Representation of Genetic Individuals for Unmixing Multispectral Data

Arnaud Quirin, Jerzy Korczak
Université Louis Pasteur
LSIIT, CNRS
Strasbourg, France
{quirin,korczak}@lsiit.u-strasbg.fr

Abstract- The traditional classification algorithms for multispectral images assign only one class to each pixel. However, such pixels are actually a mixture of the spectral reflectance values of several different types of ground, of which the various abundances characterize the final shape of the observed spectrum. Within the framework of supervised learning, a representative solution was defined to solve this kind of problem using a genetic algorithm. This paper introduces a representation of the selected and various associated genetic operators (*fitness*, crossover, mutation) used in remote sensing image classification, as well, it describes a comparison of various representation using two more algorithms on three data sets.

1 Introduction

Exploitation of satellite images with high spatial and spectral resolution leads to a double problem. The first question is how to deal with the treatment of the huge quantity of available data (30 to 50 cm of spatial resolution on 10 to 60 km^2, which can represent several GB of data), while often exploiting precise measurements by geographers, who recover and process the samples of ground truth data with the goal of training or validation (*ground truthing*) ?

In spite of the very low spatial resolution, in rural as well as in urban areas, a pixel often contains various kinds of vegetation in a symbiotic relationship, one in relation to the other, and the resulting pixel is called a *mixel*. Expert measurements are often very localized on the ground with a resolution often lower than that of the image, but at the same time, increasingly precise. Composition percentages of certain samples are known and can be used to detail the contents of the pixels and to pass from the resolution problem of using standard classification (*one-to-one classification* where a pixel is equivalent to a class) to an *unmixing* problem (*one-to-N classification* or *subpixel classification*, [9]), i.e. to consider the relative proportions of the various components of the land cover types present in each pixel, in order to obtain an inverse Bidirectional Reflectance Distribution Function (BRDF) model, which gives the reflectance of a target as a function of illumination geometry and viewing geometry, [4, 16]).

In the literature, the problem of *subpixel classification* is mainly solved by two methods: Spectral Mixture Analysis and Fuzzy Classification [14]. For instance, in the Linear Mixture Model (LMM), an approach that was tried in many early studies [13, 15, 19], the BRDF of a non-homogeneous surface is expressed as the weighted sum of the BRDF of the homogeneous components of this surface [4]. In spite of its simplicity, many problems have been encountered in this approach, in particular, the fact that the models are not sufficiently robust for separating BRDFs with acceptable precision. Moreover, the choice of the pure pixels for training requires human intervention, which is not always free of errors. Other models exist such as Probabilistic Models (Maximum Likelihood, neural networks) and Geometric Models [10], which take into account the shape of the trees and the direction and distribution of the illumination from the sun, but usually they are much more complex and relatively recent [14, 10].

Conversely, unsupervised techniques have been proposed, consisting of the identification of proportions directly from the observed spectra, and some are based on simplex [21, 20]. Moreover, more recent studies, using the ICA algorithm, have modeled reflectances of the pixel for each wavelength as independent processes [18]. But these methods are very sensitive to noise; the displacement of some boundary points can considerably modify the results [17].

All these data constitute a significant problem with the search space, in learning well as in exploitation. This paper presents an evolutionary approach based on genetic algorithms, known to be scalable [11]. Relatively simple rules of classification are discovered which can be applied thereafter to other remote sensing images. The structure of this paper is as follows: section 2 presents the ICUX algorithm in a general way, section 3 gives the appointed representation for the genetic individuals, and sections 4 to 6 report in detail the genetic operators used. In section 7, the presented algorithm is compared to two approaches, one based on a neural network (Probabilistic Model) and the other on SVM. These approaches were chosen because they are known for their robustness in this field [5, 8]. And finally the conclusion.

