A genetic integrated fuzzy classifier

Vito Di Gesù *, Giosuè Lo Bosco

Università di Palermo, Dipartimento di Matematica ed Applicazioni, Via Archirafi 34, 90123 Palermo, Italy

Available online 23 September 2004

Abstract

This paper introduces a new classifier, that is based on fuzzy-integration schemes controlled by a genetic optimisation procedure. Two different types of integration are proposed here, and are validated by experiments on real data sets of biological cells. The performance of our classifier is tested against a feed-forward neural network and a Support Vector Machine. Results show the good performance and robustness of the integrated classifier strategies.

Keywords: Clustering; Classification; Fusion; Fuzzy sets; Genetic algorithms; SVM; Neural network

1. Introduction

Clustering and classification problems can be often hard or vague, in spite of the simplicity of their formulation. For example, a good knowledge of the probability distribution of the feature space is not always available; the Gaussian assumption, that is often considered valid, hardly applies; linear separation between classes is usually a rough approximation.

Moreover, features and parameters, representing each element of the universe, \( U \), may have a non-numerical nature, so that, their interpretation becomes subjective and does not follow any statistical model. In such a situation, data-analysis system should include a suitable “fuzzy” representation of both data-model and expert-knowledge.

Fuzzy-clustering (Bezdek, 1987; Ruspini, 1969; Backer and Jain, 1981) has been introduced to tackle these situations. Fuzzy-clustering algorithms are, usually, objective functionals; i.e. the clustering criteria is based on the maximisation (minimisation) of a gain (loss) function, that is computed, for instance, from a possibility distribution (Yager, 1980) of the data that is provided by experts. The fuzzy \( c \)-mean (FCM) algorithm provides fairly good results when applied to the analysis of multi-spectra thematic maps (Cannon et al., 1986) and to the analysis of Magnetic Resonance Images (Di Gesù et al., 1991).

Moreover, the starting of multi-sensors experiments in the astronomy, high energy physics, and
remote sensing domains involves the fusion of heterogeneous data and the combination of several methods embedded in the so-called integrated data analysis systems (Gerianotis and Chau, 1990). The main motivation for considering combined technique is that even humans take decision models by using more than one evaluation paradigm, and usually complex decisions are taken by considering the evaluation of more than one expert.

In this paper, we introduce two Integrated Fuzzy Classification (IFC) schemes. They are grounded on the weighted combination of classifier $C_l$, with $1 < s \leq S$. The accuracy and the precision of each classifier can be represented by a weight, $\pi$, defined in the interval $[0,1]$. In (Di Gesù, 1994) it is shown the application of an integration method for the segmentation of medical MRI images. The explicit form of the IFC depends on the modality of the integration (sequential, parallel, and hybrid). In the following, we will consider the parallel and sequential integration schemes.

In the last decade, several methods for the integration of multiple classifiers have been developed (Dietterich, 2000; Valentin and Masulli, 2002). Recently, the fusion of classifiers has been performed by weighting the training samples, where a training sample with a high weight has a larger probability of being used in the training of the next classifier. To this purpose, a sequential integration modality is used. Adaptive Boosting (Freund and Schapire, 1995; Schapire et al., 1998) is an example of such integration algorithm. Other approaches assign different feature subsets to each single classifier (Kuncheva and Jain, 2000), integrate individual decision boundaries (e.g., mixture of experts (Jordan and Jacobs, 1994), ensemble averaging (Hashem, 1997)).

Moreover, the search for the best combination, corresponding to the near-optimal classes separation, can be always formulated as an optimization problem. This suggests the use of a genetic approach (Holland, 1975; Goldberg, 1989; Michalewicz, 1996) to face up to IFC. Genetic algorithms (GA) differ from more traditional optimization techniques in that they involve a search starting from a population of solutions, and not from a single point. Each iteration of a GA involves a competitive selection that weeds out poor solutions. Solutions with a higher fitness are recombined with other solutions by swapping parts of a solution with another. Solutions are also "mutated" by making a small change to a single element of it. This generation mechanism provides the ability to span the solution space escaping very often (but not always) from local maxima (minima). GA have been already applied successfully to clustering problems (Hall and Ozyurt, 1999; Lo Bosco, 2001).

