Evolutionary Mining for Image Classification Rules

Jerzy Korczak, Arnaud Quirin

LSIIT, CNRS/Université Louis Pasteur,
Pôle API, Boulevard Sébastien Brant F-67400 Illkirch cedex
{korczak,quirin}@lsiit.u-strasbg.fr

Abstract. In this article, an approach for creating image classification rules using evolutionary operators is described. Classification rules, discovered by application of a genetic algorithm on remote sensing data, are able to identify spectral classes with comparable accuracy to that of a human expert. Genetic operators and the fitness function are detailed, and then validated for hyperspectral images (more than 80 spectral bands). Particular attention is given to mutation operators and their efficiency in the creation of robust classification rules. In our case studies, the hyperspectral images contain voluminous, complex and frequently noisy data. The experiments have been carried out on remote sensing images covering zones of Lagoon of Venice and the city of Strasburg, France. It has been shown that the evolution-based process can not only detect and eliminate noisy spectral bands in remote sensing images but also produce comprehensive and simple rules which can be also applied to other images.

Keywords : Remote sensing image, classification rules, high resolution image, hyperspectral image, supervised learning, evolutionary learning, genetic algorithm.

1 Introduction

The design of robust and efficient image classification algorithms is one of the most important issues addressed by remote sensing image users. For many years, a great deal of effort has been devoted to generating new classification algorithms and to refine methods used to classify statistical data sets (Bock, Diday, 1999). At the time of this writing, relatively few workers in the machine learning community have considered how classification rules might be genetically discovered from raw and expertly classified images. In this paper, a new data-driven approach is proposed in order to discover classification rules using the paradigm of genetic evolution.

The unique source of information is a remote sensing image and its corresponding classification furnished by an expert. The images have been registered by various satellites (e.g. SPOT, LANDSAT, DIAS, ROSIS) that use different cameras having various spectral and spatial resolutions (Weber, 1995). These types of remote sensing images generally contain huge volumes of data, for instance an image of DAIS contains 79 bands of each one 2.8 Mbytes. And, sometimes they are very noisy due to coarse spatial resolution or unfavorable atmospheric conditions at the time the images are acquired. In addition, data may be also erroneous due to inexperienced operators of the measurement devices.

The aim of this research is to detail an evolutionary classification method applied to remote sensing images. More about evolutionary classifiers can be found in (DeJong, 1988) and (Ross, Gualtieri et al., 2002). As stated, the approach to discover classifiers is data-driven because the formulated classification rules are generated from data and are able to adapt themselves according to this available data, environment, and the evolution of classes. In remote sensing, the initial population of classification rules is randomly created from raw images and given classes, and then evolved by a genetic algorithm until the acceptable classification accuracy is reached.

In remote sensing literature, several classification approaches are presented, namely:

- pixel-by-pixel, each image pixel is analyzed independently of the others according to its spectral characteristic (Fjørtoft, Marthon et al., 1996),
- zone-by-zone, before classification, the pixels are aggregated into zones, the algorithms detect the borders of the zones, delimit them by their texture, or their repetitive patterns (Kurita, Otsu, 1993),
- by object, this is the highest level of recognition, the algorithms classify semantic objects, detect their forms, geometrical properties, spatio-temporal relations using domain knowledge (Korczak, Louis, 1999).

Our approach uses spectral reflectances; therefore, discovered classification rules are only able to find spectral classes rather than semantic ones. This spectral component of class description is essential to well recognize thematic classes. The approach has been validated using our software environment, called *I See You* (ICU). In this software, the object representation is not too sophisticated but it offers a high degree of freedom in description of symbolic expressions of rules and definition of genetic operators. The goal was to evaluate the capacity of the genetic approach to handle problems of over-generalization and over-fit in highly noisy and complex data. The ICU is a genetic-based classifier, where we have adapted and extended ideas of learning classifier systems, such as XCS (DeJong, 1988; Wilson, 1999), the s-classifiers, and "Fuzzy To Classify System" (Rendon, 1997). We have also been inspired by the works of Riolo (Riolo, 1988) on gratification and penalization, and of Wilson (Wilson, 1999) on the exploration of the search space.

