Normalized Mutual Information Feature Selection

Pablo A. Estévez, *Senior Member, IEEE*, Michel Tesmer, Claudio A. Perez, *Senior Member, IEEE*, and Jacek M. Zurada, *Fellow, IEEE*

***Abstract*—A filter method of feature selection based on mutual information, called normalized mutual information feature selection (NMIFS), is presented. NMIFS is an enhancement over Battiti's MIFS, MIFS-U, and mRMR methods. The average normalized mutual information is proposed as a measure of redundancy among features. NMIFS outperformed MIFS, MIFS-U, and mRMR on several artificial and benchmark data sets without requiring a user-defined parameter. In addition, NMIFS is combined with a genetic algorithm to form a hybrid filter/wrapper method called GAMIFS. This includes an initialization procedure and a mutation operator based on NMIFS to speed up the convergence of the genetic algorithm. GAMIFS overcomes the limitations of incremental search algorithms that are unable to find dependencies between groups of features.**

***Index Terms*—Feature selection, genetic algorithms, multilayer perceptron (MLP) neural networks, normalized mutual information (MI).**

I. Introduction

In pattern recognition, each pattern is represented by a set of features or measurements, and viewed as a point in the n-dimensional feature space. The aim is to choose features that allow us to discriminate between patterns belonging to different classes. In practice, the optimal set of features is usually unknown, and it is common to have irrelevant or redundant features at the beginning of the pattern recognition process. In general, it is desirable to keep the number of features as small as possible to reduce the computational cost of training a classifier as well as its complexity. Other objectives of dimensionality reduction are improving the predictor performance, and facilitating data visualization and data understanding [1]. To deal with these problems, two main dimensionality reduction approaches are typically used: *feature extraction* and *feature selection* [2]. According to Jain *et al.* [2], feature extraction are methods that create new features based on transformations or combinations of the original feature set. The term feature selection refers to methods that select the best subset of the original feature set.

Feature selection algorithms can be classified into *filters* and *wrappers* [3]. Filter methods select subset of features as a preprocessing step, independently of the induction (learning) algorithm. Wrappers utilize the classifier (learning machine) performance to evaluate the goodness of feature subsets. Several different criteria have been used for evaluating the goodness of a feature [4] including distance measures [5], [6], dependency measures [7], [8], consistency measures [9], [10], information measures [11], [12], and classification error measures [13]. Some authors have explored the combination of *filter* and *wrapper* algorithms, which allows the latter to exploit the knowledge delivered by the *filter* algorithm in order to speed up the convergence of the *wrapper* algorithm [6], [14].

Yu and Liu [8] decompose the set of features into irrelevant features, redundant features, weakly relevant but nonredundant features and strongly relevant features. According to the authors, an optimal feature subset should include all strongly relevant features and a subset of the weakly relevant features (those nonredundant). They use the concept of Markov blanket [15] to define feature redundancy. A new framework of feature selection is proposed that decouples relevance analysis and redundancy analysis. In [10], methods for selecting relevant but nonredundant attributes are proposed. The authors claim that employing different sets of relevant but nonredundant features improves classification accuracy.

In this paper, we focus on feature selection methods based on mutual information (MI) as a measure of relevance and redundancy among features. Battiti [11] defined the feature reduction problem as the process of selecting the most relevant k features from an initial set of n features, and proposed a greedy selection method to solve it. Ideally, the problem can be solved by maximizing $I(C;S)$, the joint MI between the class variable C and the subset of selected features S. However, computing Shannon's MI between high-dimensional vectors is impractical because the number of samples and the central processing unit (CPU) time required become prohibitive. To overcome these limitations, Battiti [11] adopted a heuristic criterion for approximating the ideal solution. Instead of calculating the joint MI between the selected feature set and the class variable, only $I(C;f_i)$ and $I(f_i;f_j)$ are computed, where f_i and f_j are individual features. Battiti's mutual information feature selector (MIFS) selects the feature that maximizes the information about the class, corrected by subtracting a quantity proportional to the average MI with the previously selected features.

Kwak and Choi [12] analyzed the limitations of MIFS and proposed a greedy selection method called MIFS-U, which in general, makes a better estimation of the MI between input attributes and output classes than MIFS. Another variant of Battiti's MIFS is the min-redundancy max-relevance (mRMR) criterion [16]. The authors showed that for first-order incremental search, i.e., when one feature is selected at a time, the mRMR criterion is equivalent to max-dependency, i.e., estimating $I(C;S)$. The MI for continuous variables was estimated using Parzen Gaussian windows [16]. To refine the results of the incremental search algorithm, i.e., minimize the classification error, mRMR is combined with two wrapper schemes. In the first stage, the mRMR method is used to find a candidate feature

Manuscript received February 16, 2007; revised January 04, 2008 and July 12, 2008; accepted July 17, 2008. First published January 13, 2009; current version published February 06, 2009. This work was supported by Conicyt-Chile under Grant Fondecyt 1050751. The work of Dr. J. M. Zurada was supported by the international cooperation project Fondecyt 7060211 from Conicyt-Chile.

P. A. Estévez and C. A. Perez are with the Department of Electrical Engineering, University of Chile, Casilla 412-3, Santiago 8370451, Chile (e-mail: pestevez@cec.uchile.cl).

M. Tesmer is with Cruz Blanca S.A., Santiago, Chile (e-mail: michel-tesmer@gmail.com).

J. M. Zurada is with the Department of Computer and Electrical Engineering, University of Louisville, Louisville, KY 40208 USA.

Digital Object Identifier 10.1109/TNN.2008.2005601

set. In the second stage, the backward and forward selections are used to search a compact feature subset from the candidate feature set that minimizes the classification error. However, MIFS-U and mRMR present similar limitations as MIFS in the presence of many irrelevant and redundant features, as will be discussed in Section III.

Chow and Huang [17] proposed the optimal feature selection MI (OFI-MI) algorithm. In this method, a pruned Parzen window estimator and the quadratic mutual information (QMI) are used to estimate MI efficiently. This allows estimating directly the high-dimensional QMI between the set of selected features and the output class. However, only 2-D QMI between features are computed. The OFI-MI method selects features one by one using a criteria for feature relevancy and another for feature similarity (redundant). OFI-MI outperformed MIFS and MIFS-U on four data sets.

In a different approach, Hild *et al.* [18] estimated the Renyi's quadratic entropy using Parzen windows and Gaussian kernels, instead of estimating Shannon's entropy, thus reducing the computational complexity. Then, the MI was approximated from Renyi's entropies to perform feature extraction using supervised training. In [19], entropic spanning graphs are used to estimate the MI between high-dimensional set of features and the classes. In this method, entropies are estimated directly from data samples, avoiding the estimation of *pdfs*. In this approach, the complexity does not depend on the number of dimensions but on the number of samples. However, a greedy forward feature selection algorithm is used, which adds features one at a time. In [20], a wrapper algorithm that uses an output information theoretic objective function for evaluating classifiers is proposed. The MI between the class labels and the discrete labels output by the classifier is used for the task of feature selection in multilayer perceptrons (MLPs) and support vector machines (SVMs).

The MIFS, MIFS-U, mRMR, and OFI-MI algorithms are all incremental search schemes that select one feature at a time. At each iteration, a certain criterion is maximized with respect to a single feature, not taking into account the interaction between groups of attributes. In many classification problems, groups of several features acting simultaneously are relevant but not the individual features alone. If any attribute of the group is absent, then the rest become irrelevant to the problem at hand. This phenomenon is known in evolution theory as epistasis [21]. Feature selection algorithms that evaluate the relevance of a single feature at a time will not select the optimal feature subset if the classification function depends on two or more features concurrently; see, for example, the extended parity problem with relevant, irrelevant, and redundant features [13], [20], or the continuous XOR problem [1].

Genetic algorithms (GAs) have been successfully applied to feature selection [22], [23]. The selection of groups of features can be done efficiently by using GAs, since they explore the solution space and exploit the most promising regions without doing an exhaustive search. In addition, niching genetic algorithms can solve multimodal problems, forming and maintaining multiple optima [24]. The computational load of simple GAs can be reduced by introducing a strategy for speeding up their convergence. For example, by introducing *a priori* knowledge in the iterative process and new genetic operators that accelerate the convergence toward the best solutions. In [25], hybrid GAs are proposed that include local search operators to improve the fine-tuning capabilities of simple GAs. The local search operators allows to add (remove) the most (least) significant feature to individuals in the GA population. The hybrid GAs outperformed both simple GAs and sequential search algorithms with several standard data sets. In [26], a hybrid GA is proposed that uses MI for feature ranking in the local search operations. The authors used a modified version of the MIFS-U criterion [12] to remove the insignificant features of every subset generated by GA in each generation. The GA for feature selection proposed in [13] introduced a guided mutation operator to accelerate convergence. Such mutation operator eliminates the irrelevant inputs to the neural classifier based on a pruning mechanism for neural networks.

