Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Data Mining (CIDM 2007)

Classification of Biomedical Spectra Using Fuzzy Interquartile Encoding and Stochastic Feature Selection

Nick J. Pizzi, *Senior Member*, IEEE, Mark D. Alexiuk, and Witold Pedrycz, *Fellow*, IEEE

***Abstract*—Accurate classification of biomedical spectra is often difficult due to the large number of features, which tends to have a confounding effect. We present a strategy where the original spectral feature space is transformed using a fuzzy set theoretic method, which analyzes the features' interquartile ranges, coupled with a stochastic feature selection mechanism, which identifies highly discriminatory feature subsets. We demonstrate the effectiveness of this strategy using biofluid data acquired from a magnetic resonance spectrometer.**

I. Introduction

The selection of a classifier is only one aspect of the problem of data classification. Equally important (if not, more so) is the preprocessing strategy to be employed. Many classification problems, particularly those involving the analysis of voluminous biomedical spectra, involve features whose specificity demand some form of feature transformation to construct effective decision boundaries between different classes. We present a strategy that uses fuzzy interquartile encoding to transform the original feature space using membership functions constructed using the interquartile ranges of the respective features. This transformation has a normalizing effect on the feature space and is more robust to feature outliers. We couple this preprocessing strategy with stochastic feature selection, which assesses the discriminatory performance of feature subsets that are sampled using an ad hoc cumulative distribution function of a frequency histogram of features that contributed to prior "successful" classifications. The motivation behind this selection process is that often only a subset of features possesses discriminatory power while the remainder has a tendency to confound the effectiveness of the underlying classifier.

Fig. 1 illustrates the strategy that is used. The original feature space is transformed using fuzzy interquartile encoding. Subsets of features are selected from the transformed space using the stochastic selection mechanism. The subsets, which may undergo a quadratic transformation, are presented to a set of corresponding classifiers. The design (training) phase is substantiated using n-fold validation.

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

N. J. Pizzi is a senior research officer with the Institute for Biodiagnostics, National Research Council, Winnipeg, MB, R3B 1Y6, Canada. He is also an adjunct professor with the Department of Computer Science, University of Manitoba, Winnipeg, MB, R3T 2N2, Canada (e-mail: pizzi@nrc-cnrc.gc.ca; phone: +1 204 983 8842; fax: +1 204 984 5472).

M. D. Alexiuk is a PhD student with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, R3T 2N2, Canada.

W. Pedrycz is a professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, T6G 2N4, Canada (e-mail: pedrycz@ece.ualberta.ca). He is also the Canada Research Chair in Computational Intelligence.

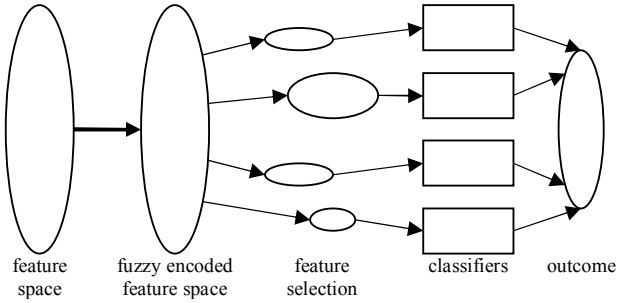

Fig. 1. Schema of fuzzy encoded feature selection strategy.

To empirically validate this strategy, we study a classification problem coming from the area of biomedical data analysis, specifically the assignment of magnetic resonance spectra of a biofluid into either a normal or abnormal class. We demonstrate that the classification accuracy using the fuzzy interquartile encoding strategy is superior to using the original spectral features.

II. Classification

Formally, consider a c-class classification problem in which $\boldsymbol{X}=\{(\boldsymbol{x}_k,\omega_k),\ k=1,2,\ldots,N\}$ is a set of class labeled patterns. Here, $\boldsymbol{x}_k\in\Re^n$ and $\omega_k\in\Omega$, where $\Omega=\{1,2,\ldots,c\}$. In essence, a classifier is a mapping $g:\mathbf{X}\rightarrow\Omega$. Let ω_{ip} be the class label predicted by classifier p for pattern, x_i. If $\omega_{ip}=\omega_i$ we say that p generated a correct classification result for $\boldsymbol{x}_i$, otherwise it is a misclassification. Many classifier architectures exist (supervised artificial neural networks, multivariate statistical methods, evolutionary computation approaches, hybrid strategies, and so on). The classifier used in this investigation is the radial basis function neural network.