2 The ICUX algorithm

The presented algorithm is based on a previous version called ICU [12, 1]. Very briefly, but we refer the reader to our previous publications, ICU is able to discover classification rules based on conjunctions of disjunctions of spectral intervals (see Fig. 1). The ICU rules are shorter than those which are produced with ICUX (ICU Unmix), but they only deal with the *one-to-one classification* problem. The integration of the class proportions in ICU is only possible by the integration of the geographers domain knowledge for validating the increasingly vast and precise knowledge acquired on the ground. This remains especially useful for rural areas, in which the delimitation of the objects is not as

contrasted as in urban cases, and for which an object or a *contextual* approach is more suitable.

The search of a solution for the *unmixing* problem begins at the definition then the adaptation to the subjacent problem of a genetic classifier taking as a parameter the reflectance values of a pixel, and returning, for each learned class, the proportion of these classes in the pixel. The selected representation for this classifier allows the encoding, formally by constraints, of the range of acceptable values of the spectrum for each class. These constraints are in fact conjunctions of disjunctions of intervals, and could be viewed as hyper-rectangles in the space of data. Thus learned knowledge could easily be representable for the purpose of presentation and validation by a human expert. The presence of disjunctions in the constraints makes it possible to solve in certain cases some non-linear problems. Finally, these rules are evolved directly from the data, without requiring preliminary preprocessing of the whole of the image, therefore this can represent a fast manner for classifying new images, provided that the learned rules are sufficiently robust.

3 Genetic individuals

3.1 Representation of a classifier

A classifier (a *genetic individual*) is composed of as many rules as there are classes to learn. Thus the adaptation of the classifier is done in parallel for all the classes. Each rule contains as many conditions C_i as there are bands in the image or exogenic attributes for each sample of data (see figure 1).

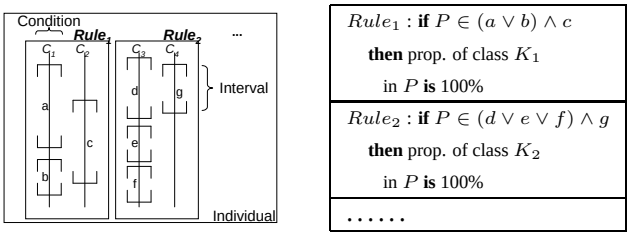

Figure 1: Genotypic and semantic representation of a genetic individual.

The conditions are linked together by conjunctions, for instance $(a \vee b) \wedge c$. Each condition contains a variable number of intervals, for which the domain of value and type (float, integer, boolean) depend on the data. The intervals are linked together by disjunctions. Thus each rule represents a constraint as a hyper-rectangle for a given class. A data is said to be in "*perfect matching*" if all the values respect the conjunction of disjunctions of intervals for a given class and is apart from the intervals for the other classes. For instance, if there are 5 classes to learn, a *perfect matching* for class 2 could be represented by the vector $[01000]$. To make it possible for the classifiers to report various concentrations of pure classes in the form of continuous percentages, a calculation of correlation is carried out when a datum does not respect the intervals of a given rule. The goal of the genetic algorithm is to adapt the size and position of these hyper-rectangles so that they conform to the model of concentration fixed by the expert. With condition components of the rules evolving in an independent way, it is possible for the classifier to return a null or a maximum concentration at the same time for all classes (if necessary), or to discover classes which overlap partially or are not in the data (correlated attributes). The following section presents the techniques used for quantifying the *matching* of a rule and thus of a classifier for a given pixel.

3.2 To match a rule

The *matching* function $\mathcal{M}(R, P) = r$ is defined for each rule R composing a classifier and returns a real between 0 and 1 corresponding to the proportion r of pure class contained in the pixel $P = [p_0, \ldots, p_i, \ldots, p_n]$ (n is the number of bands). The application of a classifier on a pixel consists in applying the various rules contained in this classifier and in collecting the various proportions for each class. For each interval $I_i = [a_i; b_i]$ $(0 \leq i \leq n)$ of rule R, the function $\mathcal{M}(R, P)$ calls upon a function $\mathcal{M}(I_i, p_i)$, which can be whatever, but which takes its maximum in the interval $[a_i; b_i]$. At the time of a disjunction of intervals, the largest value of $\mathcal{M}$ is retained.