In this paper, an enhancement over Battiti's MIFS, MIFS-U, and mRMR methods is proposed. The proposed feature selection method is called normalized mutual information feature selection (NMIFS). Due to its incremental nature, the proposed method is fast and efficient, but its performance degrades in problems where group of features are relevant but not the individual features composing the group. For this reason a second method, called genetic algorithm guided by mutual information for feature selection (GAMIFS), is proposed. It is a hybrid filter/wrapper method that combines a genetic algorithm with NMIFS. GAMIFS is able to find both individual relevant features and groups of features that are relevant.

Section II presents a background on MI, and the procedures used for estimating it. Section III describes the limitations of MIFS, MIFS-U, and mRMR criteria. Sections IV and V introduce the proposed feature selection methods NMIFS and GAMIFS, respectively. Section VI describes the artificial and benchmark data sets used in the simulations and presents the simulation results, as well as their discussion. Finally, the conclusions are drawn in Section VII.

II. Background on MI

Let X and Y be two continuous random variables with joint probability density function (pdf) $p(x,y)$, and marginal pdfs $p(x)$ and $p(y)$, respectively. The MI between X and Y is defined as [27]

$$I(X;Y) = \int\int p(x,y)\ \log \frac{p(x,y)}{p(x)p(y)}\ dxdy. \tag{1}$$

Consider two discrete random variables X and Y, with alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively. The MI between X and Y with a joint probability mass function $p(x,y)$ and marginal probabilities $p(x)$ and $p(y)$ is defined as follows:

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\ \log \frac{p(x,y)}{p(x)p(y)}. \tag{2}$$

The MI has two main properties that distinguish it from other dependency measures: first, the capacity of measuring any kind of relationship between variables; second, its invariance under space transformations [28]. The former property has its root in that the MI is built from joint and marginal pdfs of the variables and does not utilize statistics of any grade or order. The second property is based on the fact that the argument of the logarithm in (1) is nondimensional [29], thus the integral value does not depend on the coordinates chosen (transformation in the feature space). This property is preserved for transformations that are invertible and differentiable [28], such as translations, rotations,

and any transformation that preserve the order of the original elements of the variables.

Since the selection criterion is based on the values of the MI between attributes and classes, feature selection methods based on MI are extremely sensitive to the computation of this measure. The MI computation requires to estimate pdfs or entropies from the data samples. Even small estimation errors can reduce the efficacy of the selection method drastically. One approach is to utilize kernels [30] to approximate pdfs by combining basis functions. Kernel-based methods consist in superposing a basis function to each point of the feature, typically a Gaussian. The final pdf approximation is obtained by taking the envelope of all the basis functions superposed at each point. The quality of such estimation is generally high, but the computational load is high. Another approach is to use histograms [30] that partitions the space in equal segments and counts the number of elements in each partition. Selecting the partition size is the main source of error, because segments too large tend to underestimate the pdf and segments too small do not reflect the coarse detail of the distribution. Histogram-based methods are computationally very efficient but they could produce large estimation errors.

A different approach is to estimate entropies directly from data using nearest-neighbor distances [31]. In [32], an independent component analysis (ICA)-based method is proposed for estimating high-dimensional MI. The main idea is that after achieving mutually independent components, the high-dimensional MI can be obtained as the summation of marginal (1-D) MI estimates. The sample spacing approach [33] is used to estimate marginal entropy. In practice, the algorithm uses a greedy incremental search for ranking features.

Fraser [29] proposed a fast and efficient method to estimate the MI between two random variables using adaptive histograms. That algorithm is an intermediate level between kernel-based methods and those methods based on histograms, since the precision of Fraser's estimation method is better than plain histograms, but it is equally fast and efficient. Feature selection methods proposed in the literature based on MI are usually applied to problems with continuous features [11], [12], [17], [34]. However, real problems usually have both continuous and discrete features, and the method used to estimate the MI in each case should be different. In this paper, we use an extended version of Fraser's algorithm [35] for continuous random variables, while contingency tables are used for discrete variables.

III. Limitations of MIFS, MIFS-U, and MRMR Selection Criteria

Battiti [11] posed the feature selection problem as follows: Given an initial set F with n features, find subset $S \subset F$ with k features that maximizes the MI $I(C;S)$ between the class variable C, and the subset of selected features S. Battiti's MIFS is a heuristic incremental search solution to the above defined problem. The MIFS algorithm [11] is the following.

1) Initialization: Set $F \leftarrow$ "initial set of n features"; S $\leftarrow$ "empty set."
2) Computation of the MI with the output class: For each $f_i \in F$, compute $I(C;f_i)$.
3) Selection of the first feature: Find the feature f_i that maximizes $I(C;f_i)$; set $F \leftarrow F\backslash\{f_i\}$; set $S \leftarrow \{f_i\}$.
4) Greedy selection: Repeat until $|S| = k$.
   a) Computation of the MI between variables: For all pairs (f_i, f_s) with $f_i \in F$ and $f_s \in S$, compute $I(f_i;f_s)$, if it is not yet available.
   b) Selection of the next feature: Choose the feature $f_i \in F$ that maximizes

$$I(C;f_i) - \beta \sum_{f_s \in S} I(f_s;f_i) \tag{3}$$

   Set $F \leftarrow F\backslash\{f_i\}$; set $S \leftarrow \{f_i\}$.
5) Output the set S containing the selected features.

The parameter β is a user-defined parameter that regulates the relative importance of the redundancy between the candidate feature and the set of selected features.

The MIFS-U [12] algorithm only changes the selection criterion (3), which is rewritten as

$$I(C;f_i) - \beta \sum_{f_s \in S} \frac{I(C;f_s)}{H(f_s)} I(f_s;f_i) \tag{4}$$

where $H(f_s) = -\sum_{f_s \in S} P(f_s) \log P(f_s)$ is the entropy.

The mRMR criterion [16], for the first-order incremental search algorithm, optimizes the following condition:

$$I(C;f_i) - \frac{1}{|S|} \sum_{f_s \in S} I(f_s;f_i) \tag{5}$$

where $|S|$ is the cardinality of the set S. Notice that the mRMR criterion becomes the MIFS's criterion when β is adaptively chosen as $1/|S|$.

In all cases, the left-hand side term $I(C;f_i)$ measures the relevance of the feature to be added (information about the class) and the right-hand side term estimates the redundancy of the ith feature with respect to the subset of previously selected features.

One problem with MIFS and MIFS-U approaches is that the left-hand side and right-hand side terms in (3) and (4) are not comparable. Because the right-hand side term in (3) and (4) is a cumulative sum, it will grow in magnitude with respect to the first term, as the cardinality of the subset of selected features increases. When the left-hand side term becomes negligible with respect to the right-hand side term, the feature selection algorithm is forced to select nonredundant features with the already selected ones. This may cause the selection of irrelevant features earlier than relevant and/or redundant features. This problem is partly solved in the mRMR criterion (5) by dividing the sum with the cardinality of the set S, $|S|$. Another drawback is that MIFS and MIFS-U rely on the parameter β for controlling the redundancy penalization, but the optimal value of this parameter depends strongly on the problem at hand. In addition, the MIFS-U criterion is based on the assumption that conditioning by the class C does not change the ratio of the entropy of f_s and the MI between f_s and f_i, which is only valid for uniform probability distributions.

In order to analyze the main limitation of the three selection criteria described above, we need to revise some basic information theoretic concepts. The MI definition can be rewritten in terms of entropies and conditional entropies as follows [27]:

$$I(f_i;f_s) = H(f_i) - H(f_i|f_s) = H(f_s) - H(f_s|f_i) \tag{6}$$

where $H(f_i)$ and $H(f_s)$ are entropies and $H(f_i|f_s)$ and $H(f_s|f_i)$ are conditional entropies. From (6), the MI can take values in the following interval:

$$0 \leq I(f_i; f_s) \leq \min\{H(f_i), H(f_s)\}. \tag{7}$$

From (7), it follows that the MI between two random variables is bounded above by the minimum of their entropies. As the entropy of a feature could vary greatly, this measure should be normalized before applying it to a global set of features. The normalization compensates for the MI bias toward multivalued features, and restricts its values to the range $[0, 1]$. The MI bias is a well-known problem in the decision tree community, where a criterion is needed for selecting the best attribute to form the root of a tree [7]. Quinlan [36] showed that the MI of a feature A, measured with respect to another variable, is less than or equal to the MI of a feature A' created from A by randomly adding more values. For this reason, MI should be normalized with their corresponding entropy.

IV. Normalized MIFS

We define the normalized MI between f_i and f_s, $NI(f_i; f_s)$, as the MI normalized by the minimum entropy of both features

$$NI(f_i; f_s) = \frac{I(f_i; f_s)}{\min\{H(f_i), H(f_s)\}}. \tag{8}$$

In this paper, we propose to use the average normalized MI as a measure of redundancy between the ith feature and the subset of selected features $S = \{f_s\}$, for $s = 1, \ldots, |S|$, i.e.,

$$\frac{1}{|S|} \sum_{f_s \in S} NI(f_i; f_s) \tag{9}$$

where $|S|$ is the cardinality of set S. Equation (9) is a kind of correlation measure, that is symmetric and takes values in [0,1].[1] A value 0 indicates that feature f_i and the subset S of selected features are independent. A value 1 indicates that feature f_i is highly correlated with all features in the subset S.