Data preprocessing is of paramount importance for problems of classification. Advanced technologies contribute ever more sophisticated models upon which to build ever more sophisticated classifiers. Herein lies a major problem: if these models are highly non-linear, they may be unstable, if they are iterative, they may not converge, if they are probabilistic, they may be based on underlying statistical assumptions that are often not true in real-world scenarios. Preprocessing may address these concerns: data may be transformed such that a non-linear model may be replaced by a linear one, the dimensionality of the data may be reduced so that an iterative method may converge or may be substituted for an analytic one, or the data may be "normalized", in some sense, such that the underlying statistical assumptions of a probabilistic model are realized. The preprocessing strategy used in this study is a

combination of fuzzy interquartile encoding ("normalization") and stochastic feature selection (dimensionality reduction).

Finally, successful classification demands a reliable and conservative measure of performance. To satisfy this requirement, we employ the following approach. The dataset is divided into a design set and a test set. Interquartile parameters for the encoding method are computed using only the design set and subsequently applied to the test set. We use n-fold validation; data resubstitution is performed n times and average (test set) classification accuracies with standard deviations are accumulated.

Classification accuracy is measured using the $C \times C$ confusion matrix of desired versus predicted class labels. Rather than using the standard performance (fitness) measure, P_o (the ratio of correctly classified test patterns to the total number of test patterns)

$$P_o = N^{-1} \sum_i C_{ii} \, (i = 1, \ldots, c) \tag{1}$$

we use the κ-score [1], a chance-corrected measure of agreement [2],

$$\kappa = (P_o - P_L) / (1 - P_L) \tag{2}$$

where P_L is the agreement due to chance

$$P_c = N^{-2} \sum_i \left(\sum_j C_{ij} \sum_j C_{ji} \right) (i, j = 1, \ldots, c) \tag{3}$$

We also use the corresponding κ measures for sensitivity, κ(1,0), and specificity, κ(0,0) [3][4][5]

$$\begin{aligned} \kappa(0,0) &= (C_{11}C_{22} - C_{12}C_{21}) / ((C_{11} + C_{21})(C_{21} + C_{22})) \\ \kappa(1,0) &= (C_{11}C_{22} - C_{12}C_{21}) / ((C_{11} + C_{12})(C_{12} + C_{22})) \end{aligned} \tag{4}$$

A. Radial Basis Function Neural Network

A radial basis function neural network (RBF) [6][7][8] has an internal representation of hidden processing elements (PEs) with a topology as shown in Fig. 2. A PE output possesses radial symmetry

$$f(\boldsymbol{x}) = \varphi(\|\boldsymbol{x} - \boldsymbol{\mu}\|) \tag{5}$$

where: $\boldsymbol{\mu}$ is its center represented by a vector in the input space (and stored in its input layer weights); $\|\cdot\|$ is a distance metric (usually Euclidean) that determines how far an input pattern is from $\boldsymbol{\mu}$; and, the transfer function, φ (usually a Gaussian), must output high values when the distance from a pattern to $\boldsymbol{\mu}$ is small, and low values otherwise. RBFs are a class of universal function approximators [9] that are often used as classifiers. That is, given an RBF with enough hidden layer PEs, it can approximate any continuous function with arbitrary accuracy [10]. RBFs typically train more quickly than conventional multilayer perceptrons [11]. Also, the PEs represent density functions for the input space and may be used as a probability measure for new patterns. However, there are two problems with RBFs: since the receptive fields are localized they do not perform well if discriminatory features are globally distributed throughout the input space; and, selecting the number of receptive fields is strictly ad hoc.

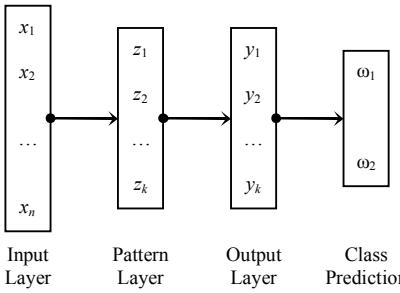

Fig. 2. The topology of a radial basis function neural network.