The paper is structured as follows. The basic concepts of image classification rules are introduced in Section 2. Section 3 details the discovery process of the classification rules. In this Section, the behavior of genetic algorithm functions is explained. Finally, two case studies on real remote sensing data are presented in Section 4.

2 Concept of classification rule extracted from remote sensing images

In general, classification rules are symbolic expressions and describe conditions to be held and actions to be taken if the conditions are satisfied. It must be underlined that in our approach the rules are discovered by an evolutionary process and are not given a priori by a domain expert.

From a functional point of view, a rule represents a piece of knowledge about a class by a conditional expression, such as *if* <*conditions*> *then* <*class*>. The "*conditions*" part described an entry information in the system such as value, color, form, shape, etc, corresponding to conditions that must be fulfilled in order to activate this rule. The "*class*" part defines the class of the instance currently treated by the rule when the appropriate conditions were satisfied. We assert that the evolved rules must be rapidly evaluated and easy to interpret by any user. As a result, condition representation using the concept of an interval could be fully adequate for remote sensing image classification. In terms of machine learning, the rules have to be maximally discriminant generalizations, meaning that they have to cover the maximum pixels belonging to a given class and the minimum pixels belonging to another classes.

Before rule specification, recall that a pixel is encoded as a spectral vector, describing values of reflectance for the n bands of the remote sensing image, i.e. a pixel can be considered as a point in a R^n space :

$$<pixel> := [b_1\ b_2\ b_3 \ldots\ b_n] \qquad (1)$$

In our system, the condition for any rule is built on the concept of spectral intervals defining a given band, corresponding to a given class. Such intervals are a pair of integer numbers, between 0 and the maximum possible value for a pixel of a given band (i.e. 65536 for pixels defined on 16 bits). This solution allows to partition the space of the spectral values in two ranges: the first containing the pixel values which corresponds to a given class, and the second containing the remainder.

To precisely specify the class definition, a set of intervals is defined for each band of the remote sensing image. Taking into consideration all bands, the condition part is defined as a set of hyper-rectangles in a R^n space :

$$<condition> := \bigwedge_{i=1}^{n} \bigvee_{j=1}^{k} (\mathrm{m}_i^j \le b_i \le \mathrm{M}_i^j) \qquad (2)$$

where m_i^j and M_i^j are, respectively, the minimal and maximum reflectance values allowed for a pixel belonging to a class C for the band i. k is a parameter which defines the maximum number of disjunctions allowed.

These intervals are not necessarily disjunctive. By experiments, we have found that if we allow the genetic algorithm to create non-disjunctive intervals, instead of merging them, the results of genetic operators are more interesting. We have also noticed that merging intervals significantly diminishes the number of intervals, and in the same time, reduces the possibilities to create more efficient rules. To illustrates the concept of interval merging, $E = [11;\ 105] \vee [138;\ 209] \vee [93;\ 208]$ corresponds after merge operation to $E = [11;\ 209]$.

To satisfy a rule, a pixel has to match at least one spectral interval for each band. Logically speaking, to associate a pixel to a class, its values have to satisfy the conjunction of disjunctions of intervals that define a condition part of the classification rule.

This representation of the rule has been chosen mainly because of its simplicity, compactness and uniform encoding of spectral constraints. During experimentation, this representation has also demonstrated rapid execution of genetic operators and efficient computing. Of course, one may specify more complex structures using spatial properties of the pixel, with respect to the pixel neighborhood. Also, one may include features resulting from thematic indices or mathematical operators applied to pixel environment. These semantically extensions are interesting, however they not only require more sophisticated genetic operators, but also more powerful computers to perform the calculation in an acceptable amount of time.

