pattern classification strategy that determines linear boundaries between c classes while taking into account between-class and within-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 assigns a pattern, $\boldsymbol{x}$, to class i for which the probability distribution, $p_i(\boldsymbol{x})$, is greatest. That is, $\boldsymbol{x}$ is allocated to class i, if $q_i p_i(\boldsymbol{x}) \geq q_j p_j(\boldsymbol{x})$ $(\forall j=1,2,\ldots,c\ [j\neq i])$, where q_i is the class' prior (or proportional) probability. The discriminant function for class i is

$$D_i(\boldsymbol{x}) = \log q_i + \boldsymbol{\mu}_i^{\mathrm{T}} \boldsymbol{W}^{-1} \left(\boldsymbol{x} - \tfrac{1}{2} \boldsymbol{\mu}_i \right) \qquad (4)$$

where $\boldsymbol{\mu}_i$ is the mean for class i and $\boldsymbol{W}$ is the covariance matrix of the patterns in $\boldsymbol{X}$. The feature space hyperplane separating class i from j is defined by $F_{ij}(\boldsymbol{x})=D_i(\boldsymbol{x})-D_j(\boldsymbol{x})=0$. As a classification system, LDA is often used with the first m principal components of a dataset rather than the original features. This combination will serve as the classification benchmark against which our novel pattern classification method will be evaluated.

2.4 Fuzzy Adaptive Logic Network

Our novel approach is partly based upon a modification to the fuzzy adaptive logic network (cf. [16] for a thorough description). This network, which can be used for pattern classification, combines two different subsystems within a general architecture. A neurocomputing subsystem uses a set of perceptrons to construct class boundaries. Through a set of weights and respective inputs, a perceptron is defined as $P(\boldsymbol{x},\boldsymbol{w})=f(\sum_i w_i x_i + w_0)$ (f is a sigmoidal transfer function) describing an n-dimensional hyperplane. This geometric information is presented to the logic processing subsystem composed of a layer of fuzzy conjunctions ("and" elements) and 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, $Q = P_1(\boldsymbol{x},\boldsymbol{w}_1)$ *and* $P_2(\boldsymbol{x},\boldsymbol{w}_2)$ *and* … *and* $P_p(\boldsymbol{x},\boldsymbol{w}_p)$, which produces values close to one (in other words, approaching the crisp notion of *true*) when all contributing predicates are *true* (that is, the respective perceptrons produce high outputs).

To capture the geometric notion of disjoint regions one may take a union (in the fuzzy set theoretic sense) of the individual regions described by the Q's, $R = Q_1$ *or* Q_2 *or* … *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

$$\begin{aligned} OR(\boldsymbol{x};\boldsymbol{w}) &= \wedge_i \left(w_i \vee x_i \right) \\ AND(\boldsymbol{x};\boldsymbol{w}) &= \vee_i \left(w_i \wedge x_i \right) \end{aligned} \qquad (5)$$

where $\boldsymbol{x}$ are the inputs and $\boldsymbol{w}$ are the corresponding adjustable weights (connections) confined to the unit interval. In the case of $OR(\boldsymbol{x};\boldsymbol{w})$, 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 $AND(\boldsymbol{x};\boldsymbol{w})$, 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. [16] for details).

3 Fuzzy logic classification network using principal components

Building upon the concepts described in Section 2, we now describe a novel pattern classification algorithm, COAP (Classification using a fuzzy Or/And network with Principal components), a component of which extends the fuzzy logic network architecture developed by the authors [17]. There are three major steps to the COAP algorithm: (i) apply principal component analysis on the original features to find all n principal components; (ii) Use a genetic algorithm to determine the optimal weights for the fuzzy logic network given the principal component values found in (i); (iii) use the patterns from the validation subset to assess the classification performance using the selected feature regions and principal component values. Fig. 2 illustrates the architecture of the COAP system.

Let us now look at each algorithmic step in more detail. After applying principal component analysis to the feature space, we replace pattern, $\boldsymbol{x}_i=[x_1,x_2,\ldots,x_n]$, with its respective principal component values, $\boldsymbol{y}_i=[\boldsymbol{Y}_1(\boldsymbol{x}_i),\boldsymbol{Y}_2(\boldsymbol{x}_i),\ldots,\boldsymbol{Y}_n(\boldsymbol{x}_i)]$. These principal component values for all design set patterns are subsequently presented to the fuzzy logic network component.

