International Journal of Approximate Reasoning 51 (2010) 883–894

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Aggregating multiple classification results using fuzzy integration and stochastic feature selection

Nick J. Pizzi [a,b,*], Witold Pedrycz [c]

[a] National Research Council of Canada, Winnipeg, MB, Canada R3B 1Y6
[b] Department of Computer Science, University of Manitoba, Winnipeg, MB, Canada R3T 2N2
[c] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada T6G 2N4

ARTICLE INFO

Article history:
Available online 25 May 2010

Keywords:
Data classification
Fuzzy sets
Pattern recognition
Fuzzy integrals
Feature selection
Computational intelligence

ABSTRACT

Classifying magnetic resonance spectra is often difficult due to the curse of dimensionality; scenarios in which a high-dimensional feature space is coupled with a small sample size. We present an aggregation strategy that combines predicted disease states from multiple classifiers using several fuzzy integration variants. Rather than using all input features for each classifier, these multiple classifiers are presented with different, randomly selected, subsets of the spectral features. Results from a set of detailed experiments using this strategy are carefully compared against classification performance benchmarks. We empirically demonstrate that the aggregated predictions are consistently superior to the corresponding prediction from the best individual classifier.

1. Introduction

Magnetic resonance (MR) spectroscopic experimental techniques, which exploit the interaction between an external homogenous magnetic field and a nucleus that possesses spin, are often used to create non-invasive, robust diagnostic clinical procedures for the analysis of biofluids and tissues [1–3]. Unfortunately, the curse of dimensionality typically plagues these biomedical spectra; the dimensionality of the spectral feature space is high, $O(10^3–10^4)$, while the sample size is small, $O(10^1–10^2)$. Combined, these two characteristics present a significant challenge for the classification of biomedical spectra as the inordinate degrees of freedom during the design (training) phase easily cause overfitting that affects the reliability of the classification outcomes [4]. While a careful selection of a classifier is important in dealing with this issue, equally significant are the pre- and post-processing strategies to be employed.

Feature selection is a typical pre-processing strategy for dealing with this curse of dimensionality by reducing, in some fashion, the dimensionality of the feature space to an order similar to the sample size. The strategy used here is *stochastic feature selection* [5] that randomly selects feature subsets using a frequency histogram of the discriminatory power of each feature as stochastically determined by previous classification iterations. The feature subsets, which may undergo a quadratic transformation, are used to construct corresponding classifiers. After a classification accuracy threshold is reached, or a maximum number of iterations is exceeded, the best sets of feature subsets, and their respective classification outcomes, are retained.

Classification may sometimes involve the aggregation of predictions as a post-processing strategy [6]. An aggregation technique combines the predicted classification outcome (here, a disease state class label), for each pattern (here, an MR spectrum), from the best set of individual outcomes. A typical aggregation technique involves the use of a fuzzy integral,

[*] Corresponding author at: National Research Council of Canada, 435 Ellice Avenue, Winnipeg, MB, Canada R3B 1Y6. Tel.: +1 204 983 8842; fax: +1 204 983 3154.
E-mail addresses: pizzi@nrc-cnrc.gc.ca, pizzi@cs.umanitoba.ca (N.J. Pizzi), pedrycz@ee.ualberta.ca (W. Pedrycz).

0888-613X/$ - see front matter Crown Copyright © 2010 Published by Elsevier Inc. All rights reserved.
doi:10.1016/j.ijar.2010.05.003

a non-linear numeric approach to combining multiple information sources to arrive at a "confidence value" for a decision. While several interpretations exist for the meaning of a fuzzy integral [7], in this study, it is considered to mean the maximum degree of belief (for a predicted classification outcome) obtained by the fusion of several sources of objective evidence (credibility).

In this investigation, these two pre- and post-processing strategies are combined in order to collate feature subsets for presentation to a set of respective classifiers whose independent predictions are aggregated using fuzzy integration. The motivation behind this approach is twofold: often, only a subset of features possesses discriminatory power while the remainder has a tendency to confound the effectiveness of the underlying classifier; often, a prediction based on an aggregated consensus from a set of classifiers operating upon different feature subsets will be more accurate than the prediction of any individual classifier.

The paper will be divided into several sections beginning with a discussion of the stochastic feature selection strategy including: the underlying classifier types; the random sampling approach using the feature frequency histogram; the validation procedure; and, parameters and stopping criteria. The next section describes the aggregation technique and the fuzzy integrals that were tested. This is followed by a description of the heterogeneous classifiers used in the experiments. The characteristics of the MR spectral dataset will then be described. The classification results coming from the independent classifiers are aggregated via fuzzy integration to produce a final consensus, the predictive power of which is assessed using an external validation dataset. Results are presented with the aggregated results benchmarked against the corresponding best individual classifier. We empirically demonstrate that the aggregated predictions are consistently superior to the corresponding prediction from the best individual classifier.

2. Stochastic feature selection

Consider a c-class classification problem in which $\boldsymbol{X} = \{(\boldsymbol{x}_k, \omega_k), k = 1,\ldots,N\}$ is a set of N labeled patterns (MR spectra). Here, $\boldsymbol{x}_k \in \Re^n$ and $\omega_k \in \Omega$, where $\Omega = \{1,\ldots,c\}$. A classifier may be viewed as a mapping f: $\boldsymbol{X} \to \Omega$. Let ω_{ip} be the class label predicted by classifier p for pattern, x_i. If $\omega_{ip} = \omega_i$, classifier p has generated a correct classification (prediction) result for $\boldsymbol{x}_i$. The motivation for pre-processing strategies exploiting feature selection [8–10] is to simplify the determination and construction of optimal decision boundaries for classification problems. Formally, feature selection involves finding a mapping g': $X \to X'$, where $X' \subseteq \Re^m (m \ll n)$ is the reduced feature space. Classification involves the subsequent determination of a mapping from the reduced feature space to the space of class labels, g: $X' \to \Omega$.

Stochastic feature selection (SFS) is a dimensionality reduction technique used in problems of classification. SFS may be used with any homogeneous or heterogeneous set of classifiers. 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. SFS randomly assigns the original dataset samples (in this case, MR spectra) into design and test sets. Once the design phase is complete, the test set is used to validate the classification performance. Coupled with internal n-fold validation, this provides a reliable measure of the effectiveness of the underlying classification system. During the design phase, SFS generates classification coefficients and assesses their performance.

Fig. 1 is a flowchart for the SFS approach. The first step involves parameter initialization. The user selects the minimum and maximum number of feature regions and the minimum, a, and maximum, b, sizes (cardinality) for a feature region. For a pattern, $\boldsymbol{x} = [x_1,\ldots,x_n]$, a feature region is defined to be a contiguous subset of its features, $\boldsymbol{x}^{\alpha\beta} = [x_\alpha,\ldots,x_\beta]$ $(1 \leqslant a \leqslant \alpha \leqslant \beta \leqslant b \leqslant n)$. Feature regions may be either disjoint or overlapping. The user may also choose to transform the regions by computing their mean, variance, or other statistical moment. The feature regions may also be quadratically transformed (see below). Other parameters include: those specific to each selected classifier type; sampling rate for each classifier type; fitness function used to evaluate performance; stopping criteria (accuracy threshold, P_ε, and maximum number of iterations, η); and the number, k, of top performing classifier instances to retain.