An example of classifier integration has been proposed in (Mitra et al., 2001; Pal et al., 2003), where, a modular evolutionary strategy is presented to design a hybrid connectionist system that is based on the optimal combination of classification subtasks performed by a set of Multiple Layer Perceptron.

In the following, the IFC controlled by a genetic procedure are named Genetic-IFC. Two examples of Genetic-IFC algorithm are described in detail. They are supervised and both of them integrate three classifiers that run a nearest neighbor (NN) algorithm using three different distance functions. The first algorithm performs the classification by a direct assignment of each element to one of the classes, we name it all-against-all (AAA). The second one performs a tree classification of one class respect the remaining classes. The procedure is repeated for the total number of classes and we name it one-against-remaining (OAR).

Moreover, the genetic optimization yields the Genetic-IFC entirely free from the tuning of "ad hoc" parameters, and this is a very useful property for a classifier.

Both versions of Genetic-IFC have been validated and tested on three datasets. The first represents a breast cancer databases from the University of Wisconsin, the second is a waveform data-set. Both of them have been retrieved from the public domain (ftp://ftp.ics.uci.edu/pub/machine-learning-databases).

The third dataset represents five types of biological cells (classes): bacteria, white blood cell casts, non-squamous renal epithelial, and non-squamous transitional epithelial named A, B, C, and D respectively (see Fig. 1). From the figure it is evident that the classification is quite hard because...
non squamous renal epithelial and non squamous transitional epithelial classes have similar shapes, features and texture. This data-set has been kindly provided by IRIS Diagnostic, CA, USA.

The same data set has been classified using a feed-forward neural network (Bishop, 1996) and a Support Vector Machine (SVM) (Cristianini and Shawe-Taylor, 2000; Guyon et al., 1993) for comparison purposes. A naive Bayesian classifier has been also considered for comparison.

In Section 2 the integration problem is outlined. Section 3 describes the all-against-all and one-against-remaining classifiers and the genetic optimisation procedure. Section 4 describes an application of the proposed methods to the classification of biological cells. Experimental results are shown in Section 5. Section 6 is dedicated to final remarks and future developments.

2. Integrated fuzzy classification

A classification problem can be stated as follows: given a universe \( U \subset \mathbb{R}^n \) find the partition of \( K \) elements (clusters) \( \mathcal{S}(U) \equiv \{C_1, C_2, \ldots, C_k, \ldots, C_K\} \) such that \( \forall x \in U, \; x \in C_k \iff k = \arg\min_{1 \leq i \leq K} (\delta(x)) \); where \( \delta(k) \) is said discriminant function. In the following, \( C \equiv \{C_1, C_2, \ldots, C_S\} \) is the sets of \( S \) classifiers, \( M \equiv \{M^{(1)}, \ldots, M^{(S)}\} \) are \( K \times K \) confusion matrices derived from each classifier, and \( \Pi_M \equiv \{ \Pi_{M}^{(1)} / \pi_1, \; M^{(2)}/\pi_2, \ldots, M^{(S)}/\pi_S \} \) is the initial possibility distribution of \( M \), with \( \pi_s \in [0,1] \). The meaning of \( \Pi_M \) depends on the experimental situation and it could be evaluated using a priori subjective knowledge or from the information provided by a pre-classified data-set.

Formally, IFC consists of the evaluation of the final \( M \) and \( \pi \) by means of two functions \( G(M; \Pi_M) \), and \( m(\Pi_M) \). Their explicit forms depend on the particular integration paradigm used for combining the methods. Below we describe two main strategies: the parallel integration (IFC-parallel) and the sequential integration (IFC-sequential).

IFC-parallel. In this case all algorithms are executed independently. This strategy should be preferred whenever the classification algorithms do not exchange information. The evaluation of \( \pi \) is the crucial point in the procedure, and it is related to the relevance of each classification algorithm. The computation of \( G(M; \Pi_M) \) and \( m(\Pi_M) \) can be performed as follows:

\[
M = \frac{\sum_{s=1}^{S} \pi_s \times M^{(s)}}{\sum_{s=1}^{S} \pi_s} \quad \pi = \sum_{s=1}^{S} \pi_s \times \pi_s
\]

where \( \pi_s \geq 0 \) and \( \sum_{s=1}^{S} \pi_s = 1 \).