3 From the rule creation to the evolution

3.1 Genetic algorithm

In order to efficiently develop the classification rules, a genetic algorithm initializes interval values according to spectral limits of the classes designated by an expert, for valid zones of the remote sensing image. Initial classification rules are created based on the extreme maximum and minimum values for defined spectral intervals of each class. It should be noted that by this initialization, rule searching is considerably reduced, and initial intervals are very close to the final solution. More about initialization algorithms can be found in (Kallel, Schoenauer, 1997). During the process of evolution, the initial spectral limits are slightly perturbed by adding a random value to lower and upper spectral limits. Hence, the initial population of classification rules is quite diversified.

A Michigan-like approach is used to discover independently a classification rule for each class. A major reason for choosing this approach is the efficiency of computations; that is, the process of rule discovery is not perturbed by other rules.

The quality of classification rules is based on a comparison of these results with the image classified by an expert. If pixels covered by the rule perfectly overlap those indicated by an expert, then the system assigns the highest quality value to the rule; otherwise, in the case of some mismatching, the quality factor is reduced (between 0 and 1). An associated fitness function will be detailed in the next section. During the evolution process, the rules are selected according to the quality for a given class. It should be noted that it is also possible to define global system quality based on rule classification qualities. The process of rule evolution is defined in the algorithm below.

Algorithm 1. Process of rule discovery.

R is a classification rule and P, P' and P" populations of classification rules.

```
R: = INITIAL_RULE(images)  // Creation of rule according to spectral extremes
P: = INITIALIZATION(R)     // Random perturbation of rules
EVALUATION(P)              // Calculation of the fitness function for each rule
do while TERMINATION_CRITERION(P) = false
   P' : = SELECTION_X(P)                  // Selection for crossover
   P' : = CROSSOVER(P') U COPY(P)
   P'' : = SELECTION_MUT(P')              // Selection for mutation
   P'' : = MUTATION(P'') U COPY(P')
   EVALUATION(P'')
   P: = REPLACEMENT(P,P'')                // New generation of rules
end_while
```

Result: A print of the classification rule R for a given class, statistics and quality measures for the discovered rule.

As mentioned before, this algorithm must be designed to run independently for each class. This allows for obtaining rules according to user requirements without the necessity of carrying out computations for all classes with the same level of quality. This also allows to preserve the previously generated rules, as well as to introduce of new ones. Further, the user may define a hierarchy of classes and specialize some rules while respecting newly created sub-classes with different levels of classification quality.

3.2 The evaluation function

The evaluation function serves to differentiate the quality of generated rules and guide genetic evolution. Usually, this function depends strongly on application domain. In our work, we define a pixel that the rule classifies as being in the class when the expert classifies the pixel as in the class as a true positive. Conversely, we define a pixel that the rule classifies as being not in the class when the expert classifies the pixel as not in the class as a true negative. Other pixels are said to be correctly classified.

We normally use as a quality measure the proportion of pixels that are correctly classified by the rule. In some cases, when classes are under- or over-represented, we care more about one misclassification than another. In these cases, we use $\alpha.p_{tp}+(1-\alpha).p_{tn}$ as a quality measure, where p_{tp} is the proportion of true positive pixel classifications by the rule (called *sensitivity*), p_{tn} is the corresponding proportion of true negatives (called *specificity*), and α is a parameter that lets us adjust the relative weight given to true positives and false negatives. By default the value of the coefficient α is fixed to ½.

The proposed function shows a number of advantages; it is independent of the pixel processing sequence, invariant of the size of classes, and efficient for class discovery with a highly variable number of pixels.

The evolution process converges according to some statistical criteria indicating if the current rule is near to a global optimum or if the population of rules will not evolve anymore. The termination criterion of the algorithm leans on the statistics of rule quality evolution. In our system, we take into consideration not only the evolution of quality of the best discovered rule, but also the minimum acceptable quality defined by a user, the process stability measure and a maximal number of generations to run. If one of these criteria is satisfied, then the process is stopped.