The remaining steps in the flowchart (except the last one) are iteratively performed until one of the stopping criteria is satisfied. The first block of steps (see Fig. 1) involves: (i) selection of the classifier instance; (ii) selection of a candidate set of feature regions from the dataset's original features; and (iii) possible transformation of the selected feature regions. The classifier instance is randomly selected from one of several classifier types based on their sampling distribution (pre-determined by the user). Here, linear discriminant analysis, radial basis function neural network, and probabilistic neural networks (see Section 4) were used with equal likelihood of being selected; however, any classifier type may be used with SFS. The set of feature regions is randomly selected (satisfying the above mentioned criteria) and all other features are pruned.

The second block of steps in SFS assesses the performance of each specific classifier instance (as selected in the previous block). First, the feature region set is randomly allocated to either a design set or a test set. Second, the classifier instance is trained using the design set feature regions to produce prediction coefficients. Third, its performance (classification accuracy) is assessed using the prediction coefficients with the test set feature regions. This block is repeated several times (n-fold validation) with different random allocations to design and test sets.

If the performance, P, of the current classifier instance exceeds the histogram fitness threshold then the feature frequency histogram (see below) is updated to reflect the fact that the feature regions contributed to a "successful" classification. Furthermore, if P exceeds the performance, P^i $(i = 1,\ldots,k)$, of any of the previous top k classifier instance list, the list is updated to include the current classifier instance.

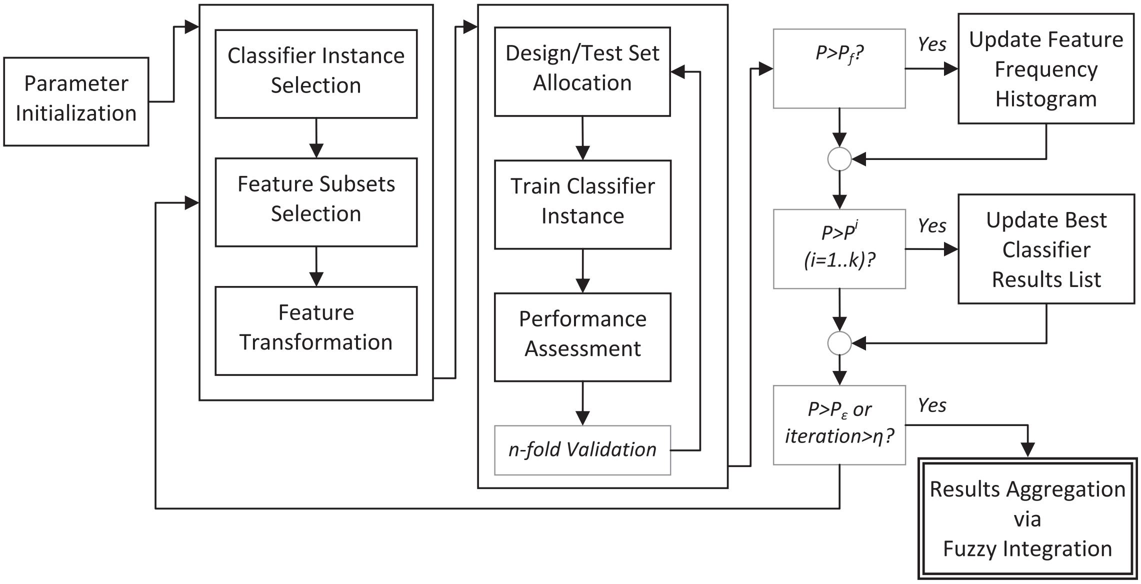
Parameter Initialization
Classifier Instance Selection
Feature Subsets Selection
Feature Transformation
Design/Test Set Allocation
Train Classifier Instance
Performance Assessment
n-fold Validation
P>P_f?
Yes
Update Feature Frequency Histogram
P>P^i (i=1..k)?
Yes
Update Best Classifier Results List
P>P_ε or iteration>η?
Yes
Results Aggregation via Fuzzy Integration

Fig. 1. Flowchart for stochastic feature selection and classifier integration.

The above SFS steps are repeated until: (i) $P > P_\varepsilon$; or (ii) the number of iterations exceeds η. The final step is to aggregate the classification results from the best k classifier instances using fuzzy integration as described in Section 3.2.

Classification accuracy, P, is measured using the $c \times c$ confusion matrix, $\boldsymbol{R}$, of desired versus predicted class labels. Several fitness functions may be used ($i, j = 1, \ldots, c$) including: $P_o = N^{-1}\sum_i \boldsymbol{R}_{ii}$, the (conventional) ratio of correctly classified patterns to total number of patterns; $P_A = c^{-1}\sum_i(\boldsymbol{R}_{ii}/\sum_j \boldsymbol{R}_{ij})$ the average classification accuracy for each class; or a chance-corrected measure of agreement ($P_L = N^{-2}\sum_i(\sum_j \boldsymbol{R}_{ij}\sum_j \boldsymbol{R}_{ji})$ is the agreement due to chance) such as the κ-score, $P_\kappa = (P_o - P_L)/(1 - P_L)$ [11].

The stochastic selection mechanism of SFS may be controlled by a feature frequency histogram whereby the performance of each classification task (the current selected classifier instance coupled with the current selected set of feature regions) is assessed using the selected fitness function. If the fitness exceeds the histogram fitness threshold, P_f, 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, $0 \leqslant t \leqslant 1$, 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 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; 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.

SFS was implemented using Scopira [12], a C++ software development library that provides facilities for high performance numerical computing, visual application development (2D/3D) and parallel processing [13]. Algorithm functionality may be mapped onto a computer cluster to better utilize CPU processing power while decreasing overall execution time. SFS takes full advantage of parallel computations using the Scopira Agent Library [14], a sophisticated message-passing library similar in functionality to MPI [15]. In the specific case of SFS, classification tasks are distributed to slave nodes for computation. A master node coordinates the distribution of tasks, records intermediate results, and updates the feature frequency histogram and cumulative distribution function. To minimize inter-process communication and maximize CPU loads, SFS "bundles" classification tasks. Furthermore, while SFS exploits parallelism, it remains (optionally) strictly deterministic. That is, experimental results are perfectly reproducible regardless of computational load, which is extremely important in the analysis, and interpretation of complex biomedical data.

3. Prediction using fuzzy integration

3.1. Fuzzy integrals

The fuzzy integral [16–18], which is based on a fuzzy measure [19] (a set function used to express the grade of fuzziness), is a non-linear aggregation scheme for combining multiple sources of information to arrive at a "confidence value" for a decision (hypothesis). Let us define a mapping h: $\boldsymbol{X} \to [0,1]$ where a finite ordered $\boldsymbol{X} = \{x_1, \ldots, x_n\}$ is of interest. In this investigation, we use the Sugeno, $Su(x)$, Choquet, $Ch(x)$, and Shilkret, $Sh(x)$, integrals [20–22]. The fuzzy integrals of h over $\boldsymbol{X}$ with respect to the Sugeno fuzzy measure [23], g_λ, are defined as:

$$\begin{aligned} &Su(x) = \vee_i[h(x_i) \wedge g_\lambda(X_i)], \\ &Ch(x) = \vee_i[(h(x_i) - h(x_{i-1}))g_\lambda(X_i)], \\ &Sh(x) = \vee_i[h(x_i) \cdot g_\lambda(X_i)], \end{aligned} \tag{1}$$

where $\boldsymbol{X}_i = \{x_1, \ldots, x_i\}$, x_i are descending ordered with respect to $h(x_i)$, the max and min operators are used respectively for disjunction and conjunction, and $h(x_0) = 0$. While several interpretations exist for the meaning a fuzzy integral [22,24], here it is considered to mean the maximum degree of belief (for a classification outcome) obtained by the fusion of several sources of objective evidence.