IFC-sequential. In this case \( M \equiv M^{(S)} \), and \( \pi \equiv \pi_S \). The computation at stage \( s > 1 \) depends on both \( (M^{(s)}, \pi_s) \) and \( (M^{(s-1)}, \pi_{s-1}) \). A sequential strategy is preferred whenever the classifiers interact in finding the best solution. One possible choice for the explicit form is:

\[
M^{(s)} = \frac{\pi_s \times M^{(s)} + \pi_{s-1} \times M^{(s-1)}}{\pi_s + \pi_{s-1}}
\]

\[
\pi'_s = \pi_{s-1} \times \pi_{s-1} + \pi_s \times \pi_s \quad \text{for } 1 < s \leq S
\]

where, \( \pi_{s-1} + \pi_s = 1 \). Note that in the IFC-sequential procedure, \( \pi_s \) is updated after the computation of \( M^{(s)} \). Moreover, in this case the result may depend on the order of execution of each classifier.

The optimisation problem in both cases can be formulated as follows:

Find \( x_1, x_2, \ldots, x_S \) such that \( \max \left( \sum_{k=1}^{K} M_{k,k} \right) \) under the constraint \( \sum_{s=1}^{S} \pi_s \times \pi_s \leq V \)

where \( 0 < V < +\infty \) is a finite constraint. The hybrid combination can be performed by alternating parallel and sequential modalities.
3. The Genetic-IFC

In the following two Genetic-IFC are described. The first one is based on IFC-parallel strategy, the latter is an example of IFC-sequential.

All-against-all (Fig. 2). Each classifier provides its own confusion matrix and possibility distribution, then the genetic optimisation (GO) procedure is applied as described later.

One-against-remaining (Fig. 3). Here, the all-against-all procedure is applied \( K - 1 \) times for two classes. In fact, at the \( k \)-th stage the \( k \)-th class is discriminated against the remaining ones considered as a superclass \( \mathcal{U} = \mathcal{U}_k \), where \( \mathcal{U}_k = \bigcup_{i=1}^{k-1} \mathcal{C}_i \), where \( \mathcal{C}_i \) is the set of elements in the \( i \)-th class. Note that a local genetic optimisation (GO) procedure is applied at the end of each stage. Therefore, the one-against-remaining algorithm is a binary tree classifier resulting from the application of all-against-all algorithms \( K - 1 \) times.

In the following we will denote the two algorithms all-against-all and one-against-remaining by AAA and OAR respectively.

3.1. The genetic optimisation

Here, the GO is described in detail; note that we start with the hypothesis of a complete ignorance about the relevance of each classifier, therefore initially \( \pi_s = 1/S \), \( 1 \leq s \leq S \). This allows us to reformulate the optimisation problem (3) in a simpler way:

\[
\text{Find } \pi_1, \pi_2, \ldots, \pi_S \text{ such that } \max \left( \sum_{c=1}^{L} M_{c,c} \right) \text{ under the constraint } \sum_{s=1}^{S} \pi_s = 1
\]  

(4)

In the following chromosomes are represented by floating point vectors of size \( K \times S \) (number of classes \( \times \) number of classifiers) that are exactly the values of \( \Pi \)’s.

Let us denote by \( M_\Pi \) the confusion matrix that is computed using \( \Pi \). A key point is the choice of the fitting function that is used to solve the global optimization problem. In our case it corresponds to the sum of the diagonal elements of \( M_\Pi \):

\[
f(\Pi) = \sum_{k=1}^{K} M_\Pi(k,k)
\]  

(5)

and the matrix \( M_\Pi \) is build using the two paradigms AAA and OAR above described.

The evolution of the population is determined by the classical single point crossover and a mutation operator that differs from classical ones, because it is non uniform with the iterations; it has been used in order to reduce the drawbacks of random mutation in the floating point representation of chromosomes (Janikowa and Michalewicz, 1991).