The most difficult question is whether the quality of a rule is not continuing to evolve. To detect stabilization of the quality evolution, instead of taking into account the best rule generated recently we have based our heuristics on statistics regarding quality evolution of the best discovered rules in a time. For example, let Q_k be the quality of the best rule obtained during the last k generation, and Q_o be the quality of the best rule of the current generation. Formally, the algorithm is stopped if the following equation is satisfied:

$$\left| \frac{\sum_{k=1}^{P} Q_k}{P} - Q_0 \right| \leq E \tag{3}$$

where P represents the maximum period of quality stabilization, and E is a maximal variation of this stabilization compared with the current quality.

It is important to have an initial population of rules within the vicinity of the solution to be found. We have proposed two algorithms allowing for the generation of a diversified pool of rules close to the expert hidden classification rule. The first, called MinMax, creates maximum intervals covering all the pixels belonging to a given class, and the second algorithm, called Spectro, integrates the spectral distribution density and interval partitioning.

With respect to software engineering, the genetic algorithm has been structured into layers corresponding to consecutive genetic operations (e.g. selection, mutation, crossover and replacement). This modular approach makes the program maintenance and future extensions much easier.

3.3 Genetic operators

One of the most important tasks while designing a genetic algorithm is to invent operators that will create new potential solutions. All of our operators have been adapted to the rule representation, and they have been validated on remote sensing images.

Selection of classification rules. In general, selection is the operation of allocating reproductive opportunities to each rule. The reproductive force of a rule is expressed by a fitness function that measures relative quality of a rule by comparing it to other rules in the population. There are many methods for selecting a rule (Blickle, Thiele, 1995). In our system, the selection operator is applied in the following cases:

- choosing the rule to be reproduced for crossing, or muting;
- repetition of the rule, depending on whether it completes the genetic pool after having completed the crossover;
- preservation of a rule from the former genetic pool for the next generation;
- elimination of a rule in a newly created genetic pool based on an assigned rank.

Selection methods are well known: the roulette wheel, ranking, elitism, random selection, the so-called tournament, and eugenic selection. Our experiments have shown that roulette wheel selection is most advantageous for the reproductive phase, but the tournament strategy with elitism is best for the generational replacement scheme.

Crossover of rules. Crossover requires two rules, and cuts their chromosomes at some randomly chosen positions to produce two offspring. The two new rules inherit some rule conditions from each parent rules. A crossover operator is used in order to exploit the qualities of already generated classifiers. Each result of the crossover process has to be validated. Consistency of the various rule attributes (border limits violation, over-passing, etc) is carried out respecting the intervals boundaries. However, merging not only decreases the number of intervals in the rules, but also generates some information loss. In fact, in order to avoid a premature convergence of rules, it is generally important to preserve for the following generation two distinct intervals instead of a single aggregated one. On the other hand, it is also interesting to note that the positive or negative effects of an interval on the quality of the rule can be related to other intervals encoded in the classification rule.

Mutation of rules. The mutation operator plays a dual role in the system: it provides and maintains diversity in a population of rules, and it can work as a search operator in its own right. The mutation processes a single classification rule and it creates another rule with altered condition structure or variables. The mutation operator for several may be applied on three levels: band level, interval level and border level. Figure 1 shows the different variants of mutation as applied to remote sensing images.

Band mutation consists of a deletion of spectral bandwidth in a chosen classification rule. Its interest is twofold; firstly, the *band mutation* allows to simplify and generalize a rule; secondly, it allows to eliminate of noisy bands that frequently appear in hyper spectral images. The existence of noisy bands significantly perturbs the learning process, as well as the process of evolution convergence.

Interval mutation allows for a chosen band to add, eliminate or cut an interval in two spectral ranges. In case of addition, the new rule is completed by a new interval centered randomly with a user-defined spectral width. The cutting of an interval is done by random selection of a cutting point within the interval (for example, the cutting of *[10;100]* can generate two intervals: *[10;15]* and *[16;100]*). Interval mutation such as this allows splitting of continuous spectral ranges. And, this allows for the definition of a spectral tube in which spectral values of the pixels belong to a given class.