The selection criterion used in NMIFS consists in selecting the feature that maximizes the measure G

$$G \doteq I(C; f_i) - \frac{1}{|S|} \sum_{f_s \in S} NI(f_i; f_s). \tag{10}$$

The right-hand side of (10) is an adaptive redundancy penalization term, which corresponds to the average normalized MI between the candidate feature and the set of selected features. In (10), there is no need of a user-defined parameter, as the β parameter in (3) and (4).

The complete NMIFS algorithm is as follows.

1) Initialization: Set $F = \{f_i \ / \ i = 1, \ldots, N\}$, initial set of N features, and $S = \{\emptyset\}$, empty set.
2) Calculate the MI with respect to the classes: Calculate $I(f_i; C)$, for each $f_i \in F$.
3) Select the first feature: Find $\hat{f}_i = \max_{i=1,\ldots,N}\{I(f_i; C)\}$. Set $F \leftarrow F \backslash \{\hat{f}_i\}$; set $S \leftarrow \{\hat{f}_i\}$.
4) Greedy Selection: Repeat until $|S| = k$.
   a) Calculate the MI between features: Calculate $I(f_i; f_s)$ for all pairs (f_i, f_s), with $f_i \in F$ and $f_s \in S$, if it is not available.
   b) Select next feature: **Select feature $f_i \in F$ that maximizes measure (10)**. Set $F \leftarrow F \backslash \{f_i\}$; set $S \leftarrow \{f_i\}$.
5) Output the set S containing the selected features.

The computational cost of NMIFS is given by the concurrent sorting of each pair of features, which is required when estimating the MI by Fraser's algorithm. The sorting method used was *heapsort* [37], whose complexity is $O(N \log N)$, even in the worst case, where N is the number of data samples. In addition, NMIFS calculates the entropy of each feature in the same step when calculating the MI between that feature and the class variable $I(C; f_i)$. Therefore, NMIFS has the same computational cost as MIFS, i.e., $O(N \log N)$.

V. Genetic Algorithm Guided by Mutual Information for Feature Selection

A filter/wrapper hybrid feature selection method is proposed that has two parts: a genetic algorithm and a neural network classifier. A GA called deterministic crowding (DC) is used [24]. This is a niching algorithm that in contrast to simple GAs it can find and maintain multiple optima in multimodal problems. In DC, all individuals in the population are randomly paired and recombined, i.e., probability of crossover is one. The binomial crossover is used here because it has no positional bias. Mutation is optional in DC. The resulting offspring has a tournament with its nearest parent in terms of Hamming distance. The winners are copied to the new population for the next generation.

For the feature selection problem, subset of selected features are represented as bit strings of length L (total number of features in the problem at hand), where "1" in the ith position indicates that the ith feature is included in the subset, and "0" indicates that the ith feature is excluded [38]. In order to evaluate the fitness of an individual (chromosome), the corresponding binary string is fed into an MLP classifier. The size of the input layer is fixed to L but the inputs corresponding to nonselected features are set to 0. The fitness function includes a classifier accuracy term and a penalty term for a large number of features. The fitness of a chromosome c is expressed as [22]

$$J(c) = \text{accuracy}(c) - \lambda \left(\frac{n\text{features}(c)}{L} \right) \tag{11}$$

where $\text{accuracy}(c)$ is the error rate per unit of the classifier associated to c, $n\text{features}(c)$ is the number of selected features. The parameter λ controls the tradeoff between the two terms in (11). To compute $\text{accuracy}(c)$, a three-layered MLP classifier is trained to minimize the sum of square errors by using a second-order backpropagation quasi-Netwon (BPQ) learning algorithm [39]. The accuracy of the classifier is measured as the maximum rate per unit of correct classifications on a validation set. BPQ is faster than first-order algorithms and many second-order algorithms such as Broydon–Fletcher–Goldfarb–Shanno (BFGS) [39].

A method to initialize the initial GA population with good starting points that makes use of the feature *ranking* delivered by NMIFS is introduced. In addition, a new mutation operator guided by NMIFS is used to speed up the convergence of the

[1] Theoretically, the MI between two random continuous variables can become infinite. In practice, when estimated from a finite number of samples using nonzero width histogram bins or kernel functions, this will not happen, because of the smoothing bias of these estimators. Consequently, the normalization property of (9) is guaranteed if such smoothing density estimators are employed.

GA. This operator allows adding a relevant feature or eliminating an irrelevant or redundant feature from individuals in the GA population. The mutated individual is evaluated first to verify whether mutation improves the classifier accuracy. Only if the mutant's fitness obtained is better than that of the original individual the mutation is completed, otherwise the mutated feature is restored. This is the only mutation operator used here. The proposed mutation operator can be used with any classifier, not necessarily an MLP neural network, since the mechanism for accelerating the convergence does not depend on the nature of the classifier.

A. Initialization Procedure by Using NMIFS

Let P be the population size, and L the length of individuals. NMIFS *ranking* is used to initialize a fraction $\rho \in [0,1]$ of the population. The user defined parameter $\theta \in [0,1]$ allows choosing the best θL features of the NMIFS ranking. The rest of the population is initialized randomly.

```
initialize_nmifs()
{
  Find the subset S of the best θL features using NMIFS
  ranking;
  Initialize ρP individuals by using NMIFS {
    For all f_i ∈ S {
      set the i th bit to 1;
    }
    Else {
      set the i th bit randomly in {0,1};
    }
  Initialize (1 − ρ)P individuals randomly;
  }
```

For example, let $P = 100$ be the population size, and $L = 50$ the length of the individuals $\rho = 0.3$ and $\theta = 0.2$. The initialization procedure will select $\rho P = 30$ individuals to be initialized using NMIFS ranking. The bits corresponding to the top $\theta L = 10$ features in the ranking will be set to 1, and the rest will be set randomly.

B. Mutation Operator Using NMIFS

Since the goal is to exploit the best solutions, the mutation operator is applied to the top δ percent individuals in the population. For a given iteration, the procedure is to draw an individual and evaluate whether its *fitness* value is greater than $1-\delta$ times the maximum fitness obtained in the last generation. In such a case, a feature is added with probability p_a; otherwise, a feature is eliminated with probability $1 - p_a$. When adding a feature, it is considered that the $|S|$ bits with value "1" correspond to the subset of features that are present in the individual. The NMIFS selection criterion is used to select the best feature to be included among those features that are not present in the current individual. In the elimination mode, the least relevant feature present in the individual is eliminated with probability p_i or the most redundant feature is eliminated with probability $1 - p_i$. In the latter case, the feature to be eliminated is the one that produces the largest increase in MI with respect to the remaining $|S| - 1$ features. After mutation, if the *fitness* of the mutated individual is greater than the *fitness* of the original individual, the latter is replaced in the population. Otherwise, the original individual is kept in the population.

The following three local search operators are defined. These operators act over a chromosome c in the population. The bits of c in "1" define the subset of selected features S. The bits of c in "0" define the set of features that have not been selected, i.e., $F \setminus S$, where F is the set of all features.

add_nmifs: Add the most informative feature among the set of features that are not present in the current individual, c, i.e., compute

$$i^* = \text{argmax}_i \left\{ I(C; f_i) - \frac{1}{|S|} \sum_{f_s \in S} NI(f_s; f_i) \right\}$$

where $f_i \in F \setminus S$. Set the i^*th bit of c to 1, generating new individual $\bar{c}$.

remI_nmifs: Remove the most irrelevant feature among the set of features already present in the current individual, c, i.e., compute

$$\hat{i} = \text{argmin}_s \{ I(C; f_s) \}$$

where $f_s \in S$. Set the $\hat{i}$th bit of c to 0, generating new individual $\bar{c}$.

remR_nmifs: Remove the most redundant feature among the set of features already present in the current individual, c, i.e., compute

$$\tilde{i} = \text{argmax}_s \left\{ \sum_{j=1, j \neq s}^{|S|} NI(f_j; f_s) \right\}$$

where $f_i, f_s \in S$. Set the $\tilde{i}$th bit of c to 0, generating new individual $\bar{c}$.