Consider the chromosome \( \Pi(g) = \{ \pi_1^{(1)}(g), \pi_2^{(2)}(g), \ldots, \pi_r^{(h)}(g), \ldots, \pi_S^{(S)}(g) \} \) at the generation \( g \), and suppose that \( \pi_r^{(h)}(g) \) has been selected for mutation, the new chromosome \( \Pi'(g) \) is then evaluated as follows:

\[
\pi_r^{(h)'}(g) = \begin{cases} 
\pi_r^{(h)}(g) + A(g,1-\pi_r^{(h)}(g)) & \text{if a random digit is 0} \\
\pi_r^{(h)}(g) - A(g,\pi_r^{(h)}(g)) & \text{if a random digit is 1}
\end{cases}
\]  

(6)

where \( A(g,y) = y \times (1-r) + (1-\frac{\pi}{y}) \), \( G \) is the maximum number of iterations, and \( r \) and is a random number in the interval \([0, 1]\).
In the case of the floating point representation the cross-over operator does not change the global content of each chromosome, while the mutation has the main responsibility for changing the population space.

The mutation rule used in this paper allows an adaptive mutation that generates a wider kind of chromosomes initially, and very selected ones at later stages. After the application of crossover and mutation, we select the chromosomes using the binary tournament method. The genetic operator and the selection process are applied until a maximum fixed number \( G \) of iteration is reached.

The genetic algorithm used to find the best weighting parameters may be sketched as follows:

**Program GA\_Search**

Set up a random population of chromosomes \( P(0) = \{\Pi_1(0), \Pi_2(0), \ldots, \Pi_n(0)\} \); \( t \leftarrow 0 \);

Initialize \( \Pi_b \) with the value of the best chromosome in \( P(0) \);

**while** \( t < G \)

for each \( \Pi_i(t) \in P(t) \) evaluate \( f(\Pi_i(t)) \);

find the best chromosome \( \Pi_b(t) \in P(t) \);

if \( f(\Pi_b(t)) > f(\Pi_b) \) then \( \Pi_b \leftarrow \Pi_b(t) \);

Apply crossover and mutation to the current population \( P(t) \), according to crossover probability (pc) and mutation probability (pm);

Apply binary tournament selection to the temporary population obtained from the previous instruction;

\( t \leftarrow t + 1 \);

**end**

4. An application to cell classification

In the following we describe an application of Genetic-IFC to the classification of biological cells.

4.1. The features extraction

Here, the data set \( U \) is composed of gray level images representing four different kind of cells. A vector \( X \in U \), of eight different features is extracted from each image:

\[
F(X) = (f_1(X), f_2(X), \ldots, f_8(X))
\]

The chosen features represent shape and gray levels properties of the cells.

The first feature \( f_1 \) is defined using a shape descriptor called the Iterated Object Transform (IOT) (Di Gesù et al., 2003), that stems from a combination of morphological operators and symmetry transformation. It allows to retrieve information on the internal structure of an object. More formally, the IOT, is given by:

\[
\text{IOT}_{\theta,n}(X) = S_\theta \circ \left(E^{n-1}(X)\right) \text{ for } n \geq 1
\]

where, \( \circ^n \) denotes the application of an operator \( \circ \), \( n \) times.

The \( E \) operator is the morphological erosion and has been implemented using the min operation (Vincent and Soille, 1991); the \( S_\theta \) operator has been implemented by computing the normalized axial moments of an object around its center of mass, with slope \( \theta \). The IOT computes the \( S \) transform on steadily intensity reduced versions of the input image, until a minimum of intensity is reached. At each iteration of the process we calculate the object elongation so defined:

\[
\eta_n(X) = \frac{\min_{\theta \in [0,\pi]} \{ \text{IOT}_{\theta,n}(X) \}}{\max_{\theta \in [0,\pi]} \{ \text{IOT}_{\theta,n}(X) \}}
\]

Assuming that the IOT has been applied \( N \) times, \( f_1 \) is defined as

\[
f_1(X) = \frac{1}{N} \sum_{n=1}^{N} \eta_n(X)
\]

i.e. the mean of the elongation over all the iterations.