3.2. Combining classifier results

Integrating the results from multiple classifiers involves using their respective confusion matrices to compute the fuzzy densities for each of the classifiers in order to determine the fuzzy measures used in (1). To this end, the technique described in [25] will be followed primarily and is briefly described here. Let $\boldsymbol{R}_k = (n_{k\omega_i\omega_j})$ be the $c \times c$ confusion matrix for classifier, k, where $n_{k\omega_i\omega_i}$ is the number of ω_i MR spectra that were correctly classified by k and $n_{k\omega_i\omega_j}$ $(\omega_i \neq \omega_j)$ is the number of ω_i spectra that were incorrectly assigned to ω_j by k. The preliminary fuzzy density of ω_i with respect to classifier k, $0 < g^*_{k\omega_i} < 1$, is

$$g^*_{k\omega_i} = \frac{n_{k\omega_i\omega_i}}{\sum_{j=1}^{c} n_{k\omega_i\omega_j}}. \tag{2}$$

These densities must be adjusted to take into account the frequencies of correct and incorrect classifications within and across the set of classifiers. This leads to the following expressions

$$\delta_{k\omega_i\omega_j} = \begin{cases} 1, & \omega_i = \omega_j, \\ \frac{n_{k\omega_i\omega_i} - n_{k\omega_i\omega_j}}{n_{k\omega_i\omega_i}}, & \omega_i \neq \omega_j, \\ \varepsilon, & n_{k\omega_i\omega_i} < n_{k\omega_i\omega_j}, \end{cases} \qquad \gamma_{k\omega_i\omega_j} = \begin{cases} 1, & n_{k\omega_i\omega_j} < n_{l\omega_i\omega_j}, \\ \frac{n_{l\omega_i\omega_j}}{n_{k\omega_i\omega_j}}, & n_{k\omega_i\omega_j} \geqslant n_{l\omega_i\omega_j}, \\ \varepsilon, & n_{k\omega_i\omega_j} = 0, \end{cases} \tag{3}$$

where ε is a small positive value. The corrected fuzzy density, $g_{k\omega_i}$, may now be computed as

$$g_{k\omega_i} = g^*_{k\omega_i} \times (\delta_{k\omega_i\omega_r} \times \cdots \times \delta_{k\omega_i\omega_s})^{w_1} \times \left(\gamma_{k\omega_i\omega_r} \times \cdots \times \gamma_{k\omega_i\omega_s}\right)^{w_2}, \tag{4}$$

where w_1 and w_2, $(w_1 + w_2 = 1)$ are weighting factors and r and s are the indices of those classes for which classifier k produced the highest accuracy score. The first adjustment, $\delta \in (0,1]$, reflects the misclassifications within the confusion matrix for k. As the misclassifications increase, $\delta \to 0$ (the third condition in (3) is the degenerate case when more patterns of a particular class are misclassified than correctly classified). The second adjustment, $\gamma \in (0,1]$, reflects the misclassifications across all classifiers with respect to k. As the misclassifications increase, $\gamma \to 0$ (the third condition in (3) is the degenerate case when no patterns of a particular class are correctly classified). Finally, the Sugeno, Choquet, and Shilkret integrals will use several variants of h:

$$\begin{aligned} &h_c(x) = \begin{cases} 2x^2, & 0 \leqslant x \leqslant 0.5, \\ 1 - 2(x-1)^2, & 0.5 < x \leqslant 1, \end{cases} \\ &h_p(x) = x^p (p > 0), \end{aligned} \tag{5}$$

where $x \in [0,1]$ is the classifier's predicted class assignment and $h_c(x)$ is the standard definition for contrast intensification [26]. When $0 < x < 1$, $h(x)$ will act to dilate membership values, while concentration will occur when $x > 1$. In order to constrain the number of parameters in the experiments described in Section 6, we will use the standard fuzzy set based definitions for concentration ($p = 2$) and dilation ($p = 0.5$) [26]. In total, four variants will be used: $h_c(x)$, $h_{0.5}(x)$, $h_2(x)$, and $h_1(x)$ (identity). Fig. 2 shows the effect on membership values for these variants of $h(x)$. Finally, using (1)–(5), the actual class label output from the set of classifiers is the one with the highest integrated value.

4. Classifiers

In this section, we provide a brief discussion of the three different classifiers, linear discriminant analysis (LDA), radial basis function networks (RBF), and probabilistic neural networks (PNN), used in the experiments described in Section 6. All three classifier types exploit the notion of radial functions in the feature space. In PNN they are hyper-spherical Gaussians located at data point clusters, in LDA they are hyper-ellipsoidal Gaussians located at the class means, and in RBF they are usually hyper-spherical (or hyper-ellipsoidal) Gaussian, of a user-defined number, located at points in the data space that is determined by a gradient descent approach.

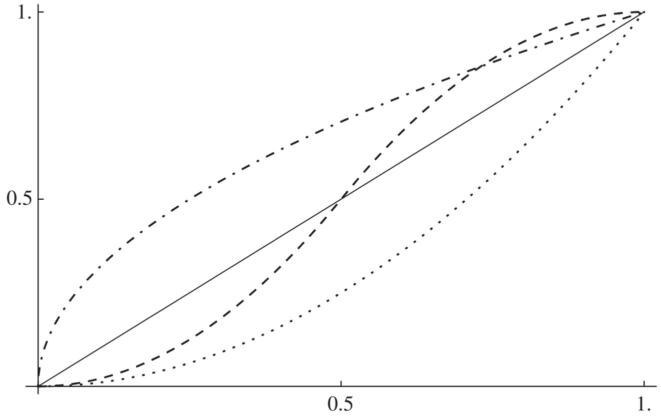

Fig. 2. The effect on membership values for different h: identity, $h_1(x)$ (solid line); contrast intensification, $h_c(x)$ (dashed line); concentration, $h_2(x)$ (dotted line); and dilation, $h_{0.5}(x)$ (dot-dashed line).

While we constrain the experiments to only use these classifier types, it should be noted that any type of analytic or iterative classifier may be used with the SFS strategy. Finally, as a benchmark experiment, we compare the results obtained using the SFS method to classification results using an implementation of a support vector machine (SVM).