COAP's fuzzy logic network 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 c *OR* connectives. There are two issues with this network that

ISBN: 978-989-95079-6-8

do not exist with the fuzzy adaptive logic network described in Section 2.4. First, while output from a perceptron 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, principal component values map onto $\Re$. This can be dealt with by rescaling the principal component values prior to presentation to the fuzzy logic network ((*x*–*min*)÷(*max*–*min*), where *min* and *max* are the respective minimum and maximum for all principal component values).

The second, more serious, issue is that a gradient descent strategy cannot be used to minimize the fuzzy logic network error (optimize the weights) since the weight adjustments are now based on sets of principal components rather than differentiable perceptron output. We deal with this issue by using a straightforward implementation of a genetic algorithm [18,19,20] to perform the structural optimization of the network. While much slower than gradient descent, this solution still provides adequate computational performance. In this study, we implemented a conventional genetic algorithm as described in [21], but other more sophisticated genetic algorithm variants could certainly be explored.

As mentioned in Section 2.1, $\Omega=\{1,2,\ldots,c\}$; however, it is often beneficial [13] to use 1-of-*c* encoding for the class labels for iterative classifiers such as artificial neural networks or fuzzy logic networks, namely, $\Omega=\{\gamma_1,\gamma_2,\ldots,\gamma_c\}$ where, for $\boldsymbol{x}_i$, $\gamma_{\omega_i}=1$ and $\gamma_{\omega_j}=0$ $(\forall\omega_i\neq\omega_j)$. Instead of using one output element to represent all *c* classes, we use *c* output elements.

Finally, all performance results using P_A are based on the class predictions of COAP using the biomedical patterns from the validation subset. That is, the principal component values are obtained for the validation patterns and presented to the fuzzy logic network component. Subsequently, the predictions are compared against the desired class labels (if the output element with the maximum value corresponds to the desired label's non-zero component then we have a successful classification, otherwise it is a misclassification).

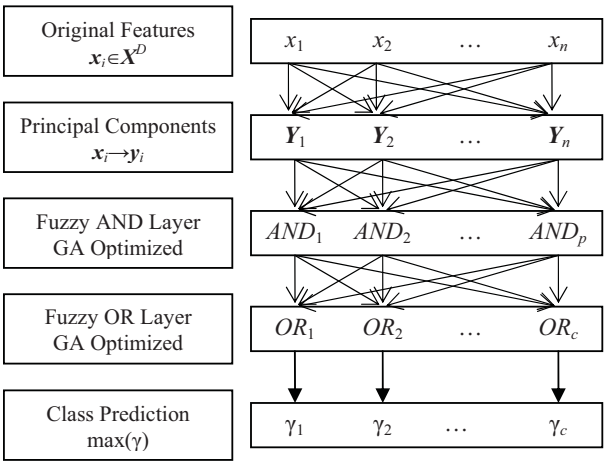

Figure 2: General architecture of COAP with the GA (genetic algorithm) optimized fuzzy logic network.

4 Experiments and discussion

4.1 Synthetic Datasets

We begin our experiments with the "exclusive or" dataset (*n*=2, *c*=2, *N*=4): patterns {{0,0},{1,1}} and {{0,1},{1,0}}; and respective principal component values of {{0,0},{1,–1}} and {{1,0},{0,–1}}. Intuitively, one expects that LDA would perform poorly in this case as no hyperplane can act as a class boundary to perfectly separate the patterns. Using LDA with the principal components, this is actually the case with P_A=0.5 (one misclassification for each class). [As this is a strictly pedagogical experiment, we skip validation.] Setting the initial genetic algorithm population to 50, the number of iterations to 5, and the number of *AND* connectives to 2, we now get perfect accuracy, P_A=1.0. The weights for the two *AND* connectives are {0.95,0.05} and {0.0,1.0}. The weights for the two *OR* connectives are {0.53,0.05} and {0.07,0.35}.

This next dataset is a variant of the exclusive or dataset described above (*n*=10, *c*=2, *N*=400, $\boldsymbol{x}\in[0,1]^n$). A pattern belongs to the first class, if all of its features are identical; otherwise it belongs to the second class. Fig. 3 is a plot of the first two features of this dataset. The initial genetic algorithm population is 100, the number of iterations is 5, and the number of *AND* connectives is 10. In this case, LDA with principal components also performed poorly, P_A=0.5, while COAP produced a higher classification accuracy, P_A=0.81.