The features from \( f_2 \) to \( f_5 \) are extracted applying directional histograms on the binary image \( X_B \) obtained from \( X \) using an adaptive threshold. Here, directional histograms on a binary image \( X_B \) of size \( \text{row} \times \text{col} \) are:

\[
V_j = \frac{\sum_{i=1}^{\text{row}} X_B(i,j)}{t_{\max}(j) - t_{\min}(j) + 1}
\]

\[
H_i = \frac{\sum_{j=1}^{\text{col}} X_B(i,j)}{j_{\max}(i) - j_{\min}(i) + 1}
\]
where \( i_{\text{max}}(j) \), \( i_{\text{min}}(j) \) are the maximum and minimum row coordinates of the object pixels along the \( j \)th column direction, and \( j_{\text{max}}(i) \), \( j_{\text{min}}(i) \) are the maximum and minimum column coordinates of the object pixels along the \( i \)th row direction.

\( f_3(X) \) and \( f_4(X) \) are the mean value and the standard deviation of \( V_j \) (\( j = 1, 2, \ldots, \text{col} \)), respectively. \( f_5(X) \) and \( f_6(X) \) are the mean value and the standard deviation of \( H_i \) (\( i = 1, 2, \ldots, \text{row} \)) respectively.

The features \( f_6(X) \) is the ratio between the area of the convex hull of \( X_B \), \( Ch(X_B) \), and the area of \( X_B \):

\[
f_6(X) = \frac{\text{area}(Ch(X_B))}{\text{area}(X_B)}
\]

\( f_7(X) \) is the ratio perimeter/area of \( X_B \), and \( f_8(X) \) is the area of the binary image obtained applying the Discrete Symmetry Transform (DST) to \( X \) as it is described in (Di Gesù, 2002).

### 4.2. Cells classification

In the following we describe the application on biological cells of the AAA and OAR classifiers. Three different classifiers \( C_{11}, C_{12}, C_{13} \) have been used to classify four type of cell images. Each classifier is characterized by a distance \( d \in \{d_1, d_2, d_3\} \) described in the following. Each \( d \) shows a different discrimination behavior. Moreover, each classifier uses the same data set \( U \) and the same feature parameters vector \( F = \{f_1, f_2, \ldots, f_8\} \).

In the following, the distance from the \( k \)th class is denoted by \( d_i \) with \( i = 1, 2, 3 \). The first distance \( d_1 \) evaluates for each component \( d_i \) the closeness of an unclassified cell to the center of mass, \( \mu_k \equiv (\mu_k^{(1)}, \mu_k^{(2)}, \ldots, \mu_k^{(8)}) \) of the \( k \)th class:

\[
d_i(X) = \frac{1}{\text{max}(F_i(X), \mu_k^{(i)})} \sum_{j=1}^{8} |F_i(X) - \mu_k^{(i)}| 
\]

\[
(13)
\]

where \( P_{\mu, \sigma} \) is the Normal Probability Density function with parameters \( \mu \) and \( \sigma \). This term has been added to consider the dispersion of the data. The other two distances are rank defined:

\[
d_i^k(X) = 1 - \frac{c_i(X, k)}{r} \quad \text{for} \quad i = 2, 3
\]

\[
(14)
\]

The term \( c_i(X, k) \) counts how many training images of the class \( k \) are in the set of the \( r \) nearest neighbors of \( X \), evaluated considering the distance \( \delta_i \) for \( i = 2, 3 \), where \( \delta_i \) are:

\[
\delta_2(a, b) = \frac{1}{n} \sum_{j=1}^{n} |a_j - b_j| \quad \frac{\max(a_j, b_j)}{\text{max}(a_j, b_j)}
\]

\[
\delta_3(a, b) = \frac{1}{n} \sum_{j=1}^{n} (a_j - b_j)^2 \quad \frac{\max(a_j^2, b_j^2)}{\text{max}(a_j^2, b_j^2)}
\]

\[
(15)
\]

where \( a = F_i(X) \) and \( b = F_i(Y) \) and \( Y \) is an element of the training set.