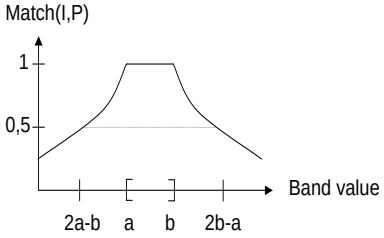

Figure 2: Representation of a satisfactory *matching* function $\mathcal{M}(I_i, p_i)$ for an interval $[a; b]$ and a given band value.

The function, which was retained in experimentation, is presented on the figure 2. The function is defined in the following way:

$$\mathcal{M}(I_i, p_i) = \begin{cases} 1 & \text{if } p_i \in [a_i; b_i] \\ \frac{b_i - a_i}{b_i - p_i} & \text{if } p_i < a_i \\ \frac{b_i - a_i}{p_i - a_i} & \text{if } p_i > b_i \end{cases}$$

where p_i is the value of the pixel P for the band i and $[a_i; b_i]$ the values of the corresponding interval defined in the genetic individual. $\mathcal{M}$ is worth 1 for a perfect *matching* and tends towards 0 if not.

This function is only dependent on the size of the intervals, therefore independent of the scale of the data. The genetic algorithm will have to adapt the position and the size of the intervals directly according to the analyzed spectral context. The function is constant in the interval $[a_i; b_i]$ so as not to influence the performance of the rules if the distribution of the spectrum is not homogeneous (checked assumption). Other functions (Gaussian, ...) are being studied.

The *matching* function for a whole rule R takes into account the score obtained for each band and is defined in the following way:

$$\mathcal{M}(R,P) = \sqrt[\epsilon]{\frac{\sum_i^n \mathcal{M}(I_i,p_i)^\epsilon}{n}}$$

where n is the number of bands and ϵ is a strictly positive parameter controlling the influence of the extreme values (for instance, $\epsilon = -2$ corresponds to a distance within the context of the LMS error estimation).

The matching of a pixel P with the totality of the classifier I is defined as being the standardized vector of the scores obtained for each rule of the classifier. Each component of the vector indicates the proportion of the corresponding class in the pixel P.

$$\mathcal{M}(I,P) = \left(\frac{\mathcal{M}(R_1,P)}{\sum_i \mathcal{M}(R_i,P)}, \cdots, \frac{\mathcal{M}(R_k,P)}{\sum_i \mathcal{M}(R_i,P)}, \cdots, \frac{\mathcal{M}(R_n,P)}{\sum_i \mathcal{M}(R_i,P)}\right)$$

3.3 Verification of the conditions

When the rules are created or modified (by the crossover or the mutation operators), it is possible that the intervals are not coherent any more. A random modification of one of the terminals or a more significant part of the rule can lead to:

- a terminal MIN higher than a terminal MAX,
- a negative terminal value or higher than the acceptable range for the values of the pixels,
- auto-intersecting intervals (for instance, [25;105] $\wedge$ [80;112] can be merged in [25;112] and [12;13] $\wedge$ [14;18] can be merged in [12;18]).

ICUX contains a procedure for checking that such cases do not occur, and correcting them if necessary by a fusion or an inversion of the terminals (*fusion process*).

4 Initial population

To avoid a significant problem with the search space (within the framework of remote sensing images) and to optimize the usually long processing time of the genetic algorithms, a procedure of initialization of an individual adapted to the image and the expertise provided (*mixels*) has been set up. In particular, the number of disjunctions of intervals for each band and each class are estimated by taking into account, at the same time, the image and the expertise.

