Ant Colony Optimization Applied to Feature Selection in Fuzzy Classifiers

Susana M. Vieira[1], João M.C. Sousa[1], and Thomas A. Runkler[2]

[1] Center of Intelligent Systems, IDMEC
Instituto Superior Técnico, Technical University of Lisbon
Av. Rovisco Pais, 1049-001 Lisbon, Portugal
{susana,j.sousa}@dem.ist.utl.pt
[2] Siemens AG, Corporate Technology
Information and Communications CT IC 4
81730 Munich - Germany
Thomas.Runkler@siemens.com

Abstract. In practice, classifiers are often build based on data or heuristic information. The number of potential features is usually large. One of the most important tasks in classification systems is to identify the most relevant features, because less relevant features can be interpreted as noise that reduces the classification accuracy, even for fuzzy classifiers which are somehow robust to noise. This paper proposes an ant colony optimization (ACO) algorithm for the feature selection problem. The goal is to find the set of features that reveals the best classification accuracy for a fuzzy classifier. The performance of the method is compared to other features selection methods based on tree search methods.

1 Introduction

Real-world data analysis, data mining, classification and modeling problems usually involve a large number of candidate inputs or features. Sometimes the number of features is too large, making the problem computationally unfeasible or simply uncomprehensible. Feature selection has been an active research area in data mining, pattern recognition and statistics communities for many years [9]. The main idea of feature selection is to choose a subset of input variables by eliminating features that contribute with little or no information. The methods found in the literature can generally be divided into two main groups: model-free and model-based methods. Model–free methods use the available data only and are based on statistical tests, properties of functions, etc. These methods do not need to develop models to find significant inputs. The methods discussed in this paper belong to the group of model-based methods. Models with different sets of features are compared and the model that minimizes the model output error is selected. Often exhaustive methods are used where all subsets of variables must be tested. Decision tree search methods, with the proper branch conditions, limit the search space to the best performed branches, but do not guarantee to find the global best solution [10].

P. Melin et al. (Eds.): IFSA 2007, LNAI 4529, pp. 778–788, 2007.

Nature inspired algorithms like ant colony optimization have been successfully applied to a large number of difficult combinatorial problems like quadratic assignment, traveling salesman problems, routing in telecommunication networks, or scheduling, [6]. Ant colony optimization is particularly attractive for feature selection since no reliable heuristic is available for finding the optimal feature subset, so it is expectable that the ants discover good feature combinations as they proceed through the search space. Recently, nature inspired algorithms have been used to select features [1,7,12].

This paper proposes an ant based feature selection approach for fuzzy classifiers. The method is compared to other feature selection methods, namely two decision tree search approaches: top–down and bottom–up [10]. Our goal is to obtain simpler and more comprehensible fuzzy models for classification. The paper is organized as follows. Fuzzy classification is briefly described in Section 2. Section 3 presents the procedure of structure identification based on decision tree methods. The ant feature selection algorithm is described in section 4. A brief description of the application example, the experiments, and their respective results are presented and commented in Section 5. Finally, some conclusions are drawn in Section 6.

2 Fuzzy Classification

We use a fuzzy classifier, more precisely a fuzzy rule based classifier, as it provides a transparent model and a linguistic interpretation in the form of rules[13]. The fuzzy rule based models used in this paper are Takagi-Sugeno (TS) fuzzy models,which are presented in the next section.

2.1 Takagi-Sugeno Fuzzy Models

Takagi-Sugeno (TS) fuzzy models [15], consist of fuzzy rules where each rule describes a local input-output relation, typically in an affine form. Usually TS fuzzy models are represented by multi–input single–output (MISO) models. However, when multi–input multi–output (MIMO) models are necessary (like in the present work), they can be obtained as a collection of MISO models without lack of generality [13]. The affine form of a TS MISO model is given by:

$$R_i : \textbf{If } x_1 \text{ is } A_{i1} \textbf{and } \ldots \textbf{and } x_n \text{ is } A_{in} \textbf{then } y_i = a_{i1}x_1 + \ldots + a_{in}x_n + b_i \, , \quad (1)$$