More formally if \( T_{mn}(X) \) is the a subset of training images which are the \( r \) nearest neighbors of \( X \) evaluated considering the distance \( \delta_i \) and \( T_{mn}^k(X) \subseteq T_{mn}(X) \) is the subset of those in class \( k \),

\[
c_i(X, k) = |T_{mn}^k(X)|
\]

\[
(16)
\]

Both AAA and OAR are supervised. In order to define the representative of each class a training set \( T \subseteq U \) is considered:

\[
T = \{t_1^{(1)}, t_1^{(2)}, \ldots, t_1^{(n_1)}, \ldots, t_1^{(n_l)}\}
\]

\[
(17)
\]

where \( t_j^{(h)} \) is an image from the \( j \)th class. The set \( U - T \) is then used as test set.

The search for the best integration coefficients is formulated as a global optimization problem as described in Section 3. The GO procedure has been implemented by using a linear combination of the three distances described above. The integrated distance, \( D_i \), is then:

\[
D_i(X) = \sum_{j=1}^{3} \pi_i^{(j)} \times d_i^k(X)
\]

\[
(18)
\]

where the vector of weighting coefficients \( II = \{\pi_i^{(j)}\} \) are used. The problem of fusion can be stated informally as it follows: find the vector \( II \) that maximize the diagonal of the global confusion matrix calculated considering the images in \( U - T \).

The decision rule to assign each image \( Y \in (U - T) \) (\( U - T \) is the test set) to the class \( j \) is (NN rule):

\[
\text{assign } Y \text{ to class } j \text{ if } D_i(X) \leq D_j(X)
\]
Assign $Y$ to $j = 1, 2, 3, 4 \iff j = \arg\min_{k = 1, 4} D_k(Y)$ (19)

Therefore, the integration strategy obtains the best weights $P_b$ such that:

$$P_b = \arg \max_{\Pi} \left( \sum_{i=1}^{4} M_{\Pi}(i, i) \right)$$ (20)

5. Experimental results

In the following we describe the experiments performed on three different data-sets:

- Breast cancer databases from the University of Wisconsin (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data);
- Waveform (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/waveform/waveform.data.Z; Breiman et al., 1984);
- Urine analysis cells.

The two classifiers, (AAA and OAR), have been compared with a two layer feed-forward neural net (FFNN) and a Support Vector Machine (SVM). The FFNN uses the Levenberg–Marquardt algorithm for the learning phase and the sigmoid transfer function for the neuron activation.

The SVM has been implemented using a kernel based on radial basis functions (Hastie et al., 2001):

$$\text{Kern}(F, F'; c) = e^{-\frac{|F-F'|^2}{c}}$$

where $F$ and $F'$ are the feature vectors and the parameter $c$ is tuned in order to give the best performance (in all experiments $c = 0.5$). Note that SVM is a binary classifier, therefore it is not used in the AAA paradigm, while it is applicable in the case of the OAR scheme.

Breast cancer. This data-set (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.data) is composed of 683 pre-classified individuals containing two classes 442 benign (class BE) and 241 malignant (class MA). The classification has been performed using 292 instances as training set, 391 instances as test set and nine features (clump Thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses) already present in the database. All these features are integer values ranging in the interval [1, 10].

In this case there are two classes, therefore AAA and OAR algorithms are equivalent.

AAA classification scheme. The optimal GO parameters have been found after $G = 3000$ iterations with a crossover probability of $pc = 0.8$ and mutation probability of $pm = 0.02$. The population size is fixed to 150 chromosomes for each iteration. In the case of the AAA-classification scheme the number of neurons was 10 for the hidden layer and 4 for the output layer. While, in the case of the OAR-classification scheme the number of neurons was 8 for the hidden layer and 2 for the output layer.

Rows 2 and 3 of Table 1 reports the percentage of recognition for the classifiers Bayesian, FFNN, Genetic-IFC, NN for the three distance functions used separately for each class; the fourth row provides the mean recognition rate.

Waveform. This data-set (ftp://ftp.ics.uci.edu/pub/machine-learning-databases/waveform/waveform.data.Z) is composed of 5000 pre-classified instances containing three classes of waves ($X_1, X_2, X_3$), 1657 waves of class 1, 1647 waves of class 2 and 1696 waves of class 3. The classification has been performed using 100 waves for each class as training set (total 300 training waves), 4700 waves as test set, and 21 attributes as features, each one of them containing noise. For further information about this experiment, see (Breiman et al., 1984).