Let n be the number of pure classes and m the number of bands in the image or attributes in the data to be analyzed. For each pixel P ther is a corresponding expertise $c_1, \ldots, c_n \in \mathbb{R}$ for each pure class $K_1, \ldots, K_n$. The value of the expertise c_i of this pixel for a pure class K_i is seen as the contribution of the value p_j of the band j of the pixel P for this class K_i. It is expressed by $h(p_j, j) = c_n$. For instance, for a pixel $P = [1000; 1500; 1300]$ and an expertise $C = [0.7; 0.1; 0.05; 0.03; 0.12]$, it is considered that the contribution of the value '1000' for the first band and the first class is 70%.

If a class K_i - a band j and a disjunction of intervals $DI = [a_1; b_1] \vee \ldots \vee [a_m; b_m]$ - is set, the average contribution $\mu_1(i, j, DI)$ from all the instances of pixels which *match* DI and the average contribution $\mu_0(i, j, DI)$ from all the instances of pixels which *do not match* DI can be calculated.

The algorithm then sets an iteration step on the interval $[MIN; MAX]$ of the observed values of the pixels for a given band, and examines in an exhaustive manner the set of disjunctions that it is possible to build. For instance, if for a band j the values of the pixels of the image are distributed in the interval [1;1000] and a step is chosen to cut out the search space in 2 equal parts, disjunctions '00', '01', '10' and '11',will be successively examined, i.e. respectively disjunctions: '$\emptyset$' (void), '$[501; 1000]$', '$[1; 500]$' and '$[1; 500] \vee [501; 1000]$' (which is reduced in $[1; 1000]$ after the application of the fusion process, see the section 3.3). For each one of these disjunctions, the contribution μ_1 from the values **in** these intervals and that (μ_0) obtained **apart from** these intervals. The algorithm no longer has to select which disjunction induces the maximum variation of contribution $\varepsilon = \mu_1 - \mu_0$, which makes it possible to be sure that the contribution of a given band for a given class is maximal in the disjunction and minimal outside.

A classifier called 'initial seed' is built by accumulating rewarded disjunctions for each band and each class, according to the above representation. The initial pool is then obtained by duplicating this seed and by adding slight noise as many times as there are individuals in the initial population.

These contributions are dependent on the class and the analyzed band. The final rule (for a given class) consists of a conjunction of disjunctions obtained for each band to be analyzed. In practice, the exhaustive examination of a search space cut out in 20 equal parts (see figure 3) require 2^{20} (a million) iterations, which represents a computing time of less than one second on a recent machine. Moreover, only half of the iterations are really carried out, because by permuting μ_0 and μ_1, the contribution of complementary disjunctions can be deduced (thus disjunction '10' substitutes the disjunction '01', for instance). The maximum number of intervals of a disjunction is limited by cutting the search space (10 per 20 parts, because the contiguous intervals are merged).

One of the interesting characteristics of this algorithm is the fact that it can produce a variable number of *holes* (within the meaning of the number of intervals in a disjunction), by taking into account, at the same time, the data (all bands) and the continuous values of the expertise.

However, in spite of the quality of the initialization, refining via the genetic algorithm is necessary for a couple of reasons: firstly, because of the partiality introduced by the user-fixed value for the iteration step and then because of the non-homogeneous distribution of the values of the attributes within the problem space. The following section presents the genetic operators which have been studied for refining the value of the terminals of the intervals apart from the rigid framework fixed by the iteration step, by cutting them out if the selected resolution for the step is not high enough.

Band	Disjonction	μ_1	μ_0
b= 0	[.....x.....]	0.0677	0.0000
b= 1	[.....x.....]	0.0619	0.0000
b= 2	[.....x.....]	0.0507	0.0000
b= 3	[.....x.....]	0.0547	0.0000
b= 4	[.....x....]	0.0700	0.0000
b= 5	[.....x.xx...]	0.0868	0.0000
b= 6	[.....xxx..]	0.0951	0.0000
b= 7	[.....xxxx.]	0.0996	0.0000
b= 8	[.....xxxxx.]	0.1650	0.0963
b= 9	[.....xxxxx.]	0.0951	0.0000
b= 10	[.....xxxxx.]	0.0949	0.0000
b= 11	[.....xxxxx.]	0.0939	0.0000
b= 12	[.....xxxxx..]	0.0939	0.0000
b= 13	[.....xxx.]	0.2200	0.0633
b= 14	[.....x]	0.3300	0.0382
b= 15	[.....x.....]	0.3300	0.0581
b= 16	[.....xx.....]	0.0563	0.0100
b= 17	[.....x.x.....]	0.0504	0.0000
b= 18	[.....x.....]	0.0479	0.0000
b= 19	[.....x.....]	0.0502	0.0000