4.1. Linear discriminant analysis

LDA [27] is a conventional classification strategy, which determines linear decision boundaries between c classes while taking into account between-class and within-class variances. LDA allocates a pattern, $\boldsymbol{x}$, to class d for which the probability distribution, $p_d(\boldsymbol{x})$, is greatest, that is, $\boldsymbol{x}$ is allocated to class d, if $q_d p_d(\boldsymbol{x}) \geqslant q_i p_i(\boldsymbol{x})$ $(\forall i \neq d)$, where q are the prior probabilities (or proportional probabilities). The discriminant function is

$$L_d(x) = \log q_d + \mu_d^T W_d^{-1} \left(x - \frac{1}{2} \mu_d \right), \tag{6}$$

where μ_d is the mean for class d and $\boldsymbol{W}_d$ is its covariance matrix. The feature space hyperplane separating class d from class i is defined by $F_{di}(\boldsymbol{x}) = L_d(\boldsymbol{x}) - L_i(\boldsymbol{x}) = 0$. If the error distributions for each class are the same, LDA will find the optimal linear class decision boundary; however, this optimality is seldom achieved with "real world" datasets since different classes typically give rise to different distributions. Nevertheless, LDA can be a useful linear classifier when combined with appropriate dimensionality reduction pre-processing techniques such as feature subset selection [28].

4.2. Radial basis function neural network

An RBF [29] is a neural network of receptive fields that possess radial symmetry, $f(\boldsymbol{x}) = \phi(\|\boldsymbol{x} - \text{ł}\mu\|)(\mu$ is the centre, $\|\cdot\|$ is a metric to determine the distance between a pattern and $\boldsymbol{\mu}$, and $\phi \in [0,1]$ approaches zero as the distance between a pattern and $\boldsymbol{\mu}$ increases. A typical definition for a receptive field, i, is

$$f_i(x) = \exp\left[-\frac{(x - \mu_i)^T (x - \mu_i)}{2\sigma_i^2} \right], \tag{7}$$

where σ_i is the diameter of the receptive region. Standard k-means clustering is used to find the location ($\boldsymbol{\mu}$) of the receptive fields, the P-nearest neighbour heuristic is used to find their shapes (σ), and a set of pattern layer weights are trained (for example, via back-propagation) [30]. As the receptive fields are localized, RBFs do not perform well if discriminatory features are globally distributed through the space.

4.3. Probabilistic neural network

PNNs are artificial neural networks that are well suited for data classification [31,32]. Class-labeled design set patterns are used to construct probability density functions (pdfs) to estimate the likelihood of a given pattern belonging to a given class. A useful PNN property is that as more training patterns are used, it converges to a Bayesian classifier [33]. A PNN may be constructed to correspond exactly to a Bayesian classifier if class pdfs are known; however, since this is rarely the case they may be approximated using Parzen estimators [34]. In general, it is not possible to determine the number of patterns required to estimate the pdf to a specified accuracy using this approach.

It is unnecessary to compute the approximated pdf but only the values at each pattern (n-dimensional point). If N_j is the number of class j patterns, the class j pdf value for pattern $\boldsymbol{x}$ (assuming normalized patterns) is

$$f_j(x) = \left(2\pi^{\frac{n}{2}}\sigma^n N_j\right)^{-1} \sum_{i=1}^{N_j} \exp\left(\frac{x^T x_{ij}}{\sigma^2}\right), \tag{8}$$

where $\boldsymbol{x}_{ij}$ is pattern i from class j, and $\sigma = N_j^{-b}$ $(0 < b < 1)$ is the estimator's smoothing parameter (as $\sigma \to 0$, the PNN approximates a nearest neighbour classifier). A pattern $\boldsymbol{x}$ is assigned to class j, if $h_j f_j(x) > h_i f_i(x)$ $(\forall i \neq j)$ where h_j is the proportional probability of a pattern belonging to class j.

4.4. Support vector machine

SVMs [35,36] are a family of supervised learning algorithms that select models that maximize the error margin of a training set. This approach has been successively used in many data classification problems [37]. 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 [38] could only be used for linearly separable problems, it may be extended by employing a "kernel trick" [39] that exploits the fact that a non-linear mapping of sufficiently high dimension can project the patterns to a new 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, for patterns $\boldsymbol{x}$ and $\boldsymbol{y}$, are [39]: power, $K(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y})^d$; polynomial, $K(\boldsymbol{x},\boldsymbol{y}) = (\mathrm{a}\boldsymbol{x} \cdot \boldsymbol{y} + \mathrm{b})^d$; sigmoid, $K(\boldsymbol{x},\boldsymbol{y}) = \tanh(\mathrm{a}\boldsymbol{x} \cdot \boldsymbol{y} + \mathrm{b})$; and Gaussian, $K(\boldsymbol{x},\boldsymbol{y}) = \exp(-1/2|\boldsymbol{x} - \boldsymbol{y}|^2/\sigma)$.

5. Experiment design

5.1. Dataset

We use $N = 191$ MR spectra of a biofluid assigned, by a medical expert, to one of two classes: normal (116 spectra) or abnormal (75 spectra). Each spectrum comprises $n = 3380$ features. For each classification task (see below), each spectrum was randomly allocated to one of two subsets: a design set ($N_d = 116$) with 58 normal and 58 abnormal spectra; and a test set ($N_t = 75$) with 58 normal and 17 abnormal spectra. Three transformed dataset variants were generated: first derivative; rank ordered; and first derivative with rank ordering. It should be noted that these transformations, including any transformations occurring during the classification runs employing SFS and fuzzy integration, maintain a direct 1:1 mapping to the original feature space. This is often an essential requirement to ensure that medical experts and biomedical researchers can map results back to the original discriminatory features in order to assess the relevance of the underlying metabolites and their relative concentrations in tissues or biofluids.

5.2. Parameters

Rather than using P_o, the standard ratio of correctly classified patterns to total number of patterns to measure performance, results are ordered using the κ-score, P_κ, a conservative chance-corrected measure of agreement [11]. In other words, P_κ is used to determine whether or not a particular classification run is considered "successful" (P_κ exceeds a pre-defined threshold). However, for the sake of simplicity, the final classification results (based on the test sets) are presented using the standard measure of agreement, P_o. That is, once the design phase is complete and the best test set classification outcomes, based on P_κ, are determined, the corresponding standard measure of agreement is then computed and presented in the tables in Section 6.

With respect to SFS, 2–10 averaged non-overlapping regions were used of length 10–50. It was equally likely that LDA, RBF, or PNN were selected for a classification run. The quadratic combination likelihoods were 70%, 15%, and 15% for using the original features, squaring them, or using pair-wise cross-products, respectively. The histogram sampling threshold was set to $P_f = 0.25$, $t = 0.1$ and 7-fold internal validation was performed. Concerning specific parameters for the classifiers: $b = 0.1$ and $P = 1$ for PNN; six receptive fields were used for RBF; and proportional probabilities were used for LDA. For the SVM benchmark, we use Platt's implementation using sequential minimal optimization [40] using the kernels described in Section 4.4 with a broad range of parameter values. For clarity, in Section 6 we report only the best classification results using the SVM benchmark. Finally, in the case of the corrected fuzzy densities in (4), $w_1 = 0.8$ and $w_2 = 0.2$ when using fuzzy integration to aggregate the best classification outcomes.