OAR classification scheme. In this case the optimal GO parameters have been found after $G = 1000$ iteration for each level of the tree, with probability of crossover $pc = 0.8$ and probability of mutation $pm = 0.02$. The cardinality of the population is fixed to 150 chromosomes at each iteration.
Table 1
Cells classification results using OAR scheme

<table>
<thead>
<tr>
<th></th>
<th>Bayesian</th>
<th>FFNN</th>
<th>SVM</th>
<th>Genetic-IFC</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE</td>
<td>0.95</td>
<td>0.97</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>MA</td>
<td>0.98</td>
<td>0.84</td>
<td>0.94</td>
<td>0.98</td>
<td>0.93</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>Average</td>
<td>0.965</td>
<td>0.91</td>
<td>0.96</td>
<td>0.985</td>
<td>0.96</td>
<td>0.98</td>
<td>0.98</td>
</tr>
<tr>
<td>$X_1$</td>
<td>0.61</td>
<td>0.71</td>
<td>0.79</td>
<td>0.84</td>
<td>0.93</td>
<td>0.91</td>
<td>0.92</td>
</tr>
<tr>
<td>$X_2$</td>
<td>0.81</td>
<td>0.78</td>
<td>0.85</td>
<td>0.83</td>
<td>0.75</td>
<td>0.75</td>
<td>0.76</td>
</tr>
<tr>
<td>$X_3$</td>
<td>0.81</td>
<td>0.78</td>
<td>0.84</td>
<td>0.85</td>
<td>0.76</td>
<td>0.72</td>
<td>0.73</td>
</tr>
<tr>
<td>Average</td>
<td>0.74</td>
<td>0.75</td>
<td>0.82</td>
<td>0.84</td>
<td>0.81</td>
<td>0.79</td>
<td>0.80</td>
</tr>
<tr>
<td>$A$</td>
<td>0.94</td>
<td>0.94</td>
<td>0.92</td>
<td>0.77</td>
<td>0.76</td>
<td>0.77</td>
<td>0.74</td>
</tr>
<tr>
<td>$B$</td>
<td>0.79</td>
<td>0.62</td>
<td>0.62</td>
<td>0.75</td>
<td>0.54</td>
<td>0.75</td>
<td>0.69</td>
</tr>
<tr>
<td>$C$</td>
<td>0.84</td>
<td>0.55</td>
<td>0.70</td>
<td>0.83</td>
<td>0.69</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>$D$</td>
<td>0.83</td>
<td>0.35</td>
<td>0.67</td>
<td>0.85</td>
<td>0.85</td>
<td>0.57</td>
<td>0.59</td>
</tr>
<tr>
<td>Average</td>
<td>0.85</td>
<td>0.61</td>
<td>0.73</td>
<td>0.80</td>
<td>0.71</td>
<td>0.75</td>
<td>0.73</td>
</tr>
</tbody>
</table>

Table 2
Cells classification accuracy using AAA scheme

<table>
<thead>
<tr>
<th></th>
<th>Bayesian</th>
<th>FFNN</th>
<th>Genetic-IFC</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>0.78</td>
<td>0.76</td>
<td>0.73</td>
<td>0.53</td>
<td>0.69</td>
<td>0.68</td>
</tr>
<tr>
<td>$X_2$</td>
<td>0.81</td>
<td>0.84</td>
<td>0.92</td>
<td>0.94</td>
<td>0.87</td>
<td>0.89</td>
</tr>
<tr>
<td>$X_3$</td>
<td>0.82</td>
<td>0.78</td>
<td>0.88</td>
<td>0.95</td>
<td>0.88</td>
<td>0.89</td>
</tr>
<tr>
<td>Average</td>
<td>0.80</td>
<td>0.80</td>
<td>0.85</td>
<td>0.81</td>
<td>0.82</td>
<td>0.82</td>
</tr>
<tr>
<td>$A$</td>
<td>0.94</td>
<td>0.91</td>
<td>0.97</td>
<td>0.95</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>$B$</td>
<td>0.66</td>
<td>0.56</td>
<td>0.65</td>
<td>0.62</td>
<td>0.58</td>
<td>0.34</td>
</tr>
<tr>
<td>$C$</td>
<td>0.74</td>
<td>0.55</td>
<td>0.69</td>
<td>0.55</td>
<td>0.64</td>
<td>0.60</td>
</tr>
<tr>
<td>$D$</td>
<td>0.49</td>
<td>0.46</td>
<td>0.45</td>
<td>0.45</td>
<td>0.45</td>
<td>0.65</td>
</tr>
<tr>
<td>Average</td>
<td>0.71</td>
<td>0.62</td>
<td>0.69</td>
<td>0.64</td>
<td>0.65</td>
<td>0.63</td>
</tr>
</tbody>
</table>