Fig. 3 illustrates the structure of a pattern layer PE. If $\boldsymbol{\mu}_i$ is a column vector representing the center of pattern layer PE, i, and σ_i is the diameter of its receptive region, then the output, z_i, of i for a given pattern, $\boldsymbol{x}$, is

$$z_i(\boldsymbol{x}) = \exp[-(\boldsymbol{x} - \boldsymbol{\mu}_i)^{\mathrm{T}} (\boldsymbol{x} - \boldsymbol{\mu}_i) / (2\sigma_i^2)] \tag{6}$$

PE j in the output layer generates the normalized sum, y_j, of the z_i product and their respective weights;

$$y_j = \sum_i z_i w_{ij} \Big/ \sum_i z_i \tag{7}$$

The values $\boldsymbol{\mu}$ and σ may analogously be viewed as the mean and standard deviation of the response curve, respectively. The response function of an RBF PE diminishes rapidly as a pattern deviates from the PE's mean. The set of pattern layer PEs is designed so that their responses cover all significant regions of the input vector space. In the simplest case, both the pattern layer and output layer weights remain fixed; there is no training at all. Further, there is one pattern layer PE for every design vector. In a slightly more complex extension, only the output layer weights are trained; this is a straightforward, and rapid, training of a single layer linear system. A further extension includes training the pattern layer weights as well as the location and shape of the response curves, which is done in this study.

There are several alternatives for determining the location of the centers of the receptive fields of the pattern layer PEs. The simplest alternative is to have one PE for every pattern in the design set. However, this may become completely impractical if there are a large number of design patterns; the amount of time required to train such a network as well as to test it would be inordinately great. A more robust strategy is to take advantage of the fact that design pattern typically tend to occur in clusters, and use an unsupervised clustering algorithm to reduce the number of pattern layer PEs.

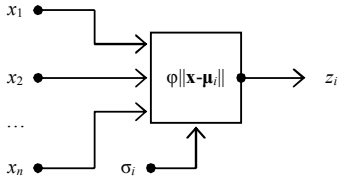

Fig. 3: The structure of an RBF pattern layer processing element.

Used in this investigation, standard *k*-means clustering is one possible approach to compute a $\boldsymbol{\mu}$ set. This algorithm assumes that all of the design input patterns are available and that there are a pre-determined fixed number, *k*, of clusters (centers). The standard *k*-means algorithm will ensure that the sum of the squares of the (Euclidean) distances between each design input pattern and its closest (receptive field) center is a local minimum.

The algorithm begins with a set of *k* random centers. Each design input pattern is examined to determine the closest center to it. A new set of centers is computed by taking the average of all design input patterns, for each center, and using those averages as the new centers. This step is repeated for a fixed number of iterations or until the membership function no longer changes.

The radius of the receptive region of y_i is determined by σ_i. If the $\boldsymbol{\mu}_i$ set is widely separated then the σ_i should be large to cover the gaps. If they are tightly packed then the σ_i should be small enough to accurately retain the distinctiveness of each receptive field. A conventional technique that may be used to determine the σ_i is to use the *P*-nearest neighbor heuristic. Given a receptive region's center, $\boldsymbol{\mu}_i$, let $i_1,i_2,\ldots,i_p$ be the indices of the *P* centers nearest to $\boldsymbol{\mu}_i$. Then the corresponding σ_i is

$$\sigma_i = \sqrt{P^{-1}\sum_{p=1}^{P}\left\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_{i_p}\right\|^2} \tag{8}$$

In this study, *P*=1.

Once the $\boldsymbol{\mu}_i$ and σ_i have been selected, the output layer weight matrix may then be optimized. A standard technique is to use a supervised training strategy such as gradient descent learning. Most of the y_i will be close to zero for a given input pattern since that pattern will be near only one receptive field. As a consequence, the corresponding weight changes will be small. To improve training time, this fact can be exploited by ignoring the receptive fields with small activations.