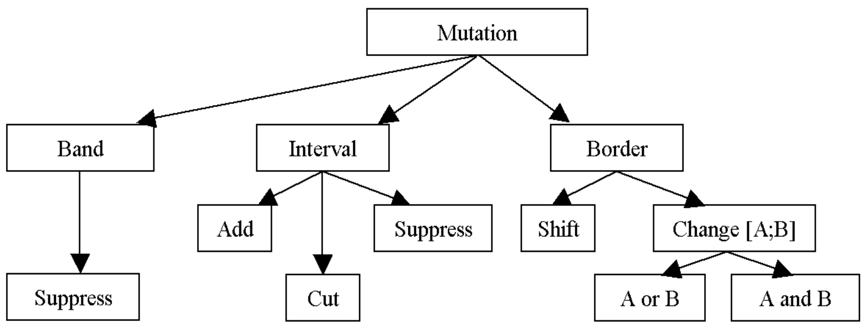

Fig. 1. Mutation operators

Finally, *border mutation* modifies both boundaries of an interval. This mutation refines the idea of targeting spectral tubes carried out by the other types of mutation. It is worthwhile to note that the mutated rules are systematically validated.

In our system, mutation operators are dynamically adapted. Adjustment is related to the probability of each mutation operator according to its current efficiency. Another schemes of mutation can be easily implemented, for instance self-adaptive mutations proposed by (Anglano et al., 1998; Thomsen, Krink, 2002).

Generational replacement. The generational replacement is an operation that determines which of the classifiers in the current population is to be replaced by newly discovered children. According to Algorithm 1, the new generation of rules is created from a population of parents (P) and their children after the crossover and the mutation operations (P”). In our system, the following replacement strategies are applied:

- the revolutionary strategy in which only the population of the children completely replaces the parent population (P),
- the steady-state strategy in which new children are inserted in the new population by replacing the worst, or the oldest rule, or the most similar rules, or by preserving the best rules (elitism).

There exist other replacement strategies integrating, for instance, the strategy where the best rule of the previous population replaces the worst one of the current population or the strategy where the new classifiers having a performance higher than a certain threshold are inserted. However, both these strategies present the risk of having classifiers remain in the population, which is not necessarily a problem except in the case of a weak genetic pool in which some classifiers of average performances that would profit from immunity.

4 Case studies and experiments

In this paper, two case studies involving the remote sensing images of Strasbourg and San Felice (Lagoon of Venice) have been chosen. These cases contain hyperspectral data (DAIS 79 bands and ROSIS 80 bands, respectively), with 16 bits per pixel and 3m terrain resolution (Wooding, 2001; Quirin, 2002). The first case study considers a typical problem of classification for urban zones including a high percentage of mixed pixels. The second case demonstrates the performance of rules on very noisy images with closed spectral classes (mostly vegetation classes). Learning was carried out on the lower half of the image (932*184 pixels), and then validation was performed on the whole image (932*368 pixels).

To well understand the formalism of rule representation, let B_i be the reflectance value for the band i of the considered pixel. For instance, the conditional portion of the rule that classifies the instances of a *Limonium Narbonense* class is given below:

$$
\begin{aligned}
& (0 \le B_0 \le 65535) \\
\wedge\ & (461 \le B_1 \le 1928) \\
\wedge\ & \ldots \\
\wedge\ & (0 \le B_5 \le 65535) \\
\wedge\ & \ldots \\
\wedge\ & ((522 \le B_{12} \le 1895) \vee (6541 \le B_{12} \le 39307)) \\
\wedge\ & \ldots \\
\wedge\ & (364 \le B_{79} \le 2107)
\end{aligned}
$$

It is easy to notice that the band B_0 and B_5 are too noisy (range maximum), and they can be eliminated from the condition. The rules simplification may be also implemented indirectly in the rule evaluation function, promoting the rule simplicity. It should be also noted that after preliminary tests, this method generated many over-generalized rules with relatively weaker performance than that obtained by the simple deletion.