where $i = 1, \ldots, K$, K denotes the number of rules in the rule base, R_i is the i^{th} rule, $\mathbf{x} = [x_1, \ldots, x_n]^T$ is the antecedent vector, n is the number of states, $A_{i1}, \ldots, A_{in}$ are fuzzy sets defined in the antecedent space, y_i is the output variable for rule i, $\mathbf{a}_i$ is a parameter vector and b_i is a scalar offset. The consequents of the affine TS model are hyperplanes in the product space of the inputs and the output. The model output, y, can then be computed by aggregating the individual rule contributions: $y = \sum_{i=1}^{K} \beta_i y_i / \sum_{i=1}^{K} \beta_i$, where β_i is the degree of activation of the ith rule, which is defined as: $\beta_i = \prod_{j=1}^{n} \mu_{A_{ij}}(x_j)$, and $\mu_{A_{ij}}(x_j) : \mathbb{R} \rightarrow [0, 1]$ is the membership function of the fuzzy set A_{ij} in the antecedent of R_i.

2.2 Identification

Firstly, the structure of the model must be identified. In this step, the significant features $\mathbf{x}$ of the model must be chosen. This is a very important step, especially for real-world problems. This task can be performed using the algorithms described in Section 3 and in Section 4. The number of variables must be small enough for the sake of simplicity, but with the sufficient number of variables to achieve the desired model accuracy. To identify the model, the regression matrix $\mathbf{X}$ and an output vector $\mathbf{y}$ are constructed from the available data: $\mathbf{X}^T = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, $\mathbf{y}^T = [y_1, \ldots, y_N]$. Here $N \gg n$ is the number of samples used for identification. The number of rules K, the antecedent fuzzy sets A_{ij}, and the consequent parameters $\mathbf{a}_i$ and b_i are determined by means of fuzzy clustering in the space of the input and output variables. Hence, the data set $\mathbf{Z}$ to be clustered is composed from $\mathbf{X}$ and $\mathbf{y}$:

$$\mathbf{Z} = [\mathbf{X}, \mathbf{y}]^T . \tag{2}$$

Given the data $\mathbf{Z}$ and the number of clusters K, several fuzzy clustering algorithms can be used. This paper uses the fuzzy c-means (FCM) [3] clustering algorithm to compute the fuzzy partition matrix $\mathbf{U}$. The fuzzy sets in the antecedent of the rules are obtained from the partition matrix $\mathbf{U}$, whose ikth element $\mu_{ik} \in [0, 1]$ is the membership degree of the data object $\mathbf{z}_k$ in cluster i. One-dimensional fuzzy sets A_{ij} are obtained from the multidimensional fuzzy sets defined point-wise in the ith row of the partition matrix by projections onto the space of the input variables x_j:

$$\mu_{A_{ij}}(x_{jk}) = \mathrm{proj}_j^{\mathbb{N}_{n+1}}(\mu_{ik}), \tag{3}$$

where proj is the point-wise projection operator [8]. The point-wise defined fuzzy sets A_{ij} are approximated by suitable parametric functions in order to compute $\mu_{A_{ij}}(x_j)$ for any value of x_j. The consequent parameters for each rule are obtained as a weighted ordinary least-square estimate. Let $\theta_i^T = \left[\mathbf{a}_i^T; b_i\right]$, let $\mathbf{X}_e$ denote the matrix $[\mathbf{X}; \mathbf{1}]$ and let $\mathbf{W}_i$ denote a diagonal matrix in having the degree of activation, $\beta_i(\mathbf{x}_k)$, as its kth diagonal element. Assuming that the columns of $\mathbf{X}_e$ are linearly independent and $\beta_i(\mathbf{x}_k) > 0$ for $1 \leq k \leq N$, the weighted least-squares solution of $\mathbf{y} = \mathbf{X}_e\theta + \varepsilon$ becomes

$$\theta_i = \left[\mathbf{X}_e^T \mathbf{W}_i \mathbf{X}_e\right]^{-1} \mathbf{X}_e^T \mathbf{W}_i \mathbf{y} . \tag{4}$$

The second step to identify a model consists of the estimation of the parameters of the model. The number of rules K, the antecedent fuzzy sets A_{ij}, and the consequent parameters $\mathbf{a}_i$ and b_i are determined in this step, by means of fuzzy clustering in the product space of the input and output variables [2].