An ill-advised strategy to determine the values of the weights is to treat the problem as a solution of the matrix equation, $\boldsymbol{W}=\boldsymbol{Y}^{-1}\boldsymbol{D}$; where $\boldsymbol{W}$ is the weight matrix, $\boldsymbol{D}$ is a matrix whose rows are the desired classification outcomes, and $\boldsymbol{Y}$ is a matrix whose rows are the outputs from the output layer for each design set pattern. The matrix $\boldsymbol{Y}$ is generally not invertible because it is typically not square. Further even if a pseudoinverse exists it may not be easily determined [12]. The matrix may be ill-conditioned because it is singular or nearly singular. Even techniques such as singular value decomposition may fail because of the possible limited accuracy of the results.

B. Fuzzy Interquartile Encoding

Fuzzy interquartile encoding (FIQ) [13] is a preprocessing strategy that has been successfully applied to a number of different classification problems [14][15]. It involves taking a feature value and intervalizing it across a collection of fuzzy sets, thereby producing a list of degrees of membership for each of the fuzzy sets [16]. Selecting intervals for the fuzzy sets is usually an experimental or heuristic process and is similar to the techniques used in standard 1–of–*k* intervalization encoding. In this study, we use a dimension-preserving version of fuzzy interquartile encoding. Instead of constructing four triangular fuzzy sets around the quartiles of each feature, a single piece-wise linear fuzzy set is constructed whose vertices are the lower quartile, Q_L, median, *m*, and upper quartile, Q_U.

Fig. 4 illustrates an example of a single membership function, f_i, constructed from the quartiles of feature, x_i (where α and β are the respective minimum and maximum design set values for the feature). Given a feature value, *x*, and the membership threshold $h\in(0,1)$

$$f_i(x) = \begin{cases} h(x-\alpha)/(Q_L-\alpha) & \alpha < x < Q_L \\ h+(1-h)(x-Q_L)/(m-Q_L) & Q_L \le x < m \\ 1+(h-1)(x-m)/(Q_U-m) & m \le x < Q_U \\ h-h(x-Q_U)/(\beta-Q_U) & Q_U \le x < \beta \\ 0 & x \le \alpha \vee x \ge \beta \end{cases} \tag{9}$$

FIQ exhibits the useful property that the transformed feature space is "normalized" in that for any given feature, *x*, its corresponding membership function maps feature values onto the unit interval. This is particularly useful in the classification process since scaled data tend to stabilize the effects of extreme variance disparities across features [17]. Without scaled data, features with large variances will predominate, during the design phase, over features with small variances even though the latter features may be highly discriminatory. Another useful property is that, when *h*=0.5, there exists a strict 1–1 correspondence between the fuzzy encoding and the original feature value.

C. Stochastic Feature Selection

The motivation for classification preprocessing strategies exploiting feature subset selection is to simplify the determination and construction of optimal decision boundaries separating input patterns from different classes. Feature subset selection involves finding a mapping $g':\boldsymbol{X}\rightarrow\boldsymbol{X}'$, where $\boldsymbol{X}'\subseteq\Re^l$ ($l<<n$) is the reduced feature space. Pattern classification involves the subsequent determination of a mapping from the reduced feature space to the space of pattern class labels, $g:\boldsymbol{X}'\rightarrow\Omega$. An example of such a preprocessing strategy is stochastic feature selection (SFS) [18].

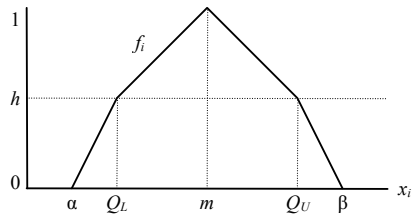

Fig. 4: The membership function, f_i, using the lower- (Q_L), mid- (*m*), and upper-quartiles (Q_U) of feature x_i (α and β are the respective minimum and maximum design set values).

SFS is a dimensionality reduction technique, which begins with the random assignment of the original dataset patterns into design and test sets. Once the design phase is complete, the test set is used to externally substantiate the classification performance. Coupled with n-fold internal validation, this provides a reliable measure of the effectiveness of the underlying classification system. The number of regions and the feature region length to be selected are bounded by the user. Feature regions may be specified to be either disjoint or overlapping. Transformations may also be performed on regions such as computing their mean, variance, or other statistical moment.

SFS exploits the quadratic combination of feature regions. The intent is that if the original feature space had non-linear decision boundaries between classes, the new (quadratic) parameter space may have decision boundaries that are more linear. SFS has three categories of quadratic combinations: using the original feature region; squaring the feature values for the region; and using all pair-wise cross-products of features from two regions. The probabilities of selecting one of these quadratic feature combination categories must sum to 1.