In the following part, the main results of evolutionary data mining are described. Each case study is illustrated by a classified image using the discovered rules, discussions and performance measures.

Image of Strasbourg, Stadium Vauban (hyperspectral image, 80 bands)

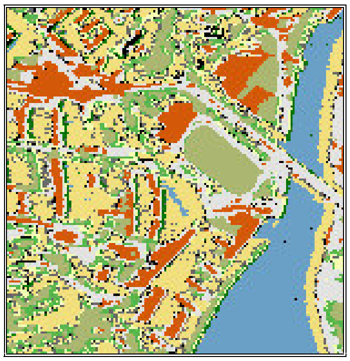

Fig. 2. Classified image

Table 1. Parameters of the GA

Parameter	Value
Population	1500 rules
Generations	250
Stabilization length	20
Stabilization error	10^{-4}
Crossover rate	80%
Mutation rate	5%
Rate of eugenic sel.	1%
CPU time (P4 2.5GHz)	16 h
Learned classes	11
Performance	86,06 %

Comments. This complex remote sensing image contains more than 50% noisy bands. Moreover, *Water* data is represented by many small, corrugated lines (signal has been scrambled by atmospheric conditions) so *Water* and *Shadow* spectral signals are very similar. Therefore, a small spot of *Water* appears in the middle of the city, instead of *Shadow*. The average quality of the best rule for each class is about 86%, which is relatively good performance.

Image of Venice, Lagoon San Felice (multispectral image, 4 bands)

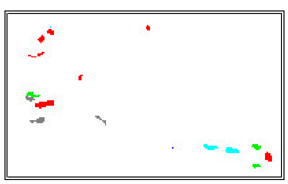

Fig. 3. Expert image

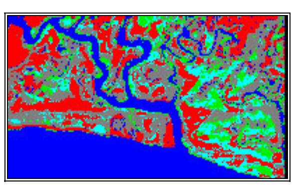

Fig. 4. Classified image

Table 2. Parameters of the GA

Parameter	Value
Population	500 rules
Generations	500
Stabilization length	10
Stabilization error	10^{-4}
Crossover rate	75%
Mutation rate	15%
Rate of eugenic sel.	1%
CPU time (P4 2.5GHz)	2 h
Learned classes	5
Performance	89,03 %

Comments. Fig. 3 presents a typical classified image by an expert using ground truth data. As illustrated the number of classified pixels (shown in color zones) is very low (1.43%). Note that in the whole image only 10 pixels were identified as the *Water* class. In spite of the small training set and the large space of search (spectral value of pixels is represented by 32 bits), the discovered set of rules is able to produce a coherent classified image.

Resistance to the noise of the learning process. The resistance of the classifier system to the noise has been evaluated numerically. Fig. 5 illustrates the protocol used for validation. ($Data_L$,$Expert_L$) are the training data and ($Data_T$,Exp_T) are the validation data. These sets are of the same size and are generated by taking for each one the half of the pixels of the whole image. *Noise(image,rate)* is a function which perturbs *rate%* pixels. *Valid()* is a function which generates the weighted performance of classifiers, according to the discovered rules (*Rules*) and an expert classification (*Expert*) of the remote sensing image (*Data*). *Cycle(rd,re)* computes the performance of the rules learned from a *rd%* perturbed data and a *re%* perturbed expert image. The curves below, C_{data}=*Cycle([0;50],0) and* C_{expert}=*Cycle(0,[0;50])* illustrate the weighted performance of the rules on the noisy training set.

```
Cycle(nd,ne)
  NoRules=50
  NoGenerations=50
  Sampling of data=10%
  Rules=Learning(
   Noise(Data_L , rd),
   Noise(Expert_L , re))
  Perf=Valid(Data_T,Expert_T,Rules)
  return Perf
EndCycle
```