The number of fuzzy rules (or clusters) that best suits the data must be determined for classification. For that purpose the following criterion, as proposed in [14], is used to determine the number of clusters:

$$S(c) = \sum_{k=1}^{N} \sum_{i=1}^{c} (\mu_{ik})^m (\parallel \mathbf{x}_k - v_i \parallel^2 - \parallel v_i - \bar{\mathbf{x}} \parallel^2), \tag{5}$$

where N is the number of data to be clustered, c is the number of clusters ($c \geq 2$), $\mathbf{x}_k$ is the k^{th} data point (usually vector), $\bar{\mathbf{x}}$ is the mean value for the inputs, v_i the center of the i^{th} cluster, μ_{ik} is the grade of the k^{th} data point belonging to i^{th} cluster and m is an adjustable weight. The parameter m has a great importance in this criterion. The bigger the m the bigger the optimum number of clusters. Therefore, this value is normally around 2. The number of clusters c is increased from two up to the number that gives the minimum value for $S(c)$. Note that this minimum can be local. However, this procedure diminishes the number of rules and consequently the complexity of the fuzzy model. At each iteration, the number of clusters are determined using the fuzzy c-means algorithm and the process stops when $S(c)$ increases from one iteration to the next one. The first term of the right-hand side of (5) is the variance of the data in a cluster and the second term is the variance of the clusters themselves. The optimal clustering achieved is the one that minimizes the variance in each cluster and maximizes the variance between clusters.

The performance criterion used to evaluate the fuzzy rule based classification model is based on misclassifications:

$$MSp = \frac{(n - mis)}{n} \times 100\%, \tag{6}$$

where n is the number of used samples and mis the number of misclassifications.

3 Decision Tree Methods

3.1 Bottom-Up Approach

The bottom-up approach described in this paper follows the principle of the regularity criterion (RC) approach [14], which is also a bottom-up approach. However, a more recent algorithm that minimizes the computational time with similar performance is used here [10]. The bottom-up approach starts with the most relevant feature(s) and successively adds the most relevant and removes the most irrelevant feature(s).

By using two groups of data, A and B, two fuzzy models are built, one for each group, starting with only one feature. At this stage, a fuzzy model is built for each of the n features in consideration. The models are evaluated using the RC performance criterion. The criterion is computed for each model at this stage, and the feature that minimizes the performance criterion is selected as the best one. The one(s) that maximizes the criterion is rejected and is not included in the next stage. At the next stage, the feature already selected is fixed, *i.e.*, it belongs to the model structure. The other feature candidates, excluding the rejected feature(s) in the prior stage, are added to the previous fuzzy model one at a time. When this second stage finishes, the fuzzy model has two features. The second feature is chosen as the one that minimizes the value of the chosen performance criterion, and as before, the feature(s) that maximizes the value of the criterion is rejected. This procedure is repeated until the value of the

Algorithm 1. Bottom–up approach

```
Cluster the data using fuzzy c-means with two initial clusters;
Increase the number of clusters until $S(c)$ in (5) reach its minimum;
Divide the data set into two groups $A$ and $B$;
For each input in the input vector that does not belong to the inputs of the model:
repeat
    Build two models, one using data group $A$ and other using data group $B$;
    Compute the PC;
    Select the input with the lowest value of PC as a new input of the model;
    Discard the input with the largest PC;
until PC increases or the end of the input vector is reached.
Select the final inputs;
Using the number of clusters given from (5) and the inputs selected using the
proposed approach, build a fuzzy model using a fuzzy clustering algorithm.
```

performance criterion increases. At this stage, one should have all the relevant features for the considered classification output. In a generic case, using the RC as proposed in [14], the maximum number of iterations is $n \times (n+1)/2$, where n is the number of possible features. The number of iterations using the bottom-up approach decreases. For an odd number of features the maximum number of iterations is $(n+1)^2/4$ and for an even number of features the maximum number of iterations is $n \times (n+2)/4$. Thus, the number of iterations reduces significantly, and then the computational time is also reduced. Assuming that input and output data are collected from a given system, the selection of inputs using this methodology generally entails the algorithm described in Algorithm 1.

Summarizing, the bottom–up approach presented in Algorithm 1 differs from the RC algorithm proposed in [14] because it is possible to exclude one or more variables. This is an advantage, as it allows the reduction of the number of iterations per stage. In some cases, it allows even the reduction of the number of stages, reducing also the computational time.