The function $\text{rand}()$ generates a random number in $[0,1]$ with uniform probability. The mutation procedure acts on a given chromosome c of the population, and it is defined as follows.

```
mutation_nmifs()
{
  If p_a ≤ rand()
    c̄ = add_nmifs(c);
  elseif p_i ≤ rand()
    c̄ = remI_nmifs(c);
  else
    c̄ = remR_nmifs(c);
}
```

C. GAMIFS Algorithm

Let c be an individual of length L and *fitness* $J(c)$. Let S be the subset of features present in c (number of bits in "1"), and $|S|$ its cardinality. Let $J_{\max}$ be the maximum *fitness* obtained in the last generation and $1 - \delta$ a fraction of $J_{\max}$ above which an individual is mutated. Let $d(c_1, c_2)$ be the Hamming distance

between binary strings c_1 and c_2. The following algorithm corresponds to the DC [24] algorithm with the NMIFS-based initialization and mutation operator.

```
gamifs()
{
   initialize_nmifs;
   repeat {
      repeat n/2 times {
         select two parents p1 and p2 from P;
         (c1, c2) = crossover(p1, p2);
         If J(c1) > (1 − δ)Jmax {
            c̄1 = mutation_nmifs(c1);
            If J(c̄1) ≤ J(c1), then c̄1 = c1;
         }
         If J(c2) > (1 − δ)Jmax {
            c̄2 = mutation_nmifs(c2);
            If J(c̄2) ≤ J(c2), then c̄2 = c2;
         }
         replace {
            If [d(p1, c̄1) + d(p2, c̄2)] ≤ [d(p1, c̄2) + d(p2, c̄1)] {
               If J(c̄1) > J(p1), replace p1 by c̄1;
               If J(c̄2) > J(p2), replace p2 by c̄2;
            }
            else {
               If J(c̄2) > J(p1), replace p1 by c̄2;
               If J(c̄1) > J(p2), replace p2 by c̄1;
            }
         }
      } /* end one generation */
   } until (stopping condition)
}
```

VI. Experimental Results

The performance of the NMIFS algorithm was compared with the results of MIFS, MIFS-U, and mRMR on four data sets: uniform hypercube synthetic data set, Breiman's waveform data set [40], spambase data set [17], and sonar data set [41]. In addition, NMIFS was evaluated in a time-series problem given by Box and Jenkins's gas furnace data set [42].

In all cases, the MI was estimated using the extended Fraser algorithm described in [35] for continuous features, and contingency tables for discrete features. The control parameter β for MIFS and MIFS-U was varied in the range $[0, 1]$ with a step size of 0.1. The results obtained with the best β value are used for comparison with NMIFS.

For evaluating feature subsets, an MLP with a single hidden layer was trained using BPQ [39] during 200 epochs. All results presented here are the average of ten trials with random initializations. All data sets, except the sonar data set, were split into three disjoint sets: training (50%), validation (25%), and testing (25%). Due to the small sample size (204 patterns), the sonar data set was partitioned in 50 samples for training, 50 samples for validation, and 104 samples for testing. The maximum rate of correct classifications on the validation set was used as stopping criterion. For the data sets with information about classes, the optimal number of hidden units N_h was selected by running the BPQ algorithm with $N_h \in [1, 20]$, and the MLP architecture with best validation results was chosen.

The performance of the GAMIFS algorithm was compared with the results of NMIFS, deterministic crowding GA (DC-GA) without mutation and DC-GA with the mutation proposed in [13] on four data sets: nonlinear AND synthetic data set, Breiman's waveform data set, spambase data set, and sonar data set. The parameter λ in the fitness function (11) was set to 0.1. In this way, the accuracy term is ten times more important than the penalty term. The central point here is that when having two solutions with the same accuracy, the one with less number of features should be preferred. The fitness of an individual was evaluated three times, and the best solution found was taken. Ten simulations with the final feature subset produced for each GA method were carried out. The average rate of classifications measured on the test set was used to present and compare results.[2]

A. Test Problem: Uniform Hypercube

In this synthetic data set, the nature of each feature is known *a priori* (relevant, irrelevant, or redundant). The order of importance of the relevant features is also known. The task is to find first the relevant features sorted in ascending order (the first feature is the more relevant), second the redundant features, and last the irrelevant features.

This problem consists of two *clusters* of 500 points, each drawn form a uniform distribution on ten-dimensional hypercubes $[0,\ 1]^{10}$. The set of relevant features $(f_1, f_2, \ldots, f_{10})$ was generated in decreasing order of importance. A given pattern belongs to class C_1 if $f_i < \gamma^{i-1} * \alpha$ for $i = 1, \ldots, 10$, and to class C_2, otherwise. For $\alpha = 0.5$ and $\gamma = 0.8$, the first feature divides the unit interval $[0, 1]$ in 0.5, the second one in 0.4, the third in 0.32, and so on. Fig. 1 shows a 3-D version of the hypercube problem. It can be seen that feature f_1 is more discriminative than f_2, and in turn, f_2 is more discriminative than f_3.

The hypercube data set consists of 50 features: features 1–10 are relevant, features 11–20 are irrelevant, and features 21–50 are redundant. The latter are linear combinations of the relevant features plus 10% of additive noise drawn from a Gaussian distribution $\mathrm{N}(0, 1)$.

Fig. 2 shows the results obtained with MIFS [Fig. 2(a)], MIFS-U [Fig. 2(b)], mRMR [Fig. 2(c)], and NMIFS [Fig. 2(d)] on the hypercube problem. The results shown for MIFS and MIFS-U correspond to the best β value in the range $[0, 1]$. The x-axis in this figure represents the selection order (feature ranking position), while the y-axis represents the feature number (index). For example, in Fig. 2(a), feature number 15 is selected in the sixth position, and feature number 20 is selected in the tenth position. The bold and thick bars illustrate the relevant features, the bars with a dot depict the irrelevant features, and the thin bars represent the redundant features. MIFS, MIFS-U, and mRMR select the redundant features after the irrelevant features, and select irrelevant features before some relevant features. This effect is stronger in MIFS and

[2]The C source codes for NMIFS and GAMIFS are available at http://www.cec.uchile.cl/~pestevez and http://micheltesmer.googlepages.com/researchinterests.

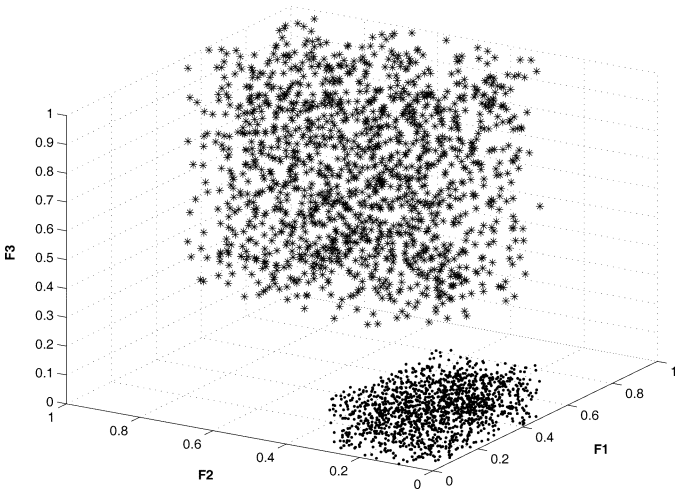


Fig. 1. Three-dimensional visualization of the hypercube synthetic problem.

mRMR [see Fig. 2(a) and (c)], where all the irrelevant features are selected before seven out of ten and eight out of ten relevant features, respectively. Fig. 2(b) shows that although MIFS-U has a better behavior than MIFS, two relevant features are selected after an irrelevant one. Fig. 2(d) shows that NMIFS selects all features in the ideal selection order: first the relevant set in the desired ascending order, second the redundant set, and last the irrelevant set.

B. Box and Jenkins's Gas Furnace

Box and Jenkins [42] described a gas furnace system where the input gas rate $u(t)$ could be varied and the resulting CO_2 concentration in the outlet gas $y(t)$ was measured. The goal is to predict the CO_2 concentration of the output gas using the past values of both variables.

Ten candidate features were considered for building a predictive model of the gas furnace time series

$$\{y(t-1), y(t-2), y(t-3), y(t-4), \\ u(t-1), u(t-2), u(t-3), u(t-4), u(t-5), u(t-6)\}.$$

An MLP classifier was trained using as inputs the features selected by the different methods. The optimal number of hidden units was determined as 3 by trial and error. The MLP network architecture was set as $N_{\text{in}} - 3 - 1$, where N_{in} is the number of selected features, 3 is the number of hidden units, and 1 is the number of output units.

This time series has been extensively used in the literature to measure and compare the performance of feature selection methods. Some fuzzy logic models select features by minimizing the prediction error [43], [44], [45], [46]. Training consists in finding a set of fuzzy rules that allows obtaining a good prediction of the desired output, using only a subset of the candidate features. Notice that these fuzzy methods belong to the *wrappers* category, because the attribute selection is made by training the model.