Rows 5–7 of Table 1 reports the percent of recognition for the classifiers Bayesian, FFNN, SVM, Genetic–IFC, NN for the three distance function used separately for each class; row 8 provides the mean recognition rate.

AAA classification scheme. The optimal GO parameters have been found after $G = 3000$ iteration with probability of crossover $pc = 0.8$ and probability of mutation $pm = 0.02$. The cardinality of the population is fixed to 150 chromosomes for each iteration. In the case of the AAA-classification scheme the number of neurons was 10 for the hidden layer and 3 for the output layer. While, in the case of the OAR-classification scheme the number of neurons was 8 for the hidden layer and 2 for the output layer.

Rows 2–4 of Table 2 reports the percent of recognition for the classifiers Bayesian, FFNN, Genetic-IFC, NN for the three distance function used separately for each class; row 5 provides the mean recognition rate.

Urine analysis cells. The data set $U$ is composed of 640 pre-classified images representing four different kind of cells: bacteria (class $A$), white blood cell casts (class $B$), non squamous renal epithelial (class $C$), non squamous transitional epithelial (class $D$).

These cells are present in the human urine and the purpose is to include the classification in a semiautomatic system for the urine analysis. The classification problem is hard because classes non squamous renal epithelial, non squamous transi-

---

1 Data sets have been kindly provided by IRIS Diagnostic, CA, USA.
tional epithelial, and white blood cell casts may have similar shape and texture. Fig. 4 shows an example of such ambiguity.

The training set, $T$, is composed of 240 images (60 for each class). The number of elements of $T$ has been set equal for all classes because they are represented in $U$ with the same abundance and their structural complexity is assumed comparable. In the OAR-classification schema the training set for the two classes is built starting from $T$ by assigning 60 elements to the $k$th class and $240 - (k \times 60)$ for the superclass of the remaining.

**OAR classification scheme.** In this case the optimal GO parameters have been found after $G = 1000$ iteration for each level of the tree, with probability of crossover $p_c = 0.8$ and probability of mutation $p_m = 0.02$. The cardinality of the population is fixed to 150 chromosomes at each iteration.

Rows 9–12 of Table 1 reports the percent of recognition for the classifiers Bayesian, FFNN, Genetic-IFC, NN for the three distance function used separately for each class; row 13 provides the mean recognition rate. Note that the Bayesian, as expected, is optimal, the Genetic-IFC is more accurate than FFNN and NN using the three distance separately. The results show that the integration method provides a better results in the average. Only for the class $D$ the result obtained by Genetic-IFC and NN for $d_1$ and $d_2$ are comparable. Class $D$ is better discriminated by the classifier NN for $d_3$.

### 6. Conclusions

This paper introduces a new paradigm for classifiers integration that is based on a GA optimization approach, searching for the best integration parameters. The Genetic-IFC has been applied to the classification of biological cells by integrating three different classifiers. The genetic paradigm has been chosen because it allows us to search in a large solution space that can be defined subjectively. In our case the performance of three separated classifiers are combined to reach a better classification. Results show that the method proposed is competitive and its performance and accuracy can be compared successfully to the ones of Neural network and SVM classifiers. We stress that the parameters used in the Genetic-IFC can be tuned in a more flexible way. Moreover they are most robust allowing us to estimate them from a wider dominium unlike FFNN and SVM.

### References