Figure 3: Example of the disjunctions found for class '*Spartima*' on a remote sensing image of 20 bands. The 'X' represents the intervals selected in the disjunction. The spectrum of the class start appearing vertically. Note the small spectrum-*hole* for bands 5 and 17.

5 The crossover and the mutation operators

The crossover operator functions via the exchange of the genetic material between the two parents. If the two parents are better than the average, the assumption is made that the resulting genetic material will contain a copy of the powerful genes and will be transmitted to the children. Thus its role is to exploit the knowledge acquired by the individuals of the current population.

In ICUX, the crossover is uniform. A class K is selected randomly, as well as the respective rules $R_{1,K}$ and $R_{2,K}$ in the two individuals to be crossed. For each band j, the condition $C_{1,K,j}$ of $R_{1,K}$, as well as the condition $C_{2,K,j}$ of $R_{2,K}$ have a probability P_{cross} for being crossed. When the crossover operator is selected, an interval is selected randomly in each condition and it is exchanged. Then, the *fusion process* is applied to the modified rules.

The mutation operator causes a small number (given by a probability P_{mut}) of minor and random modifications within genes of an individual. On the basis of the assumption that a limited number of mutations makes it possible to reach any point of the search space, the mutation operator is seen as an exploratory operator. It is also used for maintaining diversity in the population of individuals. Here, this operator is applied to three levels: a whole condition, an interval, or one of the terminals of an interval.

The mutation of a whole condition consists of removing, for a given rule, the condition corresponding to one of the bands with a probability $P_{mut,cond}$. Several interests are: the simplification of the rule, the generalization, and the purification (if the algorithm considers this band as disturbed or embracing poor information). The existence of this noisy band in the image disturbs the process of training as well as the convergence of the algorithm.

The second type of mutation consists of selecting an interval, with a probability $P_{mut,int}$, in one of the conditions C, then eliminating it, cutting it into two, replacing it if the number of authorized intervals for this particular rule have been reached, or adding an interval in C. The cutting of an interval consists of randomly selecting a point of cut and then separating the parts if their size is sufficiently broad (for instance, $[10; 100]$ is cut out in two non-consecutive intervals $[10; 15] \wedge [17; 100]$ but $[1; 3]$ is cut out in $[1; 2] \wedge [2; 3]$). This is useful to explore the continuous values of the spectrum below the resolution of the iteration step of the initialization operator.

Lastly, the mutation of the terminals of an interval with a probability $P_{mut,term}$ modify the two terminals by shifting, widening, or modifying only one of the two terminals in the following way. Let ω_1 and ω_2 be two real random variables in the interval $[0; 1]$, α the quantity of added noise, β the increase in the size of the intervals, and γ the displacement of the intervals. An interval $[a; b]$ is modified randomly in $[a'''; b''']$ as follows.