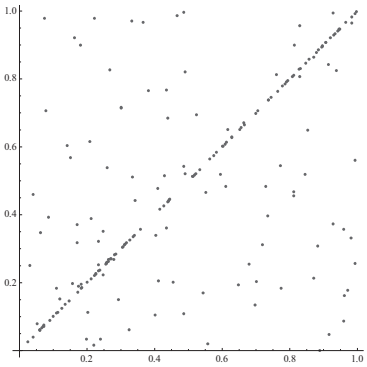

Figure 3: Plot of, x_1 and x_2, for the second synthetic dataset.

4.2 Biomedical Dataset

We used a biomedical dataset from the Machine Learning Repository at the University of California, Irvine [22]. This archive is used to evaluate machine learning algorithms. This dataset [23,24] comprises voice measurements acquired from two classes of subjects: those with Parkinson's disease (abnormal); and those without (normal). There are *N*=195 patterns from one of two classes: N_a=48 abnormal patterns; N_n=147 normal patterns. The *n*=22 features include: the maximum, minimum, and average fundamental vocal frequency (3 features); several measures of variation in fundamental frequency (5 features); several measures of variation in amplitude (6 features); measures of the ratio of noise to tonal components in the voice (2 features); nonlinear dynamical complexity measures (2 features); a

ISBN: 978-989-95079-6-8

signal fractal scaling exponent (1 feature); and nonlinear measures of fundamental frequency variation (3 features).

For this dataset, the following COAP parameters were used: p=7 (the number of *AND* connectives); crossover rate, 0.07; mutation rate, 0.005; size of genetic algorithm population, 200; and 20 iterations of the genetic algorithm. The patterns were randomly assigned to either a design subset (N^D=64 with 32 normal patterns and 32 abnormal patterns) or a validation subset (N^V=131 with 115 normal patterns and 16 abnormal patterns). Finally, given the class size disparity between the abnormal and normal patterns within the validation set, P_A was used to assess classification accuracy.

Table 1 lists the confusion matrices for COAP for the design patterns and validation patterns. For the design patterns, P_A=0.91, while P_A=0.85 for the validation patterns. While only 70% of the normal validation patterns were correctly classified, all abnormal validation patterns were correctly classified. Table 2 lists the confusion matrices for LDA using all n principal components. In the case of this benchmark, P_A=0.94 for the design patterns and P_A=0.70 for the validation patterns. The disparity between the design and validation results is a classic sign of potential overfitting. While 75% of the abnormal validation patterns were correctly classified, only 65% of the normal patterns were correctly classified.

Table 1: COAP confusion matrices for $\boldsymbol{X}^D$ and $\boldsymbol{X}^V$.

Desired vs	Design Set (P_A=0.91)		Validation Set (P_A=0.85)	
Predicted	Abnormal	Normal	Abnormal	Normal
Abnormal	29	3	16	0
Normal	3	29	35	80

Table 2: Benchmark confusion matrices for $\boldsymbol{X}^D$ and $\boldsymbol{X}^V$.

Desired vs	Design Set (P_A=0.94)		Validation Set (P_A=0.70)	
Predicted	Abnormal	Normal	Abnormal	Normal
Abnormal	31	1	12	4
Normal	3	29	40	75

Table 3 lists the confusion matrices (desired versus predicted class labels) using successive combinations of principal component in turn (ordered by variance) for both the design and validation sets. Note that: P_A is listed for the design set followed by the validation set; $\sum^{\alpha}\lambda_i$ refers to the cumulative variance of the first α principal components used to produce the corresponding results; and the last entry is the same as that described in Table 2. The best result, P_A=0.781, occurred when using the first 13 principal components. It is clear with this dataset that feature variance is not well correlated with discriminatory features (for example, the first three principal components, which account for 0.998 of the cumulative variance, produce a low accuracy score, P_A=0.60).

Fig. 3 summarizes these results by plotting the COAP validation result against the validation results using all successive combinations of principal components. It should also be noted that the last few entries in Table 3 clearly demonstrate a problem with overfitting with a disparity of approximately 0.24 between the design and validation results.