3.2 Top-Down Approach

Another approach proposed to select the input variables is the top-down (TD) approach. This approach begins with all the input variables, and removes the one(s) with the worst performance at each stage. This approach was proposed in [10]. The identification data is divided into two groups, A and B, as in the bottom–up approach.

Again, one model is built for each group A and B using all the variables. The proposed approach begins, at stage 0, by using all the variables. The performance criterion (PC) is computed. This is considered as the value to decrease at the following stages. Then, at stage 1, n fuzzy models are obtained, where each one of them is identified without one of the variables used at stage 0. The values of the chosen PC, for each of the n models, are compared to the value obtained

Algorithm 2. Top–down approach

```
Cluster the data using fuzzy c-means with two initial clusters;
Increase the number of clusters until $S(c)$ in (5) reach its minimum;
Divide the data set into two groups $A$ and $B$;
$i = 0$, where $i$ is the stage number;
repeat
   if Stage is zero then
      Build a model using all the input variables;
      $m = n$; where $n$ is the number of initial inputs
   else
      $i = i + 2$;
      Build a model using the input variables not discarded at the previous stage;
   end if
   Compute $PC_i$;
   for $j = 1$ to $m$ do
      Build two models, for groups $A$ and $B$, using all the inputs except input $j$;
      Compute $PC_{i+1,j}$
      if $PC_{i+1,j} < PC_i$ then
         Discard the input $j$ not used in modeling;
      end if
   end for
   $m = m - p$, where $p$ is the number of discarded inputs;
until (no input is discarded) OR (model has only one input) OR ($PC_i > PC_{i-2}$)
Using the number of clusters given from (5) and the inputs selected by the
proposed approach, build a fuzzy model using a fuzzy clustering algorithm.
```

at stage 0. For each new value that is smaller, the corresponding input x_i is removed from the vector of inputs. At the next stage, a fuzzy model is identified using only the inputs that have not been discarded at stage 1. The value of the chosen performance criterion is computed, and is used as reference for the next stage. The fuzzy model obtained at stage 2 has $n - p$ inputs, where n is the number of initial inputs and p is the number of inputs removed at stage 1. The presented procedure is repeated until the value of the performance criterion is not decreased by excluding any input. Thus, the inputs considered at stage 2 are the ones that are used in the final model. The top-down approach proposed in this paper is described in Algorithm 2.

This algorithm differs from the bottom–up approach, as it obtains at each stage multivariable fuzzy models, begins with the full feature vector, and discards one or more inputs at each stage. This is a clear advantage, which allows the reduction of the number of iterations per stage. Further, in some cases, it can even reduce the number of stages, and consequently the computational time can be reduced when compared to the BU approach. On the other hand, as the TD approach uses much more inputs to build each model from the beginning, and the identification of each model can be computationally intensive. This is especially critical when the number of inputs is large.

4 Ant Feature Selection

Ant algorithms were first proposed by Dorigo *et al.* [5] as a multi-agent approach to difficult combinatorial optimization problems like the traveling salesman problem and the quadratic assignment problem. There is currently a lot of ongoing activity in the scientific community to extend/apply ant-based algorithms to many different discrete optimization problems [4]. Recent applications cover problems like vehicle routing, sequential ordering, graph coloring, routing in communications networks, and so on. Ant algorithms were inspired by the observation of real ant colonies. Ants are social insects, that is, insects that live in colonies and whose behavior is directed more to the survival of the colony as a whole than to that of a single individual component of the colony.

The Ant Colony Optimization (ACO) methodology [6] is an optimization method suited to find minimum cost paths in optimization problems described by graphs. Consider a problem with n nodes and a colony of g ants. Initially, the g ants are randomly placed in g different nodes. The probability that an ant k in node i chooses node j as the next node to visit is given by

$$p_{ij}^{k}(t) = \begin{cases} \frac{{\tau_{ij}}^{\alpha} \cdot {\eta_{ij}}^{\beta}}{\sum\limits_{r \notin \Gamma}^{n} {\tau_{ir}}^{\alpha} \cdot {\eta_{ir}}^{\beta}}, \text{ if } j \notin \Gamma \\ 0, \qquad\qquad \text{otherwise} \end{cases} \tag{7}$$

where τ_{ij} and η_{ij} are the entries of the pheromone concentration matrix τ and heuristic function matrix η respectively, for the path (i, j). The pheromone matrix values are limited to $[\tau_{min}, \tau_{max}]$, with $\tau_{min} = 0$ and $\tau_{max} = 1$. Γ is the *tabu list*, which acts as the memory of the ants and contains all the trails that the ants have already passed and cannot be chosen again. The parameters α and β measure the relative importance of trail pheromone and heuristic knowledge, respectively.