The NMIFS results were compared with those obtained by the following fuzzy models [43], [44], [45], [46], by measuring the MLP performance when fed with the features selected by the different methods. The normalized mean square error (NMSE)

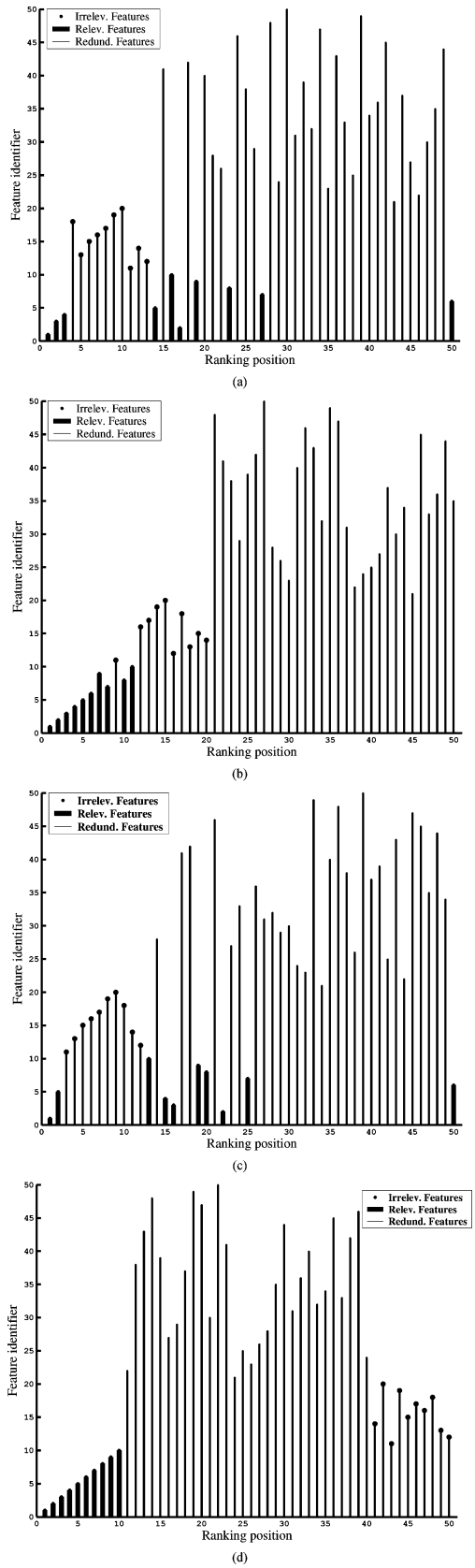


Fig. 2. Feature ranking in hypercube for (a) MIFS with $\beta = 0.4$, (b) MIFS-U with $\beta = 0.6$, (c) mRMR, and (d) NMIFS.

TABLE I
PREDICTION ERROR (NMSE) OBTAINED WITH SEVERAL FEATURE SELECTION METHODS FOR THE FURNACE GAS DATA SET

Selection Method	Number of Features	Selected Features	NMSE
NMIFS, Pedrycz [43]	2	$y(t-1), u(t-4)$	**0.068**
Tong [45], Xu [46]	2	$y(t-1), u(t-4)$	0.068
MI	2	$y(t-1), u(t-5)$	0.094
NMIFS	3	$y(t-1), u(t-4), u(t-5)$	**0.049**
MI	3	$y(t-1), u(t-5), u(t-4)$	0.049
Sugeno [44]	3	$y(t-1), u(t-4), u(t-3)$	0.062
NMIFS	4	$y(t-1), u(t-4), u(t-5), u(t-6)$	**0.042**
MI	4	$y(t-1), u(t-5), u(t-4), u(t-3)$	0.061

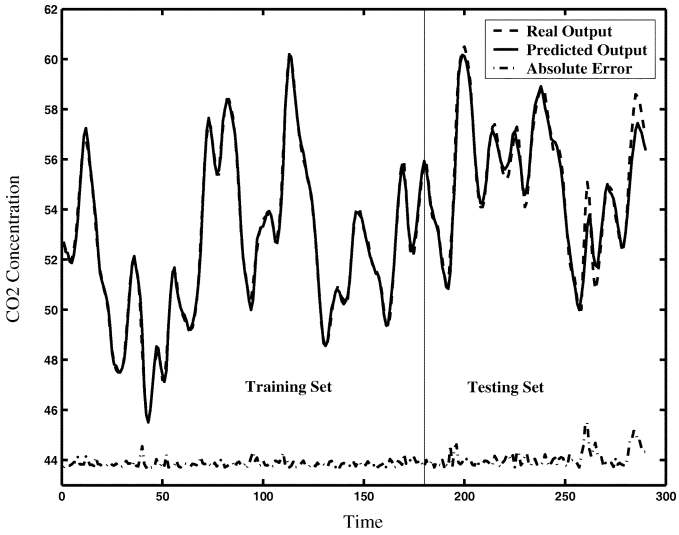


Fig. 3. Predicted output and absolute error using four features $(y(t-1), u(t-4), u(t-5), u(t-6))$ selected by NMIFS for the furnace gas data set.

was used as a performance measure to compare the results. It is defined as follows:

$$\text{NMSE} = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \tag{12}$$

where $\hat{y}_i$ is the predicted value (MLP output) and $\bar{y}_i$ is the mean value. The closer to 0 is the NMSE value, the higher is the quality of the prediction. A NMSE of 1 means that the model just predicts the mean value of the time series.

Table I shows the NMSE obtained with several selection methods for different number of features. It can be seen that NMIFS outperformed straight MI in all cases, except for the case of three features where the same performance is obtained. NMIFS achieved the same performance as those reported in the literature [43], [45], [46] when selecting two features and obtained a lower error than Sugeno *et al.* model [44] with three features. Even though NMIFS is a filter method, its performance is as good as the fuzzy models that belong to the *wrapper* category of selection methods.

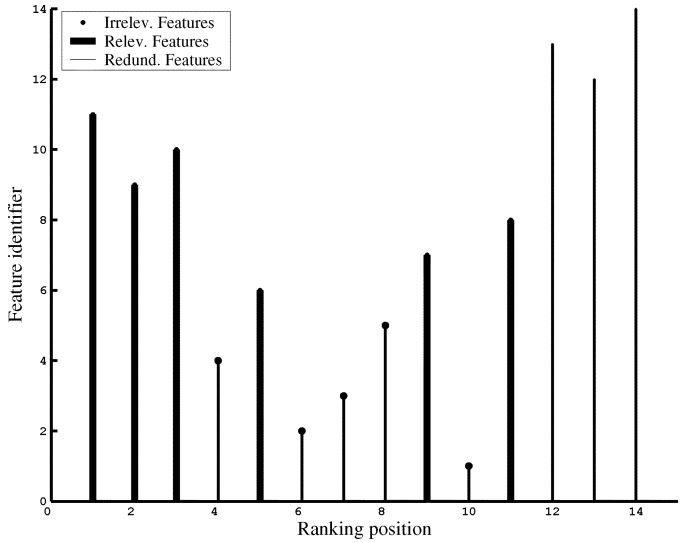


Fig. 4. Feature ranking for the nonlinear AND problem using NMIFS.

Fig. 3 shows the real and predicted outputs using four features selected by NMIFS. The small absolute error obtained is plotted at the bottom of that figure.

C. Test Problem: Nonlinear AND

The nonlinear AND is a synthetic problem devised to show a situation where NMIFS and any other incremental search algorithm will fail. The version studied here has 14 features: five irrelevant features f_1–f_5, six relevant features f_6–f_{11}, and three redundant features f_{12}–f_{14}. Each redundant feature is a duplicate of a relevant feature (features f_{12}–f_{14} duplicates f_9–f_{11} in this example). The irrelevant features were generated randomly from an exponential distribution with mean 10. The six relevant features were drawn from a uniform distribution on $[-1, 1]$. The relevant features determine to which of two classes belongs a sample x, according to the following nonlinear AND function.

---

nonlinear_AND()

{

If $((f_6 \cdot f_7 \cdot f_8) > 0)$ AND $((f_9 + f_{10} + f_{11}) > 0)$, then $x \in C_1$

If $((f_6 \cdot f_7 \cdot f_8) < 0)$ AND $((f_9 + f_{10} + f_{11}) < 0)$, then $x \in C_2$

}

NMIFS fails in this problem because features f_6, f_7, and f_8 do not provide information separately, but the presence of all of them solves the problem. In addition, the nonlinear AND problem has eight optima corresponding to the 2^3 combinations of the three duplicated features.

Fig. 4 shows the feature ranking obtained by NMIFS on the nonlinear AND problem. The relevant features that are linearly combined in the nonlinear AND function f_9–f_{11} are correctly selected in first place. But the relevant features that are multiplied in the nonlinear AND function f_6–f_8 are selected after some irrelevant features. In contrast, GAMIFS always found and maintained the eight optima, each one containing six relevant features. Fig. 5 shows the rate of convergence of the population to

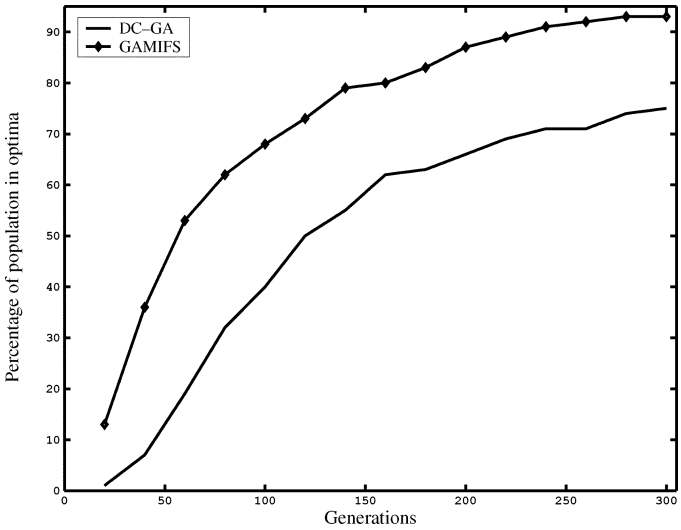


Fig. 5. Rate of population convergence to multiple optima versus the number of generations of GAMIFS for the nonlinear AND problem.