Table 3: Results using successive principal components.

	D vs P	$\boldsymbol{X}^D$		$\boldsymbol{X}^V$	
		A	N	A	N
$\sum^{1}\lambda_i$=0.729	A	24	8	8	8
P_A=0.77/0.63	N	7	25	27	88
$\sum^{2}\lambda_i$=0.947	A	22	10	7	9
P_A=0.69/0.61	N	10	22	25	90
$\sum^{3}\lambda_i$=0.998	A	23	9	7	9
P_A=0.70/0.60	N	10	22	27	88
$\sum^{4}\lambda_i$=1.000	A	22	10	9	7
P_A=0.73/0.62	N	7	25	36	79
$\sum^{5}\lambda_i$=1.000	A	28	4	10	6
P_A=0.83/0.69	N	7	25	29	86
$\sum^{6}\lambda_i$=1.000	A	28	4	10	6
P_A=0.83/0.69	N	7	25	29	86
$\sum^{7}\lambda_i$=1.000	A	28	4	10	6
P_A=0.84/0.69	N	7	25	29	86
$\sum^{8}\lambda_i$=1.000	A	28	4	11	5
P_A=0.83/0.72	N	7	25	32	83
$\sum^{9}\lambda_i$=1.000	A	27	5	11	5
P_A=0.81/0.70	N	7	25	32	83
$\sum^{10}\lambda_i$=1.000	A	27	5	12	4
P_A=0.81/0.74	N	7	25	32	83
$\sum^{11}\lambda_i$=1.000	A	29	3	13	3
P_A=0.92/0.74	N	2	30	38	77
$\sum^{12}\lambda_i$=1.000	A	29	3	13	3
P_A=0.92/0.75	N	2	30	37	78
$\sum^{13}\lambda$=1.000	A	30	2	14	2
P_A=0.92/0.78	N	3	29	36	79
$\sum^{14}\lambda$=1.000	A	30	2	14	2
P_A=0.92/0.78	N	3	29	37	78
$\sum^{15}\lambda_i$=1.000	A	30	2	14	2
P_A=0.92/0.78	N	3	29	37	78
$\sum^{16}\lambda_i$=1.000	A	32	0	12	4
P_A=0.94/0.71	N	4	28	39	76
$\sum^{17}\lambda_i$=1.000	A	32	0	14	2
P_A=0.95/0.76	N	3	29	41	74
$\sum^{18}\lambda_i$=1.000	A	31	1	12	4
P_A=0.94/0.70	N	3	29	40	75
$\sum^{19}\lambda_i$=1.000	A	31	1	12	4
P_A=0.94/0.70	N	3	29	40	75
$\sum^{20}\lambda_i$=1.000	A	31	1	12	4
P_A=0.94/0.70	N	3	29	40	75
$\sum^{21}\lambda_i$=1.000	A	31	1	12	4
P_A=0.94/0.70	N	3	29	40	75
$\sum^{22}\lambda_i$=1.000	A	31	1	12	4
P_A=0.94/0.70	N	3	29	40	75

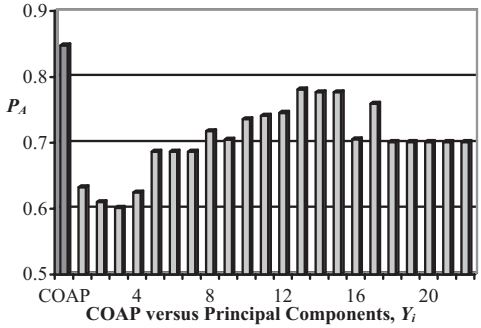

Figure 3: Summary of validation set accuracies.

ISBN: 978-989-95079-6-8

Finally, after examination of the weights of the fuzzy logic network, six principal components tended to significantly contribute to the COAP training, $\boldsymbol{Y}_1$, $\boldsymbol{Y}_4$, $\boldsymbol{Y}_5$, $\boldsymbol{Y}_{10}$, $\boldsymbol{Y}_{12}$, and $\boldsymbol{Y}_{16}$. This is further evidence that care must be taken when using principal component analysis as one may overlook discriminatory components simply because they account for little feature variance (for example, the variance for $\boldsymbol{Y}_{10}$, $\boldsymbol{Y}_{12}$, and $\boldsymbol{Y}_{16}$ is zero).