The stochastic nature of SFS is normally controlled by a feature frequency histogram whereby the performance of each classification task is assessed using the selected fitness function. If the fitness exceeds the histogram fitness threshold, the histogram is incremented at those feature indices corresponding to the regions used by the classification task. This histogram is used to generate an ad hoc cumulative distribution function, which is used when randomly sampling new feature regions. So, rather than each feature having an equal likelihood of being selected for a new classification task, those features that were used in previous "successful" classification tasks have a greater likelihood of being chosen. A temperature term, $0 \leq t \leq 1$, provides additional control over this process. If $t=0$, the ad hoc cumulative distribution function is used but, as $t \rightarrow 1$, the randomness becomes increasingly uniform (when $t=1$ a strict uniform distribution is used).

SFS was implemented using Scopira [19], an algorithm development framework, which allows the interconnection of multiple algorithm "modules" (for example, classifiers and data preprocessing techniques) [20]. SFS takes advantage of parallel computations using the MPI message-passing library specification [21]. Given a computing cluster, classification tasks are distributed to slave nodes for computation. A master node coordinates the distribution of tasks, records intermediate performance results, and updates the feature frequency histogram and cumulative distribution function. To minimize inter-process communication and maximize CPU loads, SFS "bundles" sets of classification tasks. Furthermore, while SFS exploits parallelism, it remains (optionally) strictly deterministic. That is, experimental results are perfectly reproducible regardless of computational load.

The classification tasks were executed on a homogeneous 14 node dual Opteron Linux Beowulf cluster system interconnected via TCP with 9GB of RAM per node. To conclude this section we now summarize the SFS algorithm.

SFS algorithm
1. Select values for relevant parameters.
2. Select classifier and fitness function.
3. Divide dataset into design and test sets.
4. *Repeat*: Randomly divide design set into training and monitoring components.
5. Randomly select a feature subset using feature frequency histogram.
6. Perform quadratic transformation of feature subset.
7. Train a classifier using the design set training component.
8. Use fitness function and the monitor set to assess performance.
9. If performance exceeds histogram threshold, update frequency histogram.
10. If performance exceeds current best classifier, update list.
11. If maximum number of iterations or fitness threshold is exceeded, *stop*.
12. Using the test set, compare the predictions against the actual class labels.

III. Material

A. Magnetic Resonance Spectroscopy

Magnetic resonance (MR) spectroscopy [22][23], used in many biomedical applications, exploits the interaction between an external homogeneous magnetic field and a nucleus that possesses spin in order to produce signals of tissues or biofluids under investigation. Certain nuclei, for example, those of hydrogen (^{1}H), carbon (^{13}C), and phosphorus (^{31}P) resonate when exposed to electromagnetic radiation at a particular frequency, which depends on the type of nucleus, the molecular environment, and the intensity of the surrounding magnetic field. This reliable and versatile spectroscopic modality is routinely used in the classification of biomedical spectra for many types of tissues and biofluids. However, the analysis of MR spectra is also problematic due to the presence of artifacts and a low signal-to-noise ratio. The presence of signals from a large number of metabolites makes the extraction of relevant discriminatory information difficult. Finally, it is not uncommon that one is faced with the curse of dimensionality where the ratio of sample size to feature dimensionality is extremely small.

B. Dataset

A set of $N=191$ spectra of a biological fluid were acquired from an MR spectrometer with $n=3381$ spectral features. The spectra were divided into 116 normal patterns and 75 abnormal patterns. The design set comprised 55 normal and abnormal patterns for a total of 110 design patterns. The remaining 81 patterns were used in the test set (61 normal and 20 abnormal patterns).

IV. Experiment Results

We now describe the parameter values used for the experiment discussed below. A total of 10 basis functions were used for each RBF and $h=0.5$ for the fuzzy interquartile encoding. The fitness threshold was set at 0.999 and the maximum number of iterations was 10^6 (note that the fitness threshold was not exceeded in either run). The κ-score with

sensitivity and specificity adjustments was used as the fitness function (see Section II) and 10-fold internal validation was used. For SFS, 2–20 disjoint feature regions were selected. These regions were the average of 10–100 features. The feature frequency histogram threshold was set to 0.8 and t=0.25. With respect to the selection of quadratic feature combinations: 60% of the features were unaltered, 20% were squared, and 20% involved the pair-wise cross-product of another feature region.