5.3. Outcomes analysis

Two major categories of experiments were performed. The first category involves testing the fuzzy aggregation approach described in Section 3 using SFS, the four dataset variants (as described in Section 5.1), and the Sugeno integral, $Su(x)$, with

$h_c(x)$. Here, detailed test results focus on: overall classification accuracy; the sets of discriminatory features discovered; and assessment of the different dataset transformations. As benchmarks, the aggregated prediction is compared against the corresponding (i) best individual classifier; (ii) aggregation using a simple majority voting scheme; and (iii) aggregation using a weighted vote (predictions are summed and the class with the largest sum becomes the prediction). The second category of experiments involves testing the classification performance of the three different fuzzy integrals, $Su(x)$, $Ch(x)$, and $Sh(x)$, in combination with $h_c(x)$, $h_1(x)$, $h_2(x)$, and $h_{0.5}(x)$. For this category, we constrain the analysis to only one dataset transformation, the rank ordered variant, which happened to produce the best overall classification outcomes from the first category of experiments. Here, the focus is on a comparison of overall classification performance between the 12 combinations of aggregators. For each experiment, SFS ran for 10^6 iterations retaining the best 10 classification runs, over which the fuzzy aggregation method was applied.

6. Results

6.1. Aggregation using the Sugeno integral

Fig. 3 shows the overall accuracies for the design set using the fuzzy aggregation approach with $Su(x)$ and $h_c(x)$ versus the best individual classifiers with: the original MR spectral features; the first derivative (FD) of the features; rank ordering (RO) of the features, and the first derivative combined with the rank ordering (FD + RO) of the features. Apart from FD + RO, the aggregated approach produced better predicted outcomes. While the three underlying classifiers (LDA, PNN, RBF) used in SFS had an equal likelihood of being selected for a classification run, slightly fewer "successful" runs ($P_f \geqslant 0.25$) involved RBF (30%) than PNN (34%) or LDA (36%).

Significantly more important for measuring the efficacy of this classification approach are the results based on the patterns (spectra) within the test set. Fig. 4 shows the classification accuracies based on the test set outcomes. The best result was 79% ($P_o = 0.79$, $P_A = 0.81$, $P_\kappa = 0.50$) correctly classified test set patterns (spectra) using the fuzzy aggregation approach with rank ordering of the original features. This is an 8% improvement over the corresponding best individual classifier ($P_o = 0.73$, $P_A = 0.77$, $P_\kappa = 0.42$). However, if we look at P_κ, we see a 19% improvement in performance. Further, the aggregated approach outperformed the corresponding best individual classifiers across all variants: respectively, 0.76/0.74, 0.74/0.62, 0.79/0.73, 0.75/0.73. The best individual outcome used the original features ($P_o = 0.74$, $P_A = 0.76$, $P_\kappa = 0.42$). Note that the majority vote and weighted vote aggregation methods produced results comparable to the corresponding best individual outcomes (except for the FD case where the weighted vote outperformed the best individual classifier); however, both benchmark aggregation methods produced poorer outcomes than the aggregation method using fuzzy integration. For completeness, Table 1 lists the test set confusion matrices for each transformed feature set (original, FD, RO, and FD + RO) with accuracies (P_o and standard deviation) of predicted normal (N) and abnormal (A) class labels versus desired class labels. It should be noted that due to 7-fold internal validation, matrices total 525 ($7N_t$) "patterns". Note that the standalone SVM benchmark had comparable classification results to the corresponding best individual classifiers using SFS but poorer than the aggregated outcomes using fuzzy integration.

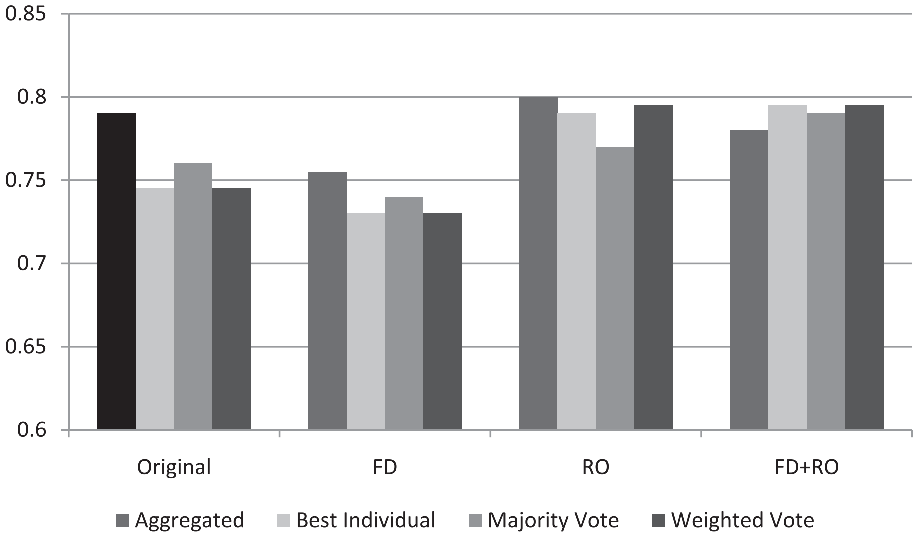

Fig. 3. Overall design set classification accuracy comparing aggregated and best individual classifier predictions using: the original features, first derivative (FD), rank ordering (RO), and FD with rank ordering (FD + RO).

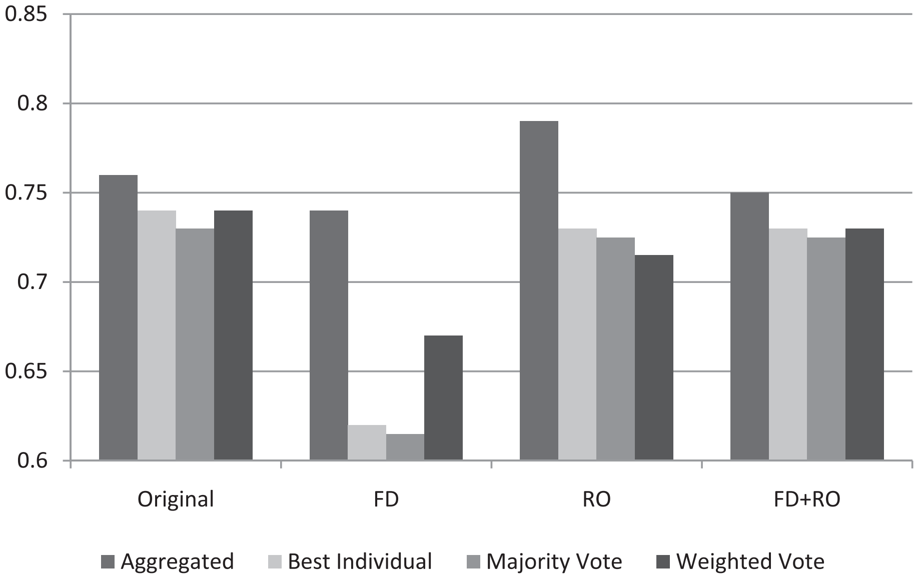

Fig. 4. Overall test set classification accuracy comparing SFS aggregated outcomes to the best individual classifier predictions, majority vote aggregation, and weighted vote aggregation.

Table 1
Test set results for aggregated versus best individual classifier predictions and the SVM benchmark (normal (N), abnormal (A)).