$l = \alpha(b - a) \quad a' = (a - l) + 2\omega_1 l \quad b' = (b - l) + 2\omega_2 l$
(adding noise)

$a'' = \frac{(a'+b')}{2} - \frac{\beta(b'-a')}{2} \quad b'' = \frac{(a'+b')}{2} + \frac{\beta(b'-a')}{2}$
(centered-widening or -reduction of the size of the intervals)

$a''' = a'' + \gamma(b'' - a'') \quad b''' = b'' + \gamma(b'' - a'')$
(left *shift* or right *shift*)

This operator allows for refining, step by step, the overall spectral constraint discovered by the other operators for a given class. The modified rules are all systematically validated by the *fusion process*. In the presented system, the probabilities assigned to each operator are dynamically adapted, according to the standard deviation of the performance measurement (*fitness*) observed on the whole pool of individuals. If the standard deviation decreases, the population becomes monotonous and there is a risk that an emerging individual, which may be a local minimal, may colonize the remainder of the population. In this case, the rate of mutation gradually went up ($P_{mut} = P_{mut} + (1 - P_{mut}) * T_{up}$, it is an increasing function, limited by 1). Conversely, a too high standard deviation means that the rate of mutation is not low enough (within the meaning of the evolutionist theory), and it is decreased with each iteration ($P_{mut} = P_{mut} - P_{mut} * T_{up}$, it is a strictly positive decreasing function). During our experiments, the rate is stabilized around 16% for an initialization at 30%.

The usefulness of the mutation operator is in the non-linear traversing of the search space. Thus, a left *shift* of 10% of the values of an interval is obtained with the same effort (an operator callback) as the addition of an interval in a particular disjunction, which can lead to a non-linear modification of the space captured by the rules, but a continuous improvement of the recognition quality of the asso-

ciated rule.

The next section presents the calculation of the evaluation function, based on the *matching* values of each rule.

6 The genetic algorithm

The genetic algorithm used has the framework of the standards well known in the literature. The fitness function is the standardized average of all the values obtained by the evaluation function. This function defines the performance of a particular individual compared to the remainder of the population. In ICUX, an individual I is evaluated by computing the correlation between the vector obtained by $\mathcal{M}(I, P)$ and corresponding expert information for a given pixel, then by calculating the average for all the pixels available in the training set. The correlation is defined as the square root mean of the squares of the distances between the found proportions compared to the expert.

Many strategies of selection were tested and are available to the user (random selection, elitism, eugenism, based on the rank, roulette wheel, tournament, ...). Most of the genetic algorithms also define some other parameters, such as the rate of crossover and of mutation, the number of individuals in the initial population, and the maximum iteration number before stopping the algorithm (called number of *generations*). Various values for the majority of these parameters were tested (the average values taken in the acceptable intervals of parameters obtained by tuning on completely different data files were finally selected), those retained for the case study are presented in the following section.

7 Comparison of several methods

To evaluate the performance of the ICUX algorithm, it was tested on three data sets, and compared with two other methods known to be robust. The first two data sets are two images of San Felice (Lagoon of Venice), with different resolutions. CASI sensor (date 2002, 15 bands, 754x293 pixels, resolution 1.3 m^2, 6 classes) and MIVIS sensor (date 2004, 20 bands, 396x170 pixels, resolution 2.6 m^2, 7 classes) were used. In these data, the expertise of each class is expressed in percentages with an accuracy of 5%. The last data set is a database of 846 samples of vehicles described by 18 continuous spatial criteria (surface, perimeter, compactness, elongation...) and is available on the UCI site [6]. This data set was used to compare the performances of the various methods on a benchmark problem.

The results for the testing set (50% of all of the data) are presented in the following tables. Accuracy is the mean of the diagonal of the confusion matrix, PPA is the *positive predicted accuracy* (measures a reliability of correctly classified instances by a classifier to all instances associated by a classifier to a given class) and SPE is the *specificity* (measures the rate of correctly classified instances as not being in a given class).

The parameters (described in this paper) for ICUX are as follows: P_{cross} = 0.7, P_{mut} = 0.15, $P_{mut,cond}$ = 0.3,

	Accuracy	PPA	SPE
Neural network	0.997	0.981	0.931
SVM-R	0.974	0.737	0.499
ICU	0.889	0.867	0.797
ICUX	0.972	0.974	0.933

Figure 4: Results obtained for the CASI image.