5 Conclusions

We have empirically demonstrated the effectiveness of a novel classification method that uses an adaptive network of fuzzy logic connectives to combine new features generated using principal component analysis. Using a "real world" biomedical dataset, COAP correctly classified significantly more patterns from a validation set compared to the benchmark.

While this novel classification method 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, alternative 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. A final area of investigation is using more sophisticated component analysis methods for feature pre-processing (non-linear and local principal component methods, other kernel based approaches, and fuzzy set based feature selection/extraction approaches).

Acknowledgment

We are grateful to Max Little of the University of Oxford who, in collaboration with the National Centre for Voice and Speech, Denver, Colorado, recorded the speech signals for the contribution of the biomedical dataset within the publicly available machine learning repository. We thank Drs. Asuncion and Newman of the University of California, Irvine who maintain the highly useful and publicly available data repository.

Conrad Wiebe and Aleksander Demko are gratefully acknowledged for the implementation and testing of some of the software framework components used in the implementation of this pattern classification method and validation protocol.

This investigation was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

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

[2] H. Friebolin. *Basic One- and Two-Dimensional NMR Spectroscopy*. New York: Wiley and Sons, 1998.

[3] M. Akay. *Nonlinear Biomedical Signal Processing, Volume I: Fuzzy Logic, Neural Networks, and New Algorithms*. New York: IEEE Press, 2000.

[4] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. New York: Wiley Interscience, 2000.

[5] B. Bouchon-Meunier, G. Coletti, R.R. Yager. *Modern Information Processing: From Theory to Applications*. Amsterdam: Elsevier, 2006.

[6] I.H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques (2nd edition)*. San Francisco: Morgan Kaufmann Publishers, 2005.

[7] K. Pearson. On lines and planes of closest fit to a system of points in space. *Philosophical Magazine*, 2:557–572, 1901.

[8] H. Hotelling. Analysis of a complex of statistical variables into principal components. *J Educational Psychology*, 24:417–441, 1933.

[9] I.H. Bernstein, C.P. Garbin, and G.K. Teng. *Applied Multivariate Analysis*. New York: Springer-Verlag, 1988.

[10] W. Krzanowski. *Principles of Multivariate Analysis*. New York: Oxford University Press, 1988.

[11] B.F.J. Manly. *Multivariate Statistical Methods: A Primer*. New York: Chapman & Hall, 1986.

[12] R. Gnanadesikan. *Methods for Statistical Data Analysis of Multivariate Observations*. New York: John Wiley, 1977.

[13] N.J. Pizzi, L.-P. Choo, J. Mansfield, M. Jackson, W.C. Halliday, H.H. Mantsch, R.L. Somorjai. Neural network classification of infrared spectra of control and Alzheimer's diseased tissue. *Artificial Intelligence in Medicine*, 7:67–79 (1995).

[14] J.D. Jobson. *Applied Multivariate Data Analysis: Categorical and Multivariate Methods*. New York: Springer–Verlag, 1992.

[15] G.A.F. Seber. *Multivariate Observations*. Hoboken: Wiley and Sons, 1984.

[16] W. Pedrycz, A. Breuer, and N.J. Pizzi. Fuzzy adaptive logic networks as hybrid models of quantitative software engineering. *Intelligent Automation and Soft Computing*, 12:189–209, 2005.

[17] N.J. Pizzi and W. Pedrycz. A fuzzy logic network for pattern classification. *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society*, June 14–17, Cincinnati, USA, 2009 (submitted for review).

[18] K.A. De Jong. *Evolutionary Computation: A Unified Approach*. Cambridge: MIT Press, 2006.

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

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

[21] C. Jacob. *Illustrating Evolutionary Computation with Mathematica*. San Diego: Academic Press, 2001.

[22] A. Asuncion and D.J. Newman. *UCI Machine Learning Repository* [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science, 2007.

[23] M.A. Little, P.E. McSharry, E.J. Hunter, and L.O. Ramig. Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. *IEEE Transactions on Biomedical Engineering*, 2008 (to appear).

[24] M.A. Little, P.E. McSharry, S.J. Roberts, D.A.E. Costello, and I.M. Moroz. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *Biomedical Engineering Online*, 6:23, 2007.

ISBN: 978-989-95079-6-8