	Aggregated outcomes			Best individual outcomes			SVM benchmark		
	N	A	Accuracy	N	A	Accuracy	N	A	Accuracy
Original features									
N	306	100	0.75 ± 0.03	297	109	0.73 ± 0.05	294	112	0.72 ± 0.05
A	24	95	0.80 ± 0.10	25	94	0.79 ± 0.10	24	95	0.80 ± 0.08
Overall			0.76			0.74			0.74
First derivative									
N	307	99	0.76 ± 0.03	221	185	0.54 ± 0.08	231	175	0.57 ± 0.05
A	37	82	0.69 ± 0.15	12	107	0.90 ± 0.08	11	108	0.91 ± 0.10
Overall			0.74			0.62			0.65
Rank ordered									
N	314	92	0.77 ± 0.06	285	121	0.70 ± 0.06	279	127	0.69 ± 0.05
A	19	100	0.84 ± 0.11	19	100	0.84 ± 0.12	21	98	0.82 ± 0.12
Overall			0.79			0.73			0.72
First derivative, rank ordered									
N	295	111	0.73 ± 0.06	295	111	0.73 ± 0.07	290	116	0.71 ± 0.06
A	21	98	0.82 ± 0.07	30	89	0.75 ± 0.10	29	90	0.76 ± 0.09
Overall			0.75			0.73			0.72

Table 2 lists the discriminatory features that were selected using SFS indicating each region's starting index, length, and quadratic transform (for pair-wise, the second region's starting index and length is listed). [Regions for the aggregated outcomes are additional to the best individuals.] It should be noted that in all cases some of the highly discriminatory feature regions were quadratic in nature. Furthermore, the best individual classifier outcomes required only 3–5% of the original features, while the aggregated classifier results required 6–9%.

Recall that in Section 1, we mentioned the motivation behind the SFS approach were the suppositions that: (i) only a subset of features possesses discriminatory power (the rest are confounding); and (ii) a prediction based on an aggregated consensus from a set of classifiers operating upon different feature subsets will be more accurate than the prediction of any individual classifier. The previous observation on the number of features used leads to an interesting question as to whether the classifier type or the feature regions has a greater influence on the improvement to classification accuracy. To test the impact on classification accuracy of the classifier type and the feature regions, we conducted an experiment with each of the four dataset variants (original, RO, FD, and FD + RO) where we took the feature regions corresponding to the best LDA, RBF, and PNN classifier instances. For each variant, this gives us three sets of feature regions that we subsequently use to train consecutive instances of the three classifier types (this produces a set of nine classification outcomes). Now, we aggregate these outcomes using our original fuzzy integration method (using $Su(x)$ and $h_c(x)$) and compare them to

Table 2
Discriminatory features selected for each experiment (aggregated versus best individual outcomes).

Aggregated			Best individual		
Start index	Length	Quadratic combination	Start index	Length	Quadratic combination
Original features					
112	11	Squared	1031	17	Original
192	18	1543 (15)	1226	13	1676 (10)
855	14	Original	1676	10	Original
1516	20	Original	1758	13	Original
1543	15	Original	2336	16	Original
1588	10	Original	2768	13	Original
1671	17	Original	3006	13	Original
1775	10	Squared			
2107	16	Squared			
2284	13	Original			
First derivative					
413	15	Squared	146	14	Original
686	13	Original	1006	20	Original
786	13	Original	1309	11	2155 (13)
1162	20	Original	1420	19	1309 (11)
1949	19	Original	2155	13	Original
2131	17	Original	2174	12	Original
2720	16	Squared	2553	19	Squared
3281	18	Original	2606	11	Original
Rank ordered					
46	16	Original	10	16	Original
782	14	Squared	88	15	Original
860	16	Original	547	19	2745 (18)
1154	20	Squared	606	14	Original
1675	16	Original	1532	16	Original
2055	12	2162 (19)	2056	10	Original
2162	19	Squared	2584	19	Squared
2844	14	Original	2745	18	Original
			2769	15	Original
			3189	14	Original
First derivative, rank ordered					
649	20	Original	8	15	Original
741	14	Original	259	10	Original
786	17	Original	1229	18	Squared
1151	11	Squared	1570	12	Original
1260	10	Squared	1626	18	Original
1639	11	Original	1822	12	Original
2313	18	Original	2213	13	Squared
3145	10	Original	2570	11	Original

the SFS aggregation outcomes. If the results are comparable this would suggest that the classification improvement was the result of using the different feature regions rather than the classifier types. On the other hand, if the results are comparable to the corresponding best individual outcomes the improvement is likely as a result of the classifier types. However, if the results are somewhere in between the SFS aggregation outcomes and the best individual outcomes, the improvement is a result of both the classifier type and the feature regions. Fig. 7 plots the results of the SFS aggregation outcomes, best individual outcomes, and the alternate aggregation outcomes described above. While certainly not conclusive, the results suggest that both classifier type and feature regions play synergistic roles in improving the classification accuracy.[1]

Finally, Fig. 5(a)–(d) shows the feature frequency histogram for the best individual classifiers using the four feature transformations: the original features (a); FD (b); RO (c); and FD + RO (d).

6.2. Aggregation using fuzzy integral variants

We now turn to the second category of classification experiments in which the various aggregation methods are compared against each other. As discussed in Section 5.3, we constrain our analysis to the rank ordered feature set transformation, which produced the best classification outcome in Section 6.1. Fig. 6 shows the aggregated classification outcomes using all 12 fuzzy integration variants. Recall that in the previous section, the best classification accuracy, 79% ($P_o = 0.79$,

[1] We thank the second reviewer for this useful suggestion for assessing the roles of classifier type and feature regions in improving classification accuracy.

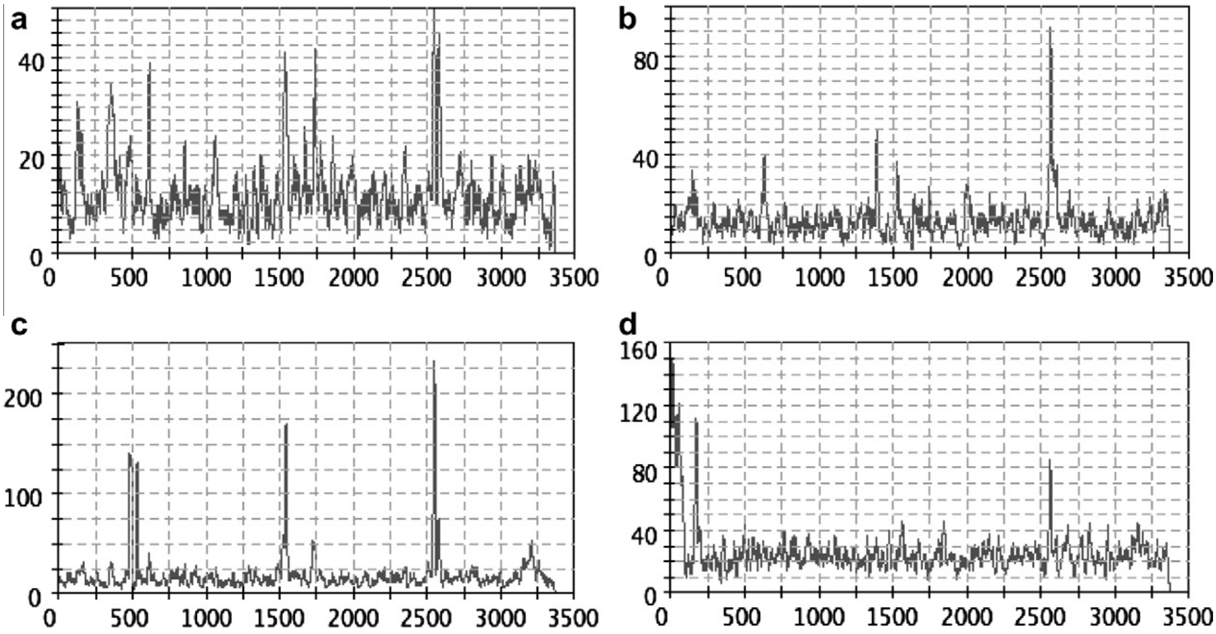

Fig. 5. The feature frequency histograms for the best individual classifiers using the original features (a), first derivative (b), rank ordering (c), and first derivative with rank ordering (d).