TABLE 1 lists the classification accuracy, based on the test set, for the SFS runs using the original MR spectral features and the same features preprocessed using FIQ. An 8% decrease in the number of classification errors was achieved when FIQ was used, clearly demonstrating the effectiveness of FIQ as a preprocessing strategy. The false positive error rate was about the same (~24%) using both the FIQ and original MR spectral features. TABLE 1 shows that the improvement was achieved through a 10% reduction in the false negative error rate.

Fig. 5 shows the feature frequency histograms for the MR spectra using the original features (Fig. 5(i)) and the FIQ encoded spectral features (Fig. 5(ii)). Note that far fewer (two orders of magnitude) RBF classifiers exceeded the SFS frequency histogram fitness threshold (0.8) using the original features compared to the SFS run using the FIQ encoded features. In other words, significantly more RBF classifiers produced "successful" classification results using FIQ features compared to the original spectral features. This is likely also reflected in the smaller standard deviations for the FIQ encoded features in TABLE 1.

TABLE 2 lists the spectral feature regions used with the best performing RBFs for the original features and the FIQ encoded features. In both cases, five averaged feature regions were used and roughly the same total number of spectral features were also used (286 original features and 275 FIQ encoded features). It is interesting to note that with the original features only one of the feature regions involved a quadratic combination (pair-wise product), while three feature regions involved quadratic combinations using the FIQ encoded features (squared terms). This demonstrates the effectiveness of using quadratic combinations of features in the new parameter space compared to using the original input feature space, regardless of whether or not that input space was transformed using FIQ.

For completeness, we also performed conventional moving averages of the spectral features using window sizes of 23, 49, and 69, yielding feature dimensionalities of 147, 69, and 49, respectively. RBF was used on these averaged datasets. Classification results for all three datasets were significantly poorer than the results listed in TABLE 1 with the best result of only 61.1% using 147 averaged features.

TABLE 1
TEST SET CLASSIFICATION ACCURACY

	Accuracy (%)		
Features	Normal	Abnormal	Overall
Original	71.8±3.8	76.0±4.1	72.8
FIQ	80.0±2.9	75.5±3.0	78.9

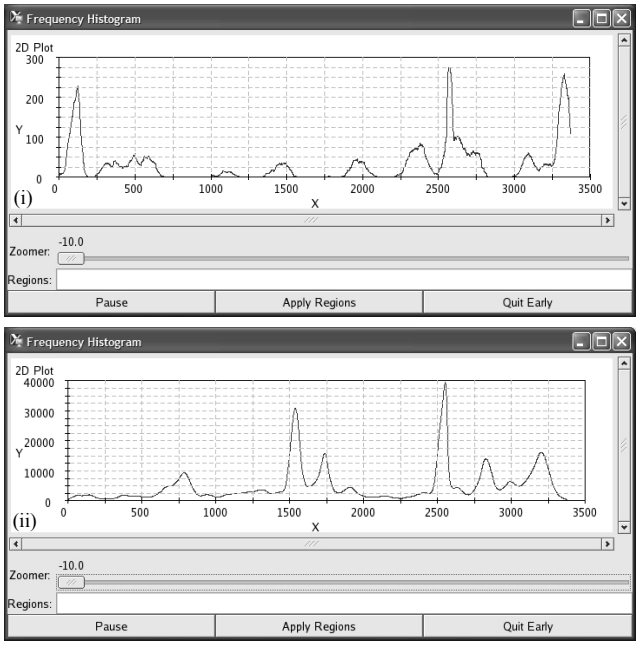

Fig. 5. SFS feature frequency histograms using RBF for (i) the original MR spectral features and (ii) the FIQ encoded features.

TABLE 2
FEATURE REGIONS SELECTED

	Feature Region		
	Start Index	Length	Quadratic
Original	93	69	Pair-wise[a]
	434	78	None
	2313	61	None
	2560	40	None
	3307	38	None
FIQ	1510	70	Square
	1582	38	None
	2516	62	Square
	2787	14	Square
	3192	91	None

[a] cross-product with region 2560–2599.