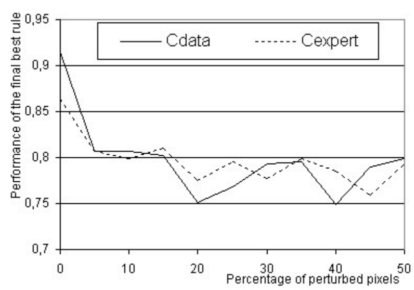

Fig. 5. The validation protocol and the resulting graph

We performed 11 runs for each test (perturbation of the data or the expert image). Standard deviations of the rule performances are : $\sigma_{C_{data}}$ = 0,042 and $\sigma_{C_{expert}}$ = 0,026. The two case studies have demonstrated the high capacity of the evolution-based rules to interpret and classify heterogeneous and complex images (e.g. high dimension, large number of bands and noisy data that provide a computational complexity of $O(n^3)$, which is quite heavy for a deterministic algorithm). The quality of classification is very high even if there were a high number of noisy bands and mixed pixels. It must be noted that the quality of learning is highly related to the quality of the classified image used for rule discovery. The discovered classification rules are simple and easy to interpret by remote sensing experts. They are also mutually exclusive and maximally specific. The learning time was relatively long due to the large image size and the chosen parameters for the evolution process, but the computing time optimization was not addressed in these experiments.

During the experiments it was observed that the best rules use 0% mutation of bands, 5% mutation of intervals, 41% mutation of borders, and 53% crossovers. In spite of weak mutation rate (5-15%), mutation operators have demonstrated high efficacy. The diagram shows that this evolutionary process is able to admit nearly 5%

of noise on the data or the expert image without significant loss of quality. We have observed that by adding more than 5% of noise, the rule quality does not clearly decrease. Rule generalization quality has also been evaluated and it is worthwhile to mention that the best set of rules on high-resolution images can be applied on a 23-times larger image with a loss of quality less than 0.02%.

Finally, high correlation was observed between obtained results and statistics carried out on the remote sensing image (spectrogram statistics, excluding noisy bands). Classified images by the discovered rules have shown that the evolution-based process is able to faithfully reproduce the human expertise.

5 Conclusions and perspectives

This article has described an evolution-based method applied to remote sensing images. The system has discovered a set of *if ... then* image classification rules using the fitness function based on class recognition quality. These rules, which were proven robust and simple to understand for the user, improve the accuracy of classifications proposed by the expert, and are sufficiently generic for reusing them on other portions of remote sensing images.

Taking into consideration image complexity and noisy data, the results of our experiments are very encouraging. Case studies have demonstrated that the obtained rules are able to reproduce faithfully the terrain reality.

The rules are well adapted to recognize large objects on the image (e.g. sport lands), as well as the smaller ones (e.g. trees, shadows, edges of the buildings). The redundant or noisy bands have been successfully identified by our rule representation. The formulation of rule representation has allowed for the modeling of a spectral tube adapted to the granularity of spectral reflectance. The proposed rules initialization seems to be well suited to large volume of data. It has considerably reduced the search space by generating initial rules close to the final solution.

The genetic system developed in this research work, called ICU, is currently available on our web site `http://lsiit.u-strasbg.fr/afd`. A new version of ICU is under development, including a more powerful representation of rules including spatial knowledge, temporal relations, hierarchical representation of objects, and new genetic operators.

Acknowledgements. The authors are grateful to Pierre Gançarski, Cédric Wemmert, Christiane Weber, Anne Puissant, Jélila Labed, Massimo Menenti and Jennifer Ayers for their advice, suggestions and comments during the early stages of this research. We also appreciated the help brought by the suggestions of four anonymous reviewers.