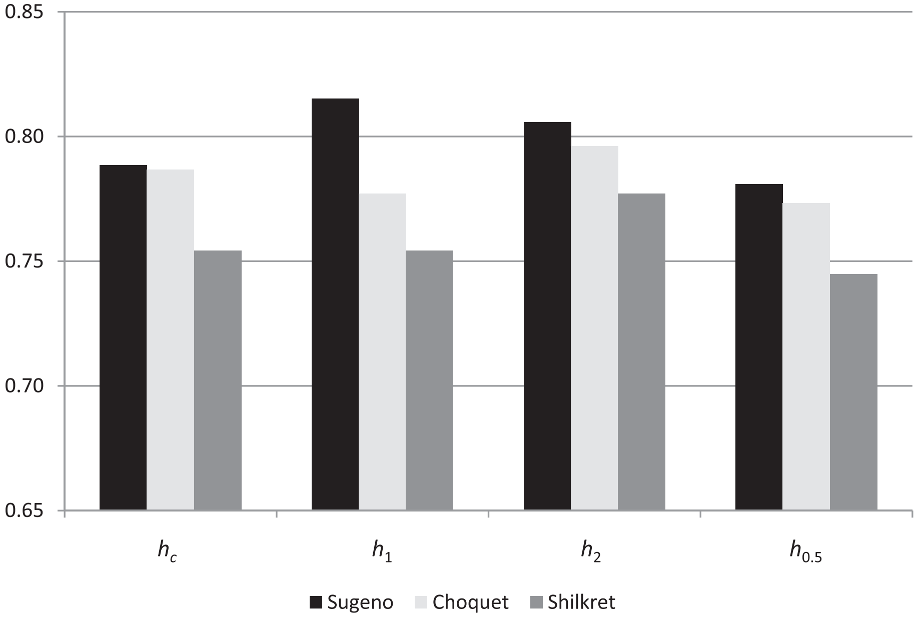

Fig. 6. Aggregated classification outcomes using all 12 fuzzy integration variants ($Su(x)$ with $h_c(x)$ produced the best outcome in Section 6.1).

$P_A = 0.81$, $\kappa = 0.50$), was achieved using $Su(x)$ and $h_c(x)$, which is represented as the left-most column in Fig. 6. Here, we see that the Sugeno integral using the identity function ($Su(x)$ and $h_1(x)$) produced an overall classification accuracy of 82% ($P_o = 0.82$, $P_A = 0.83$, $\kappa = 0.56$). Interestingly, the worst performing aggregation combination, $S_h(x)$ with $h_{0.5}$, did no worse than the overall best individual classification outcome (the one that used the original features): $P_o = 0.74$ ($P_A = 0.77$, $\kappa = 0.42$) and $P_o = 0.74$ ($P_A = 0.76$, $\kappa = 0.42$), respectively. Fig. 6 also suggests that: $Sh(x)$ generated more misclassifications than the other two fuzzy integrals regardless of h; $Su(x)$ produced the best classification outcomes across all variants of h; and, apart from the best outcome, h_2 was a consistently solid performer for all three integrals. For completeness, Table 3 lists the aggregation method confusion matrices for all 12 fuzzy integration variants.

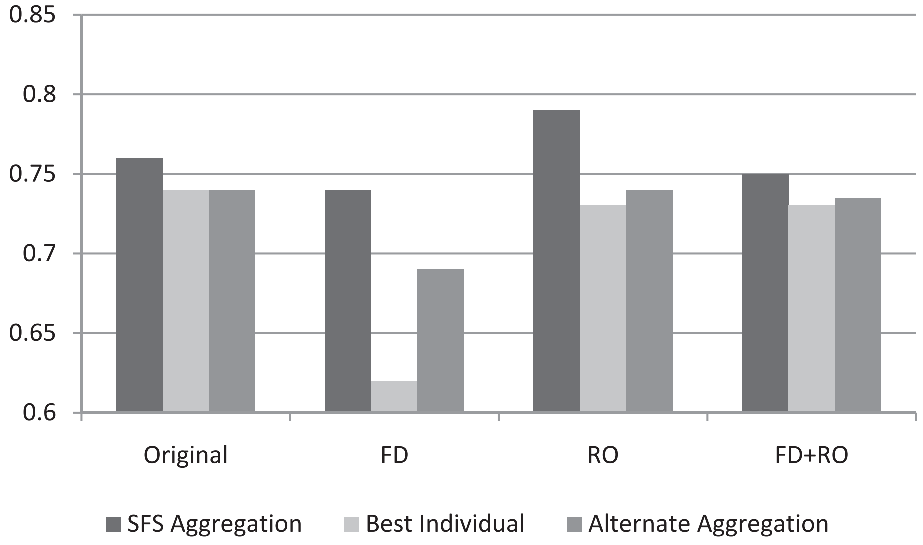

Fig. 7. Comparison of SFS aggregation with aggregation of classifier types with best feature regions.

Table 3
Test set results for 12 different fuzzy integration variants.

		Sugeno, *Su*(*x*)			Choquet, *Ch*(*x*)			Shilkret, *Sh*(*x*)		
		N	A	Accuracy	N	A	Accuracy	N	A	Accuracy
h_c	N	314	92	0.77 ± 0.06	312	94	0.77 ± 0.06	300	106	0.74 ± 0.07
	A	19	100	0.84 ± 0.11	18	101	0.85 ± 0.10	23	96	0.81 ± 0.12
	Overall			0.79			0.79			0.75
h_1	N	325	81	0.80 ± 0.04	310	96	0.76 ± 0.06	298	108	0.73 ± 0.07
	A	16	103	0.87 ± 0.11	21	98	0.82 ± 0.11	21	98	0.82 ± 0.12
	Overall			0.82			0.78			0.75
h_2	N	322	84	0.79 ± 0.06	319	87	0.79 ± 0.05	309	97	0.76 ± 0.06
	A	18	101	0.85 ± 0.11	20	99	0.83 ± 0.10	20	99	0.83 ± 0.12
	Overall			0.81			0.80			0.78
$h_{0.5}$	N	310	96	0.76 ± 0.05	308	98	0.76 ± 0.07	295	111	0.73 ± 0.08
	A	19	100	0.84 ± 0.12	21	98	0.82 ± 0.10	23	96	0.81 ± 0.12
	Overall			0.78			0.77			0.74

7. Conclusion

In this investigation, we developed an aggregation strategy using fuzzy integration to successfully classify magnetic resonance spectra of a biological fluid. The best classification results from a parallelized stochastic feature subset selection algorithm were aggregated using the original spectral features and three feature space variants. Two categories of experiments were used to demonstrate the efficacy of this approach. First, the aggregation method was constrained to only one fuzzy integral and compared to the corresponding best individual classification outcome using all four feature spaces. There was a 19% improvement in the chance-corrected measure of agreement from the best individual classifier outcome using the rank ordered spectral features to the aggregated outcome (0.42 versus 0.50). In the second category of experiments, the feature space was constrained to rank ordered features while several different fuzzy integration variants were used as outcome aggregators. In this case, there was a 33% improvement in the chance-corrected measure of agreement from the best fuzzy integration variant to the overall best individual classification outcome using any feature space variant (0.42 versus 0.56). Furthermore, these results were achieved using only 9% of the original spectral features.