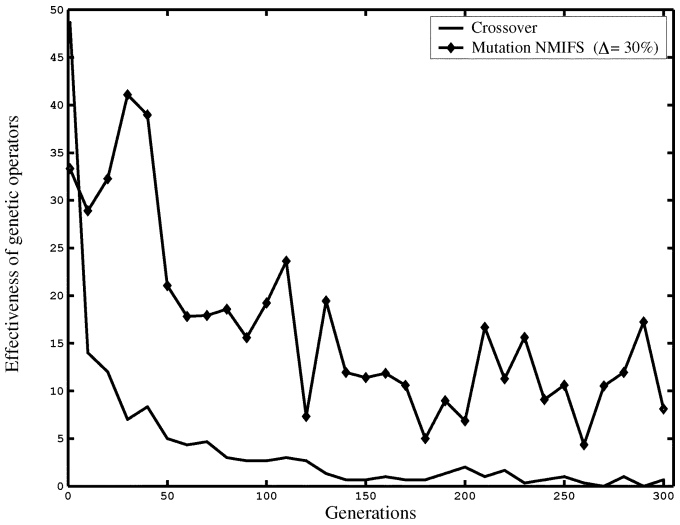


Fig. 6. Effectiveness of the crossover and mutation operators as a function of the number of generations for the nonlinear AND problem.

the eight optima for GAMIFS and DC-GA without mutation and with random initialization. The marginal gain of using GAMIFS is approximately 40%.

The effectiveness of the mutation and crossover operators is defined as the rate of successful applications of these operators. A mutation is said to be successful if the mutant's fitness is strictly greater than its parents' fitness. Likewise, a recombination (crossover) is said to be successful if the offspring's fitness is greater than their parents' fitness. Fig. 6 depicts the effectiveness of the crossover operator and the mutation_nmifs operator as a function of the number of generations. The effectiveness of the mutation operator is of 15% in average but it is applied only to the top 30% of the population. The effectiveness of the crossover operator decays to less than 5% after 50 generations, but it is applied to the entire population.

Because the solutions of the nonlinear AND data set are known, this problem was used to find a good set of parameters for GAMIFS, in terms of achieving the fastest rate of convergence. The population size was set to $P = 300$ to avoid premature convergence. The number of generations was set to $G = 300$, in order to allow for at least 75% of the population to converge toward optima.

Parameters of initialize_nmifs: The ρ and θ parameters were varied in $[0, 0.5]$. The criterion for chosen the best parameter combination was the greatest increase in the number of individuals located at any of the optima with respect to the GA with random initialization. The best results were obtained for $\rho = 0.15$ and $\theta = 0.3$. These parameters should be kept small in order to maintain diversity within the population.

Parameters of mutation_nmifs: The δ parameter was varied in $[0, 0.5]$, and the p_a and p_i parameters were varied in $[0, 1]$. The criterion for chosen the best parameter combination was the effectiveness of the mutation operator. The best results were obtained for $\delta = 0.3$, $p_a = 0.3$, and $p_i = 0.5$. This result means that the mutation operator favors eliminating features over adding features. MLPs with architecture 14–6–1 were trained for 100 epochs, where 14 is the number of inputs, 6 is the number of hidden units, and 1 is the number of outputs.

Taking into account the results for the nonlinear AND problem, the parameters of GAMIFS for the simulations with other databases were fixed as $P = 100$, $G = 200$, $\rho = 0.15$, $\theta = 0.3$, $\delta = 0.3$, $p_a = 0.3$, and $p_i = 0.5$.

D. Breiman Data Set

Breiman *et al.* [40] introduced a waveform recognition problem, where three waveforms are sampled at 21 points. Then, three classes are created C_1, C_2, C_3 by random convex combination of two of these sampled waveforms $(1, 2), (1, 3), (2, 3)$, respectively. In the noisy version of this problem, every pattern is augmented in 19 components drawn from a normal distribution $N(0, 1)$. The Breiman database contains 1000 samples (33% per class). The ideal selection order is to select first the relevant features 1–21 and then the irrelevant features 22–40.

Fig. 7 show the results obtained with MIFS, MIFS-U, mRMR, and NMIFS using the Breiman database. Fig. 7(a) and (c) shows that MIFS and mRMR produced low performance by selecting only five and two relevant features first, respectively. Both approaches leave out at the end more than half of the relevant features. Fig. 7(b) and (d) shows that both MIFS-U and NMIFS select 18 out of the 21 relevant features in the correct order. Notice that for MIFS-U the best β value ($\beta = 0.6$) was searched for in the range $[0, 1]$. This procedure is possible in the Breiman's data set because it is an artificial problem, and the solutions are known.

The subsets of features selected by NMIFS, GAMIFS, DC-GA without mutation, and DC-GA with the mutation proposed in [13] were fed into an MLP in order to obtain classification rates. The MLP architecture used was 40–15–3. Table II shows the rate of correct classifications using the subset of features selected for the different methods as inputs to an MLP classifier. For 13 features selected, GAMIFS outperformed NMIFS and DC-GA with mutation. This difference is statistically significant at the 0.01 significance level according to the t-student test, as shown in the p-value column of Table II. The classification results using the 13 features selected by GAMIFS are even better than using the entire set of 40 features.

E. Spambase Data Set

The Spam E-mail database [41] consists of 4601 patterns, with 57 features and two output classes (spam or no spam).

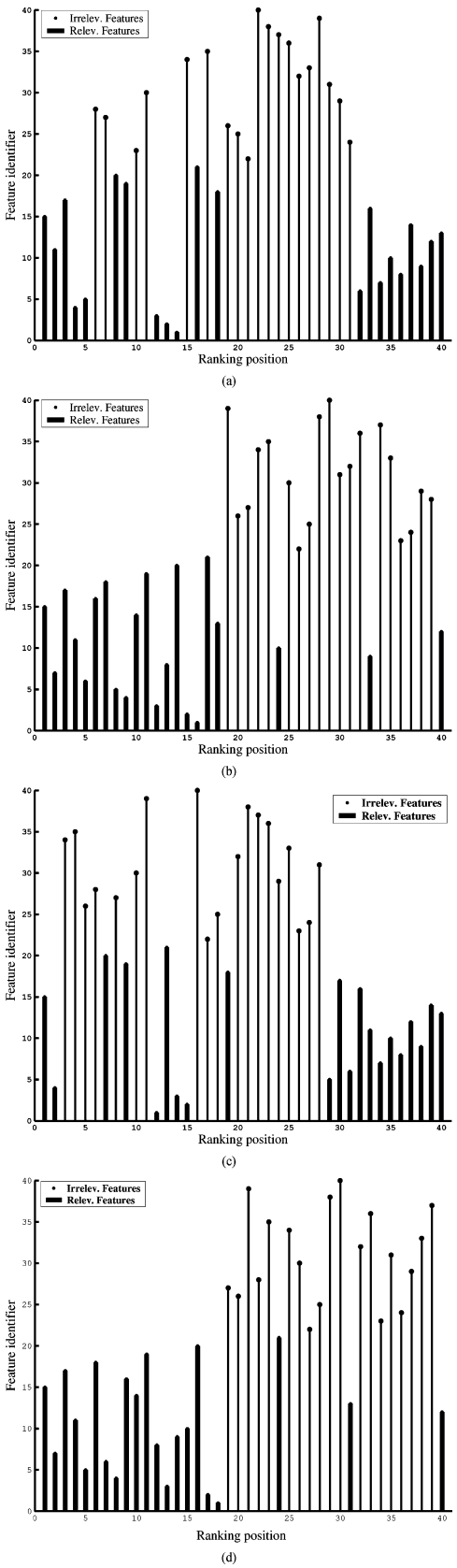


Fig. 7. Feature ranking for Breiman data set using (a) MIFS with $\boldsymbol{\beta}= \mathbf{0.3}$, (b) MIFS-U with $\boldsymbol{\beta}= \mathbf{0.4}$, (c) mRMR, and (d) NMIFS.

To evaluate the subsets of features selected for the different methods, an MLP classifier was trained on these subsets. The architecture used for the MLP was 57–12–1. Tests were carried out with 3, 6, 9, 12, 15, 18, 21, and 24 features selected by each algorithm.

TABLE II
MLP CLASSIFICATION RATES OBTAINED BY USING THE FEATURE SUBSETS SELECTED FOR SEVERAL METHODS ON THE BREIMAN DATA SET

Feature Selection Method	N° of Selected Features	% Correct Classifications	p-value (/GAMIFS)	Generation (/200)
GAMIFS	13	87.84	–	89
DC-GA with mut.	13	86.4	0.005	92
DC-GA without mut.	16	87.92	–	146
NMIFS	13	81.52	3.03e-11	–
Reference	All (40)	83.32		

TABLE III
MLP CLASSIFICATION RATES OBTAINED BY USING THE FEATURE SUBSETS SELECTED FOR SEVERAL METHODS ON THE SPAMBASE DATA SET

Feature Selection Method	N° of Selected Features	% Correct Classifications
GAMIFS	3	83.50
NMIFS	3	75.8
MIFS	3	78.4
MIFS-U	3	81.2
OFS-MI	3	78.4
Reference	All (57)	93.64

Fig. 8 shows that the generalization accuracy of an MLP classifier that uses as inputs the subsets of features selected by NMIFS, MIFS, MIFS-U, mRMR, and OFS-MI. The results of OFS-MI on the Spambase data set were reproduced from [17] for comparison purposes. For MIFS and MIFS-U, the best β parameter was selected, but no significative differences were found in the range $[0.3, 1.0]$. From Fig. 8, it can be seen that the best results were obtained with NMIFS for 12 or more features. The misclassification error using 24 features selected by NMIFS is near the 7% reported for this database in [41]. For less than ten features, OFS-MI shows better results than NMIFS. NMIFS outperformed mRMR for any number of features, as well as MIFS and MIFS-U except for three features.