	Accuracy	PPA	SPE
Neural network	0.776	0.645	0.597
SVM-R	0.978	0.782	0.777
ICU	0.757	0.620	0.701
ICUX	0.863	0.855	0.765

Figure 5: Results obtained for the MIVIS image.

	Accuracy	PPA	SPE
Neural network	0.722	0.682	0.704
SVM-R	0.898	0.869	0.869
ICU	0.537	0.546	0.531
ICUX	0.619	0.570	0.609

Figure 6: Results obtained for the vehicles database.

$P_{mut,int}$ = 0.2, $P_{mut,term}$ = 0.4, ϵ = 0.5, T_{up} = 0.1, initialization in 20 equal parts, 300 individuals, 2000 to 5000 generations and selection by rank for the crossover and the mutation operators.

For the algorithm based on neural networks, a learning rate of 0.1, 100000 iterations, 1 hidden layer of 7 to 15 neurons, an incremental method for the learning, and a symmetrical sigmoid activation function were choosen. The exit layers represent the expert continuous values and there are as many exit neurons as there are classes to be learned. The free library in *C*, named the Fast Artificial Neural Network Library (FANN, [2]), was used. The chosen neural network topology was simple but efficient (see Fig. 4).

For the algorithm based on Support Vector Machine Regression (SVM-R), the RBF kernel was used; and the parameters C and γ were discovered for each data set using free software and a step by step optimization algorithm in *Python* presented on the site of the LIBSVM [7].

Comments The image of the MIVIS sensor was most difficult to analyze for several reasons: some additional bands; less pixels in the image, thus less samples to classify (each time the whole training set consisted of less than 2% of the image); the first bands of MIVIS are slightly more disturbed than for CASI; and finally some classes of vegetation were no longer present on the saltmarsh in 2004, which caused a drop in the quality of the training data. Also noticed was a lower performance on the vehicle database because the values of the data come grouped, without particular relation between them (that is due in particular to the orientation of the various objects of the base). Also observed was the increase in total performance of ICUX compared to the previous version ICU and the comparable level of performances between ICUX and the algorithm based on

neural network. Compared to SVM-R, the performances of ICUX are variable according to the quality of the analyzed image. When the values are gathered, a system based on disjunctions proves to be more useful. Moreover, this kind of classifier would make it possible to expose the contents of the rules to a human expert contrary, for example, to a neural network (often treated as a *black box*).

8 Conclusion

In this paper, a new representation for a genetic individual able to deal with the unmixing problem without using a subjacent linear model (LMM) has been proposed. This representation has consequently required the redefinition of certain genetic operators. The creation process of the initial population of classifiers has been optimized by taking into account, at the same time, the disjunctions of spectra present in the data and continuous expertise. The system benefited from the new operators and made it possible to reach a better quality of recognition.

The representation of independent rules allows not only the exploitation of mixed expert information but also of partial information (attributes not well informed or absent). For example, if the expert does not know the precise mixture of a given pixel, but he is sure that there is not a given class, the genetic algorithm can be asked to optimize the contribution $[0;1] \wedge [0;1] \wedge [0;0]$ for this pixel, i.e. a contribution from 0 to 100% for the first two classes and a null contribution for the last class. This kind of expertise will have to be studied more deeply hereafter, because for the moment this expert information still remains very rare (oral support).

This approach can be applied to other kinds of data provided in the format of tuples of attributes with values in $\mathbb{R}$. The usage of other data sets will allow us to specify the influence of certain matching functions and to develop specific optimization functions tailored to a given problem.

Acknowledgments

The authors would like to thank Enrica Belluco and Sergio Ferrari from the UNPADU University (Padova, Italia) for the multispectral data sets used (images of San Felice and continuous expertise acquired during the TIDE project [3]), and Pete Mowforth and Barry Shepherd from the Turing Institute (Glasgow, Scotland) for the data file 'vehicle silhouettes'.

Bibliography

[1] Discovery of classification rules: evolutionnary classifiers, software available at http://lsiit.u-strasbg.fr/afd/logiciels/icu/.