Acknowledgement

The authors thank Alexander Demko for the design, implementation, and validation of the relevant Scopira framework components and Conrad Wiebe for the implementation and testing of the stochastic feature selection algorithm.

We also gratefully acknowledge the thoughtful comments provided by the two reviewers that contributed significantly to the quality of the manuscript.

References

[1] H. Friebolin, Basic One-and Two-Dimensional NMR Spectroscopy, third ed., John Wiley & Sons, New York, 1998.

[2] D.L. Pavia, G.M. Lampman, G.S. Kriz, Introduction to Spectroscopy, second ed., Harcourt Brace College Publishers, Fort Worth, 1996.

[3] R.L. Somorjai, B. Dolenko, A.K. Nikulin, N. Pizzi, G. Scarth, P. Zhilkin, W. Halliday, D. Fewer, N. Hill, I. Ross, M. West, I.C.P. Smith, S.M. Donnelly, A.C. Kuesel, K.M. Briere, Classification of ^{1}H MR spectra of human brain neoplasms: the influence of preprocessing and computerized consensus diagnosis on classification accuracy, Journal of Magnetic Resonance Imaging 6 (1996) 437–444.

[4] R.L. Somorjai, M. Alexander, R. Baumgartner, S. Booth, C. Bowman, A. Demko, B. Dolenko, M. Mandelzweig, A.E. Nikulin, N. Pizzi, E. Pranckeviciene, R. Summers, P. Zhilkin, A data-driven, flexible machine learning strategy for the classification of biomedical data, in: W. Dubitzky, F. Azuaje (Eds.), Artificial Intelligence Methods and Tools for Systems Biology, Springer, Dordrecht, 2004, pp. 67–85.

[5] N.J. Pizzi, Classification of biomedical spectra using stochastic feature selection, Neural Network World 15 (2005) 257–268.

[6] P. Bonissone, J.M. Cadenas, M.C. Garrido, R.A. Diaz-Valladares, A fuzzy random forest, International Journal of Approximate Reasoning, in press.

[7] M. Grabisch, The application of fuzzy integrals in multicriteria decision making, European Journal of Operational Research 89 (1996) 445–456.

[8] E.K. Tang, P.N. Suganthan, X. Yao, Gene selection algorithms for microarray data based on least square support vector machine, BMC Bioinformatics 7 (95) (2006), doi:10.1186/1471-2105-7-95.

[9] Q. Liu, A. Sung, Z. Chen, J. Xu, Feature mining and pattern classification for LSB matching steganography in grayscale images, Pattern Recognition 41 (1) (2008) 56–66.

[10] N. Kasabov, Q. Song, DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 144–154.

[11] B.S. Everitt, Moments of the statistics kappa and weighted kappa, British Journal of Mathematical and Statistical Psychology 21 (1968) 97–103.

[12] A.B. Demko, N.J. Pizzi, Scopira: an open source C++ framework for biomedical data analysis applications, Software—Practice and Experience 39 (2009) 641–660.

[13] http://scopira.org.

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

[15] M. Snir, W. Gropp, MPI: The Complete Reference, MIT Press, Cambridge, 1998.

[16] M. Grabish, T. Murofushi, M. Sugeno, Fuzzy measure of fuzzy events defined by fuzzy integrals, Fuzzy Sets and Systems 50 (1992) 293–313.

[17] G.J. Klir, T.A. Folger, Fuzzy Sets, Uncertainty, and Information, Prentice Hall, Englewood Cliffs, 1988.

[18] V. Torra, Y. Narukawa, The interpretation of fuzzy integrals and their application to fuzzy systems, International Journal of Approximate Reasoning 41 (2006) 43–58.

[19] A. Sugeno, Fuzzy measures and fuzzy integrals: a survey, in: Fuzzy Automatic and Decision Processes, North-Holland, Amsterdam, 1977, pp. 90–102.

[20] G. Choquet, Theory of capacities, Annales de l'Institut Fourier 5 (1953) 131–295.

[21] T. Murofushi, M. Sugeno, An interpretation of fuzzy measure and the Choquet integral as an integral with respect to a fuzzy measure, Fuzzy Sets and Systems 29 (1989) 201–227.

[22] R. Mesiar, A. Mesiarová, Fuzzy integrals—what are they?, International Journal of Intelligent Systems 23 (2008) 199–212

[23] A. Sugeno, Theory of Fuzzy Integral and its Applications, Ph.D. Thesis, Tokyo Institute of Technology, 1972.

[24] H. Tahani, J.M. Keller, Information fusion in computer vision using the fuzzy integral, IEEE Transactions on Systems, Man, and Cybernetics 20 (1990) 733–741.

[25] Z. Chi, H. Yan, T. Pham, Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition, World Scientific, New Jersey, 1996.

[26] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Transactions on Systems, Man, and Cybernetics SMC-3 (1973) 28–44.

[27] G.A.F. Seber, Multivariate Observations, Wiley and Sons, Hoboken, 1984.

[28] N.J. Pizzi, R.L. Somorjai, W. Pedrycz, Biomedical data classification using randomized feature selection and parallelized multi-layer perceptrons, in: Proceedings of the Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference, July 4–9, Perugia, Italy, 2004, pp. 1161–1166.

[29] J. Moody, C.J. Darken, Fast learning networks of locally-tuned processing units, Neural Computation 1 (1989) 294–381.

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

[31] D. Specht, Probabilistic neural networks for classification, mapping or associative memory, in: Proceedings of the IEEE International Conference on Neural Networks, IEEE Press, New Jersey, 1988, pp. 525–532.

[32] D.F. Specht, Probabilistic neural networks, Neural Networks 3 (1990) 109–118.

[33] R.L. Streit, T.E. Luginbuhl, Maximum likelihood training of probabilistic neural networks, IEEE Transactions on Neural Networks 5 (1994) 764–783.

[34] E. Parzen, On estimation of a probability density function and mode, Annals of Mathematical Statistics 33 (1962) 1065–1076.

[35] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

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

[37] L. Wang (Ed.), Support Vector Machines: Theory and Applications, Springer-Verlag, Berlin, 2005.

[38] V. Vapnik, A. Lerner, Pattern recognition using generalized portrait method, Automation and Remote Control 24 (1963) 774–780.

[39] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes: The Art of Scientific Computing, third ed., Cambridge University Press, Cambridge, 2007.

[40] J.C. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schölkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods: Support Vector Learning, MIT Press, Cambridge, 1999.