Table III shows the rate of correct classifications using three features selected for the different methods as inputs to an MLP classifier. GAMIFS outperformed NMIFS, MIFS, MIFS-U, and OFS-MI.

F. Sonar Data Set

The Sonar data set [41] consists in 60 features drawn from 204 sonar returns from a metallic cylinder and a rock. The MLP architecture used was 60–5–2.

Table IV shows the rate of correct classifications obtained by an MLP using as inputs the features selected by NMIFS, mRMR, MIFS, and MIFS-U for the Sonar data set. NMIFS outperformed both MIFS and MIFS-U for all number of features and different values of the β parameter. It can be seen also that MIFS yielded better results than MIFS-U.

TABLE IV
PERCENTAGE OF CORRECT CLASSIFICATIONS IN TEST SET FOR SONAR DATA SET

N° Inputs	NMIFS	mRMR	MIFS ($\beta = 0.3$)	MIFS ($\beta = 0.5$)	MIFS ($\beta = 0.7$)	MIFS ($\beta = 0.9$)	MIFS-U ($\beta = 0.3$)	MIFS-U ($\beta = 0.5$)	MIFS-U ($\beta = 0.7$)	MIFS-U ($\beta = 0.9$)
4	80.19	78.46	79.23	77.69	78.17	77.69	73.85	73.85	75.58	76.25
7	85.19	80.09	79.81	83.65	83.46	83.65	74.81	74.81	76.54	76.92
11	86.36	79.80	80.58	84.62	83.85	84.62	77.31	77.31	77.31	76.35
15	86.73	81.06	80.96	85.96	85.19	85.77	79.81	84.04	84.04	82.98
All (60)					80.67					

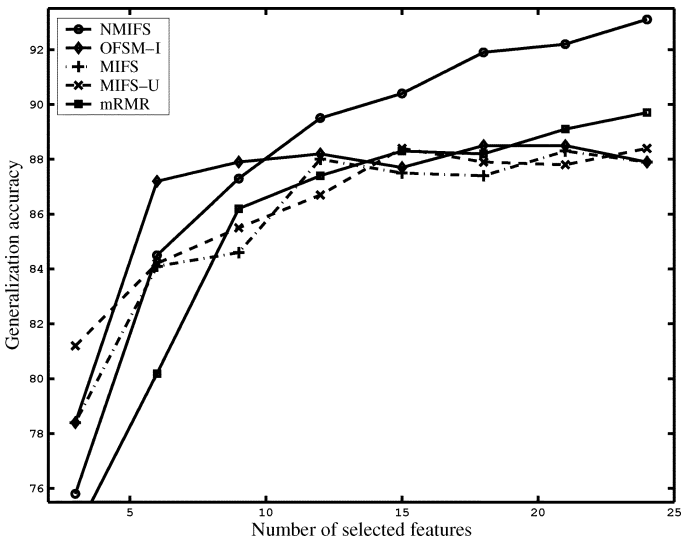


Fig. 8. Generalization classifier accuracy in Spambase data set.

TABLE V
COMPARISON OF GAMIFS WITH OTHER METHODS ON THE SONAR DATA SET

Feature Selection Method	N° of Selected Features	% Correct Classifications	p-value (/GAMIFS)	Generation (/200)
GAMIFS	11	90.96	–	186
DC-GA with mut.	14	90.38	–	104
DC-GA without mut.	15	87.06	–	135
NMIFS	11	86.36	0.01	–
Reference	All (60)	80.67		

Table V shows the rate of correct classifications using the features selected for GAMIFS and other methods as inputs to an MLP classifier. GAMIFS outperformed DC-GA with and without mutation finding the best solution with less number of features. The classification results using the 11 features selected by GAMIFS are even better than those using the entire set of 60 features.

G. Scalability of NMIFS and GAMIFS

Table VI shows the running time of NMIFS and GAMIFS on several data sets. For testing the scalability of the proposed methods, we included three data sets with a large number of features, taken from the 2003 Neural Information Processing Systems (NIPS) feature selection challenge: Madelon, Gisette, and Arcene.[3] The running times were measured in a Pentium IV, 1.8-GHz, 1-GB RAM. From Table VI, it can be seen that NMIFS can be applied effectively to data sets with more than 10 000 features. The running time of NMIFS could be reduced by using a faster method to estimate entropies such as [33].

TABLE VI
RUNNING TIME FOR NMIFS AND GAMIFS ON SEVERAL DATA SETS

Dataset	N° of Features	N° of Samples	NMIFS	GAMIFS
Sonar	60	208	1 s	3.6 hrs.
Breiman	40	500	4 s	29.1 hrs.
Spambase	57	2300	7 s	70.0 hrs.
Madelon	500	2600	16 min	–
Gisette	5000	7000	4.5 hrs.	–
Arcene	10000	200	6.0 hrs.	–

Due to the representation used in GAMIFS, the size of the search space is 2^L, where L is the number of features. For this reason, the number of features is restricted to less than 100 in GAMIFS. Notice that when using $G = 200$ and $P = 100$, the total number of combinations searched by GAMIFS is $G \times P = 20\,000$, which is a very small fraction of the size of the search space for $L > 20$. As a consequence, the appropriate search space for GAMIFS is between 20 and 100 features. An alternative approach to deal with larger sets of features is to apply NMIFS first to reduce the number of features to about 100, and then run GAMIFS over this reduced subset of features. Because the most expensive part of GAMIFS is to compute the fitness by training MLP neural networks, another option is to use a simpler and faster classifier.

VII. CONCLUSION

The proposed method for feature subset selection based on mutual information, NMIFS, is an enhancement over the MIFS, MIFS-U, and mRMR methods. We introduced the normalized MI as a measure of redundancy, in order to reduce the bias of MI toward multivalued attributes and restricts its value to the interval $[0, 1]$. NMIFS eliminates the need of a user-defined parameter such as β in MIFS and MIFS-U. This is helpful in practice because there is no clear guide on how to set this parameter for a real-world problem. NMIFS is a method of the filter type

[3]These data sets can be downloaded from http://www.nipsfsc.ecs.soton.ac.uk/datasets/

that selects feature subsets independently of any learning algorithm. The NMIFS method outperformed MIFS, MIFS-U, and mRMR on several artificial data sets and benchmark problems, except for the Breiman data set where NMIFS and MIFS-U yielded similar results, but the latter required to adjust β properly. When comparing NMIFS and mRMR results, it is clear that normalizing the MI has a great positive impact in the performance. In the gas furnace time-series problem, NMIFS obtained similar or better performance than fuzzy models of the wrapper type.

We have also proposed GAMIFS, a hybrid filter/wrapper method that combines the advantages of NMIFS with genetic algorithms. The accuracy of a trained MLP classifier was used to evaluate the goodness of feature subsets, but any classifier could be used in the wrapper part of the method. NMIFS is used also for finding good starting points for the GA search (initialization) and as part of a mutation operator. The proposed mutation operator allows adding or eliminating features to individuals, using the NMIFS selection criterion as inclusion procedure and the most redundant or most irrelevant feature as elimination criterion. GAMIFS overcomes the limitations of incremental search algorithms such as NMIFS, MIFS, MIFS-U, and mRMR that are unable to find dependencies between groups of features.

In future research, it would be of interest to use quadratic MI [47] for estimating MI between high-dimensional vectors, and to use the concept of Markov blanket for finding features that are weakly relevant but nonredundant. These ideas could be combined with the methods proposed here to get better and faster feature selection methods that may be successfully applied to large databases.

REFERENCES

[1] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[2] A. K. Jain, R. P. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 1, pp. 4–37, Jan. 2000.

[3] J. Kohavi and K. Pfleger, "Irrelevant features and the subset selection problem," in *Proc. 11th Int. Conf. Mach. Learn.*, 1994, pp. 121–129.

[4] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, no. 3, pp. 131–156, 1997.

[5] J. Bins and B. Draper, "Feature selection from huge feature sets," in *Proc. Int. Conf. Comput. Vis.*, Vancouver, BC, Canada, Jul. 2001, pp. 159–165.

[6] M. Sebban and R. Nock, "A hybrid filter/wrapper approach of feature selection using information theory," *Pattern Recognit.*, vol. 35, no. 4, pp. 835–846, Apr. 2002.

[7] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, Dept. Comput. Sci., Univ. Waikato, Waikato, New Zealand, 1999.