[2] Fast artificial neural network library, software available at http://fann.sourceforge.net/.

[3] Tidal Inlets Dynamics and Environment, Research Project Supported by the European Commission under the Fifth Framework Programme, contract n° EVK3-CT-2001-00064, http://www.istitutoveneto.it/tide, 2001-2005.

[4] M. J. Barnsley, P. Hobson, Z. Hesley, K. Evans-Jones, T. Quaife, P. Lewis, M. Disney, J-P. Muller, A. Strahler, W. Lucht, and A. Hyman. Determination and validation of land-surface biophysical properties using SPOT-4 vegetation and HRVIR: Interim report. 1998.

[5] J. A. Benediktsson, P. H. Swain, and O. K. Erase. Neural network approaches versus statistical methods in classification of multisource remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 28:540–551, 1990.

[6] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, available at http://www.ics.uci.edu/∼mlearn/mlrepository.html, 1998.

[7] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm, 2001.

[8] J. A. Gualtieri and R. F. Cromp. Support vector machines for hyperspectral remote sensing classification. *Proceedings of the 27th AIPR Workshop: Advances in Computer Assisted Recognition*, pages 221–232, 1998.

[9] R. L. Huguenin, M. A. Karaska, D. van Blaricom, and J. R. Jensen. Subpixel classification of bald cypress and tupelo gum trees in thematic mapper imagery. *Photogrammetric Engineering and Remote Sensing*, 63(6):717–725, 1997.

[10] C. Ichku and A. Karnieli. A review of mixture modelling techniques for sub-pixel land cover estimation. *Remote Sensing Reviews*, 13:161–186, 1996.

[11] H. Kargupta. Gene expression and scalable genetic search. *In Advances in evolutionary computing: theory and applications*, pages 293–319, 2003.

[12] J. Korczak and A. Quirin. Evolutionary mining for image classification rules. In *EA 2003 : 6th International Conference on Artificial Evolution*, Marseille, 2003.

[13] K. T. Kriebel and P. Koepke. Improvements in the shortwave cloud free radiation budget accuracy. Part II: Experimental study including mixed surfaces albedos. *Journal of Climate and Applied Meteorology*, 26:396–409, 1987.

[14] A. Lucieer and M. Kraak. Interactive and visual fuzzy classification of remotely sensed imagery for exploration of uncertainty. *International Journal of Geographical Information Science*, 18(5):491–512, 2004.

[15] R. Meerkoetter. Reflection functions for inhomogeneous land surfaces. *Applied Optics*, 29:4192–4198, 1990.

[16] A. A. Nielsen. Spectral mixture analysis: Linear and semi-parametric full and iterated partial unmixing in multi- and hyperspectral image data. *International Journal of Computer Vision, 42(1-2), 17-37, and Journal of Mathematical Imaging and Vision, 15(1-2), 17-37*, pages 613–619, 2001.

[17] L. Parra, C. D. Spence, P. Sajda, A. Ziehe, and K.-R. Muller. Unmixing hyperspectral data. *Advances in Neural Information Processing Systems*, 12:942–948, 2000.

[18] B. A. Pearlmutter and L. C. Parra. Maximum likelihood blind source separation: A context-sensitive generalization of ICA. *In Advances in Neural Information Processing Systems*, 9:613–619, 1997.

[19] W. Qin, S. Goel, and B. Wang. The hotspot effect in heterogeneous vegetation canopies and performances of various hotspot models. *Remote Sensing Reviews*, 14:283–332, 1996.

[20] J. Settle and N. Drake. Linear mixing and the estimation of ground cover proportions. *International Journal of Remote Sensing*, 14(6):1159–1177, 1993.

[21] M. O. Smith, J. B. Adams, and A. R. Gillespie. Reference end-members for spectral mixture analysis. *Proceedings 5th Australasian Remote Sensing Conference*, pages 331–340, 1990.