References

ANGLANO C., GIORDANA A., BELLO G. L., SAITTA L. (1998). An experimental evaluation of coevolutive concept learning, [in] *Proc. of ICML'98*, Morgan Kaufmann, San Francisco, pp. 19-27.

BLICKLE T., THIELE L. (1995). A Comparison of Selection Schemes used in Genetic Algorithms, Computer Engineering and Communication Networks Lab, *TIK-Report Nr. 11*, Second Edition, Swiss Federal Institute of Technology, Zurich.

BOCK H. H., DIDAY E. (eds.) (1999). Analysis of Symbolic Data. Exploratory Methods for Extracting Statistical Information from Complex Data, [in] *Studies in Classification, Data Analysis and Knowledge Organization*, vol. 15, Springer-Verlag, Heidelberg.

CARVALHO D. R., FREITAS A. A. (2002). A genetic algorithm with sequential niching for discovering small-disjunct rules, [in] *Proc. of GECCO-2002*, pp. 1035-1042.

DEJONG K. A. (1988). Learning with Genetic Algorithms: An Overview, *Machine Learning*, vol. 3, pp. 121-138.

FJØRTOFT R., MARTHON P., LOPES A., SERY F., DUCROT-GAMBART D., CUBERO-CASTAN E. (1996). Region-Based Enhancement and Analysis of SAR Images, [in] *Proc. of ICIP'96*, vol. 3, Lausanne, pp. 879-882.

KALLEL L., SCHOENAUER M. (1997), Alternative random initialization in genetic algorithms, [in] *Proc. of ICGA'97*, Morgan Kaufmann, San Fransisco, pp. 268-275.

KORCZAK J., LOUIS N. (1999). Synthesis of Conceptual Hierarchies Applied to Remote Sensing, [in] *Proc. of SPIE*, Image and Signal Processing for Remote Sensing IV, Barcelona, pp. 397-406.

KORCZAK J., QUIRIN A. (2003). Evolutionary Approach to Discovery of Classification Rules from Remote Sensing Images, [in] *Proc. of EvoIASP2003*, Essex.

KORCZAK J., QUIRIN A. (2003). Découverte de règles de classification par approche évolutive : application aux images de télédétection, [in] *Proc. of EGC2003*, Lyon.

KURITA T., OTSU N. (1993). Texture Classification by Higher Order Local Autocorrelation Features, [in] *Proc. of Asian Conf. on Computer Vision*, Osaka, pp. 175-178.

QUIRIN A. (2002). Découverte de règles de classification : classifieurs évolutifs, *Mémoire DEA d'Informatique*, Université Louis Pasteur, LSIIT UMR-7005 CNRS, Strasbourg.

RENDON M. V. (1997). Reinforcement Learning in the Fuzzy Classifier System, *Reporte de Investigaci No. CIA-RI-031*, ITESM, Campus Monterrey, Centro de Inteligencia Artificial.

RIOLO R. L. (1988). Empirical Studies of Default Hierarchies and Sequences of Rules in Learning Classifier Systems, PhD Dissertation, Comp. Sc. and Eng. Dept, Univ. of Michigan.

ROSS B. J., GUALTIERI A. G., FUETEN F., BUDKEWITSCH P. (2002). Hyperspectral Image Analysis Using Genetic Programming, [in] *Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, ed. W. B. LANGDON et al., CA: M. KAUFMANN, pp. 1196-1203.

THOMSEN R., KRINK T. (2002). Self-Adaptive Operator Scheduling using the Religion-Based EA, [in] *Proc. of Parallel Problem Solving from Nature VII (PPSN-2002)*, pp. 214-223.

WEBER C. (1995). Images satellitaires et milieu urbain, Hermès, Paris.

WILSON S. W. (1999). State of XCS Classifier System Research, [in] *Proc. of IWLCS-99*, Orlando.

WOODING M. (2001). *Proceedings of the Final Results Workshop on DAISEX (Digital AIrborne Spectrometer EXperiment)*, ESTEC, Noordwijk.