V. CONCLUSION

We empirically demonstrated the effectiveness of a classification preprocessing strategy, a combination of fuzzy set theoretic interquartile encoding of the original feature space and the stochastic selection of highly discriminatory feature subsets, which was employed with a voluminous biomedical dataset of normal and abnormal patterns acquired from a magnetic resonance spectrometer. Using this preprocessing strategy with a radial basis function classifier, an 8% reduction in the overall error rate was achieved compared to using the original spectral features. Moreover, this was achieved using only a small fraction of the original features.

ACKNOWLEDGMENT

The authors wish to thank Conrad Wiebe for implementing the stochastic feature selection algorithm and Aleksander Demko for his development efforts with Scopira framework.

REFERENCES

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

[2] J.L. Fleiss, "Measuring agreement between judges on the presence or absence of a trait," *Biometrics*, vol. 31, pp. 651–659, 1975.

[3] J.R. Landis and G.G. Koch, "The measurements of observer agreement for categorical data," *Biometrics*, vol. 33, pp. 159–174, 1997.

[4] H.C. Kraemer, *Evaluating Medical Tests: Objectives and Quantitative Guidelines*, Newbury Park: Sage, 1992.

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

[6] J. Moody and C.J. Darken, "Fast learning networks of locally–tuned processing units," *Neural Computation*, vol. 1, pp. 381–294, 1989.

[7] F. Girosi, T. Poggio, and B. Caprile, "Extensions of a theory of networks for approximations and learning: outliers and negative examples," in *Advances in Neural Information Processing Systems*, J.E. Moody and D.S. Touretzky, Eds. San Mateo: Morgan Kaufmann Publ., 1991 pp. 750–756.

[8] S. Lee and R.M. Kil, "A Gaussian potential function network with hierarchically self–organizing learning," *Neural Networks*, vol. 4, pp. 207–224, 1991.

[9] J. Platt, "A resource-allocating network for function interpolation," *Neural Computation*, vol. 3, pp. 213–225, 1991.

[10] E. Hartman and J.D. Keeler, "Predicting the future: advantages of semilocal units," *Neural Computation*, vol. 3, pp. 566–578, 1991.

[11] S.M. Weiss and C.A. Kulikowski, *Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*, San Mateo: Morgan Kaufmann Publ., 1991.

[12] M.S. Srivastava and E.M. Carter, *An Introduction to Applied Multivariate Statistics*, New York: North–Holland, 1983.

[13] N.J. Pizzi, "Bleeding predisposition assessments in tonsillectomy/ adenoidectomy patients using fuzzy interquartile encoded neural networks," *Artificial Intelligence in Medicine*, vol. 21, pp. 65–90, 2001.

[14] N. Pizzi, "Classification of storm events using a fuzzy encoded multilayer perceptron," *Proceedings of the International Joint Conference on Neural Networks*, Como, Italy, July 2000, pp. 547–550.

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

[16] L.A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, 1973, pp. 28–44.

[17] B.D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press, 1996.

[18] N.J. Pizzi and W. Pedrycz, "Predicting qualitative assessments using fuzzy aggregation," *Proceedings of the North American Fuzzy Information Processing Society Conference*, June 3–6, Montréal, Canada, vol. MPM3, pp. 12–17, 2006.

[19] A.B. Demko, N.J. Pizzi, and R.L. Somorjai, "Scopira – A system for the analysis of biomedical data," *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering*, May 12–15, Winnipeg, Canada, pp. 1093–1098, 2002.

[20] A. Demko, R.A. Vivanco, and N.J. Pizzi, "Scopira: An open source C++ framework for biomedical data analysis applications — a research project report," *Companion Proceedings of OOPSLA, 17th ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*, October 16–20, San Diego, USA, pp. 138–139, 2005.

[21] M. Snir and W. Gropp, *MPI: The Complete Reference, 2nd Edition*, Cambridge: MIT Press, 1998.

[22] H. Friebolin, *Basic one- and two-dimensional NMR spectroscopy*, New York: John Wiley and Sons, 2005.

[23] D.L. Pavia, G.M. Lampman, and G.S. Kriz, *Introduction to Spectroscopy*, Fort Worth: Harcourt Brace College Publishers, 1996.