[8] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, pp. 1205–1224, Oct. 2004.

[9] M. Dash and H. Liu, "Consistency-based search in feature selection," *Artif. Intell. J.*, vol. 151, pp. 155–176, Dec. 2003.

[10] G. Lashkia and L. Anthony, "Relevant, irredundant feature selection and noisy example elimination," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 34, no. 2, pp. 888–897, Apr. 2004.

[11] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, Jul. 1994.

[12] N. Kwak and C.-H. Choi, "Input feature selection for classification problems," *IEEE Trans. Neural Netw.*, vol. 3, no. 1, pp. 143–159, Jan. 2002.

[13] P. A. Estévez and R. Caballero, "A niching genetic algorithm for selecting features for neural networks classifiers," in *Perspectives in Neural Computation (ICANN'98)*. New York: Springer-Verlag, 1998, pp. 311–316.

[14] G. van Dijck and M. M. van Hulle, "Speeding up the wrapper feature subset selection in regression by mutual information relevance and redundancy analysis," in *Lecture Notes on Computer Science*. Berlin, Germany: Springer-Verlag, 2006, vol. 4131, pp. 31–40.

[15] D. Koller and M. Sahami, "Toward optimal feature selection," in *Proc. 13th Int. Conf. Mach. Learn.*, 1996, pp. 284–292.

[16] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance and min-redundancey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[17] T. W. Chow and D. Huang, "Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 213–224, Jan. 2005.

[18] K. E. Hild, II, D. Erdogmus, K. Torkkola, and J. C. Principe, "Feature extraction using information theoretic learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1385–1392, Sep. 2006.

[19] B. Bonev, F. Escalano, and M. Cazorla, "Feature selection, mutual information, and the classification of high-dimensional patterns," *Pattern Anal. Appl.*, vol. 11, no. 3-4, pp. 309–319, 2008.

[20] V. Sindhwani, S. Rakshit, D. Deodhar, D. Erdogmus, J. Principe, and P. Niyogi, "Feature selection in MLPs and SVMs based on maximum output information," *IEEE Trans. Neural Netw.*, vol. 15, no. 4, pp. 937–948, Jul. 2004.

[21] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press, 1996.

[22] F. Brill, D. Brown, and W. Martin, "Fast genetic selection of features for neural networks classifiers," *IEEE Trans. Neural Netw.*, vol. 3, no. 2, pp. 324–328, Mar. 1992.

[23] M. Raymer, W. Punch, E. Goodman, L. Kuhn, and A. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 4, no. 2, pp. 164–171, Jul. 2000.

[24] S. W. Mahfoud, "Niching methods for genetic algorithms," Ph.D. dissertation, Dept. General Eng., Univ. Illinois at Urbana-Champaign, Urbana, IL, 1995.

[25] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1424–1437, Nov. 2004.

[26] J. Huang, N. Lv, and W. Li, "A novel feature selection approach by hybrid genetic algorithm," in *Lecture Notes on Artificial Intelligence*. Berlin, Germany: Springer-Verlag, 2006, vol. 4099, pp. 721–729.

[27] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[28] S. Kullback, *Information Theory and Statistics*. New York: Dover, 1997.

[29] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Phys. Rev. A, Gen. Phys.*, vol. 33, no. 2, pp. 1134–1140, Feb. 1986.

[30] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.

[31] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, p. 066138, Jun. 2004.

[32] T. Lan and D. Erdogmus, "Maximally informative feature and sensor selection in pattern recognition using local and global independent component analysis," *J. VLSI Signal Process. Syst.*, vol. 48, no. 1-2, pp. 39–52, Aug. 2007.

[33] O. Vasicek, "A test for normality based on sample entropy," *J. Roy. Statist. Soc. B*, vol. 31, pp. 632–636, 1976.

[34] N. Kwak and C.-H. Choi, "Input feature selection by mutual information based on Parzen window," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1667–1671, Dec. 2002.

[35] M. Tesmer and P. A. Estévez, "AMIFS: Adaptive feature selection by using mutual information," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Budapest, Hungary, Jul. 2004, pp. 303–308.

[36] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, pp. 81–106, 1986.

[37] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[38] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognit. Lett.*, vol. 10, no. 5, pp. 335–347, 1989.

[39] K. Saito and R. Nakano, "Partial BFGS update and efficient step-length calculation for three-layer neural networks," *Neural Comput.*, vol. 9, no. 1, pp. 123–141, 1997.

[40] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. London, U.K.: Chapman & Hall, 1984.

[41] D. Newman, S. Hettich, C. Blake, and C. Merz, UCI Repository of Machine Learning Databases, Univ. California at Irvine, Irvine, CA, 1998 [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[42] G. E. P. Box and G. M. Jenkins, *Time Series Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2003.
[43] W. Pedrycz, "An identification algorithm in fuzzy relational systems," *Fuzzy Sets Syst.*, vol. 13, pp. 153–167, 1984.
[44] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 7–31, Feb. 1993.
[45] R. Tong, "The evaluation of fuzzy models derived from experimental data," *Fuzzy Sets Syst.*, vol. 4, pp. 1–12, 1980.
[46] C. Xu and Z. Yong, "Fuzzy model identification and self-learning for dynamic systems," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-17, no. 4, pp. 683–689, Jul./Aug. 1987.
[47] J. C. Principe, D. Xu, and J. Fisher, "Information theoretic learning," in *Unsupervised Adaptive Filtering*, S. Haykin, Ed. New York: Wiley, 1999, ch. 7.

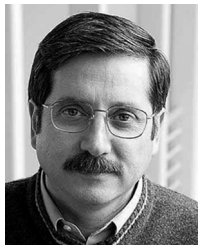
Pablo A. Estévez (M'98–SM'04) received the B.S. and P.E. degrees in electrical engineering from the University of Chile, Santiago, Chile, in 1978 and 1981, respectively, and the M. Sc. and Dr. Eng. degrees from the University of Tokyo, Tokyo, Japan, in 1992 and 1995, respectively.

He is currently Chairman of the Department of Electrical Engineering, University of Chile. He has been an invited researcher at the Communication Science Laboratory, NTT-Kyoto, ENS-Lyon in France, and visiting professor at the University of Tokyo. His research interests include neural networks, evolutionary computation and information theoretic learning applied to pattern recognition, data visualization, feature selection, clustering, classification, and prediction tasks.

Dr. Estévez currently serves as an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS and as a distinguished lecturer of the IEEE Computational Intelligence Society (CIS). He is an elected member of the IEEE CIS ADCOM committee for the period 2008–2010. He served as chairman of the IEEE CIS Distinguished Lecturer Program committee during 2005–2007.

Michel Tesmer received the B.S., P.E., and M.S. degrees in electrical engineering from the University of Chile, Santiago, Chile, in 2000, 2004, and 2004, respectively.

In 2005, he was a Research Intern at INRIA, Paris, France, as part of the scientific cooperation program between Chile and France. Currently, he is Head of the Market Intelligence's unit at the Chilean health care insurance company Isapre CruzBlanca S.A. His research interests include data mining, pattern recognition, neural networks, and evolutionary computation.



Claudio A. Perez (M'90–SM'04) received the B.S. and P.E. degrees in electrical engineering and the M.S. degree in biomedical engineering from the Universidad de Chile, Santiago, Chile, in 1980 and 1985, respectively, and the Ph.D. degree in biomedical engineering from The Ohio State University (OSU), Columbus, in 1991 (as a Fulbright student).

In 1990, he was a Presidential Fellow and received a Graduate Student Alumni Research Award from OSU. In 1991, he received a Fellowship for Chilean Scientists from Fundacion Andes. He is a Faculty member at the Department of Electrical Engineering, Universidad de Chile, where he was the Department Chairman between August 2003 and July 2006. He has been an invited speaker at The Ohio State University and part of a Fulbright Faculty Exchange with University of California Berkeley. His research interests include new models for pattern recognition and human–machine interfaces.

Dr. Perez is a member of the IEEE Systems, Man, and Cybernetics Society, IEEE Computational Intelligence Society, Pattern Recognition Society, Sigma-Xi, and the OSU Alumni Association.

Jacek M. Zurada (M'82–SM'83–F'96) received the M.S. and Ph.D. degrees (with distinction) in electrical engineering from the Technical University of Gdansk, Gdansk, Poland, in 1968 and 1975, respectively.

He serves as a Distinguished University Scholar and Professor of Electrical and Computer Engineering at the University of Louisville, Louisville, KY. He authored or coauthored several books and over 300 papers in the area of computational intelligence, neural networks learning, and logic rule extraction. He has also delivered numerous presentations and seminars throughout the world.

Dr. Zurada was the Editor-in-Chief of the IEEE TRANSACTIONS ON NEURAL NETWORKS. He also served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS and of the PROCEEDINGS OF THE IEEE. In 2004–2005, he was the President of the IEEE Computational Intelligence Society. He is an Associate Editor of *Neurocomputing* and of several other international journals. He holds the title of a National Professor of Poland, and was bestowed the Foreign Membership of the Polish Academy of Sciences. He is a Distinguished Speaker for IEEE Computational Intelligence Society.