After a complete tour, when all the g ants have visited all the n nodes, the pheromone concentration in the trails is updated by

$$\tau_{ij}(t + 1) = \tau_{ij}(t) \times (1 - \rho) + \Delta\tau_{ij}^{q} \tag{8}$$

where $\rho \in [0, 1]$ is the pheromone evaporation coefficient and $\Delta\tau_{ij}^{q}$ are pheromones deposited on the trails (i, j) followed by ant q that found the best solution $f^{q}(s)$ for this tour:

$$\Delta\tau_{ij}^{q} = \begin{cases} \frac{1}{f^{q}(s)} & \text{if arc } (i, j) \text{ is used by the ant } q \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The algorithm runs t_{max} times.

In this paper, an Ant Feature Selection (AFS) algorithm is proposed. Our goal is to assign n features to a subset of the total set of available features. The main objective is to have the best possible classification accuracy, i.e., to minimize the classification error:

$$E_{min} = |y_{est} - y| \tag{10}$$

Algorithm 3. Ant Feature Selection

```
/*Initialization*/
$n$ dimension of the subset of features
for every feature $i$ do
  $\tau_i(0) = \tau_0$
end for
for $k = 1$ to $m$ do
  Place ant $k$ on a randomly chosen feature
end for
Let $L^+$ be the best feature set found from beginning and $E^+$ its error;
/*Main Loop*/
for $t = 1$ to $t_{max}$ do
  for $k = 1$ to $m$ do
    Build feature set $L^k(t)$ by applying $n-1$ times the following step:
    Choose the next feature $j$ with probability
    $p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}$
    Where $i$ is the current feature
  end for
  for $k = 1$ to $m$ do
    Compute the fuzzy model using the feature set $L^k(t)$ produced by ant $k$
    Compute the error $E^k(t)$
  end for
  if an improved feature set is found then
    update $L^+$ and $E^+$
  end if
  for every feature $i$ do
    Update pheromone trails by applying the rule:
    $\tau_{ij}(t) \leftarrow (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$
  end for
end for
```

where y_{est} is the classification result. After completion of an iteration, each ant k lays a quantity of pheromone $\Delta\tau_{ij}^k(t)$ on each used feature. The value $\Delta\tau_{ij}^k(t)$ depends on how well the ant has performed. At iteration t, the deposited pheromone is given by:

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/E^k(t), & \text{if feature } i \in L^k(t) \\ 0, & \text{if feature } i \notin L^k(t) \end{cases} \tag{11}$$

where $L^k(t)$ is the set of features produced by ant k at iteration t, $E^k(t)$ is the error of the feature set, and Q is a parameter. The pheromone concentration in the features is updated by

$$\tau_{ij}(t+1) = \tau_{ij}(t) \times (1-\rho) + \Delta\tau_{ij}(t) \tag{12}$$

where $\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^k(t)$ and m is the number of ants. The transition rule, that is, the probability for ant k to use feature i while building its t^{th} feature set, is given by

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in J_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta} \tag{13}$$

where α and β control the relative weight of each feature between the pheromone concentration $\tau_{ij}(t)$, and the heuristic $\eta_{ij} = 1/E_{ij}$.

Table 1. Values of parameters used in the experiments

α	β	ρ	m	Q	τ	t_{max}	n
2	1	0.1	2	10	0.5	100	2, 4 or 11

5 Application

The proposed ant feature selection (AFS) algorithm is applied to a wine classification data set, which is obtained from the repository of University of California at Irvine [11]. The results are compared to decision tree methods for feature selection as described in Section 3. The classification data used in this paper contains the chemical analysis of 178 wines grown in the same region in Italy but derived from three different cultivars. Thirteen continuous attributes are available for classification: alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoids phenols, proanthocyanism, color intensity, hue, OD280/OD315 of dilluted wines and proline.

The features are selected using decision tree search methods or the AFS algorithm. The parameters used in the AFS algorithm are given in Table 1. The selected features are used to build fuzzy rule based models for classification.

First, both top-down and bottom-up approaches were applied to the database. The best number of features using these approaches are 11 and 4 features, respectively. Afterwards, AFS was applied using the same number of features. The results are shown in Table 2. This table shows also the results using AFS with only 2 features.

The features selected by the AFS algorithm are similar to the ones selected with both decision tree search approaches. Even so, the AFS has a smaller variability in results than the bottom-up approach, which means that the bottom-up approach is much more dependent on the performance criterion than AFS.

The main advantage of the ant based feature selection algorithm is the search in a much wider space of features subset. In the bottom-up approach, after choosing the best first feature, the following features subsets will always include this feature. This can be a disadvantage, because when this feature is combined with other features, it may turn out not to be the best feature, while in the AFS algorithm this is never the case. Further, AFS can achieve good classification rates even with a small number of features, see Table 2 when AFS is applied with only two features. This can be a very important characteristic for classification problems in very large data sets.

Table 2. Correct classification percentage of the wine classification using the bottom-up, top-down and ant feature selection approaches

	#11 features		#4 features		#2 features
Methods	Top–down	AFS	Bottom–up	AFS	AFS
Best	100	100	100	100	100
Average	99.9	99.8	96.7	99.3	97.7
Worst	99.4	97.7	92.7	97.7	89.8

6 Conclusions

This paper proposed an ant feature selection algorithm and compared it with tree search methods for feature selection. All three algorithms were used to select a subset of features that was then used as inputs of a Takagi–Sugeno fuzzy rule based classifier. We compared the performance of the three feature selection algorithms, when applied to the wine classification data set. The ant based feature selection algorithm yielded the best classification rate for low number of features. The top-down approach method was able to produced slightly better results only when a very high number of features was used.

In the near future we are planing to develop an enhanced algorithm to determine automatically the optimal number of features, and apply the proposed feature selection algorithm to classification problems in very large data sets.

Acknowledgements

This work is supported by the FCT grant SFRH/BD/25381/2005, Ministério do Ensino Superior da Ciência e Tecnologia, Portugal and by the Programa Operacional Ciência e Inovação (POCI) 2010, FCT, Portugal.

References

1. Ahmed Al-Ani. Feature subset selection using ant colony optimization. *International Journal of Computational Intelligence*, 2(1):53–58, 2005.
2. R. Babuška. *Fuzzy Modeling for Control.* Boston, 1998.
3. J. C. Bezdek. *Pattern Recognition With Fuzzy Objective Functions.* Plenum Press, New York, 1981.
4. D. Corne, M. Dorigo, and F. Glover. *New Methods in Optimisation.* McGraw-Hill, 1999.
5. M. Dorigo, V. Maniezzo, and A. Colorni. The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 26(1):1–13, 1996.
6. Marco Dorigo and Thomas Stützle. *Ant Colony Optimization.* The MIT Press, 2004.
7. Richard Jensen and Qiang Shen. Fuzzy-rough data reduction with ant colony optimization. *Fuzzy Sets and Systems*, 149:5–20, 2005.

8. G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic: theory and applications*. Upper Saddle River,Prentice Hall, 1995.
9. Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, April 2005.
10. L. F. Mendonça, S. M. Vieira, and J. M. C. Sousa. Decision tree search methods in fuzzy modeling and classification. *International Journal of Approximate Reasoning*, 44(2):106–123, 2007.
11. D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI Repository of machine learning databases, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.
12. Rahul Sivagaminathan and Sreeram Ramakrishnan. A hybrid approach for feature selection using neural networks and ant colony optimization. *Expert Systems with Applications*, 33:49–60, 2007.
13. J.M. Sousa and U. Kaymak. *Fuzzy Decision Making in Modeling and Control*. World Scientific and Imperial College, Singapore and UK, 2002.
14. M. Sugeno and T. Yasukawa. A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems*, 1(1):7–31, 1993.
15. T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man and Cybernetics*, 15(1):116–132, 1985.