

Scenario Analysis for Image Classification using Multi-objective Optimization

ELIANA PANTALEÃO¹
LUCIANO VIEIRA DUTRA¹
SANDRA SANDRI²

INPE - Instituto Nacional de Pesquisas Espaciais
CAP - Curso de Pós-Graduação em Computação Aplicada
Av. dos Astronautas, 1758 12227-010 - São José dos Campos(SP)- Brazil

¹(elianap, dutra)@dpi.inpe.br

²sandri.at.lac.inpe.br@gmail.com

Abstract. In a typical image classification task, the analyst decides beforehand the number of classes and which image channels to use. If there is a need to modify the classes or data channels, it is necessary to start over. This paper proposes a scenario analysis tool for the task of image classification as a way of automating this process. Each scenario represents the parameters that will be used in a complete supervised classification task, including training and classification. The proposed method uses multi-objective optimization to evaluate different sets of attributes and classes, and presents the compromising solutions, regarding the user objectives. A class hierarchy structure is used to generate different class sets, and the system attempts to find the most appropriate combinations of class and attribute sets. In this work, the system is applied to remote sensing problems and we consider three objectives: the best classification accuracy, the smallest attribute set and the biggest class set. The system shows the compromising combinations of class and attribute sets, along with the accuracy on a testing sample. The user can then choose which combination to use for the image classification.

Keywords: image classification, multi-objective optimization, class hierarchy, remote sensing.

(Received September 24th, 2012 / Accepted January 24th, 2013)

1 Introduction

This paper proposes a scenario analyst intended to be used for remote sensing applications. Remote sensing is the process of collecting data about objects or landscape features without coming into direct physical contact with them. Remotely sensed data are usually stored as arrays of two-dimensional data referred to as images. Pattern recognition aims to associate categories or classes to objects represented by a set of measurements, named feature vector or pattern. In the task of image classification for remote sensing applications, the features usually correspond to the image itself, that is, the digital values of the image matrices. Derived information such as texture, band math and principal com-

ponent analysis can also be used [10], [13], [1].

In some cases, there are more available features than it is necessary to the classification task. The aim of feature selection is to choose the most important of them in order to reduce their number and at the same time retain as much of their class discriminatory information as possible [19]. In other words, no new feature is created, the ones that are considered irrelevant or redundant are discarded, and we ideally would end up with the best possible feature subset, the one with minimum size and which leads to a minimum classification error rate. In practice, we usually try to select a reduced subset of features that does not significantly decrease the classification accuracy [5]. The feature selection strategy has

a direct influence on the accuracy of pattern recognition applications [19].

The pattern recognition task of assigning classes to feature vectors can be supervised, semi-supervised or unsupervised. In supervised learning, the user gives the classifier a training set of labeled examples for each class, which are used to estimate the class model or boundaries between classes. After that, it is possible to assign each unknown pixel or region to the appropriate class. All classifications performed in this work use supervised learning.

Several attempts have been made to improve classification by automatically choosing the input elements. In [2], decision trees are used to find a compromising solution for class set specificity and classification accuracy. In [11], the authors use a genetic algorithm to select the features and, concurrently, find the best parameters for a support vector machine classifier. A multi-objective optimization technique is used by [17] to select features in data bases. The addition of new variables and objectives in these methods can not be done with little effort. In [2], only the classes are considered, and [17] varies only the feature subset. In [11], the classes are not considered. Besides, the objectives are combined in an equation with weights that reflect how important the user considers each one of the objectives, and these weights have to be set before the optimization process.

In the method proposed in this paper, we consider as part of the varying scenario a set of features and a set of classes, as in Figure 1. The idea is to generate different feature and class sets and evaluate the performance of these different scenarios in the image classification task.



Figure 1: Scenario elements

In order to decide what a good classification is, several types of objectives can be considered. The system is flexible to allow the addition of new objectives with little programming effort. In this implementation we considered three: the best classification accuracy, the smallest attribute set and the biggest class set, meaning more specific classes. Other examples could be the best generalization power, reduced classification time or increased uniformity inside some predefined size window.

2 Multi-objective optimization

A multi-objective optimization approach is necessary when there are at least two conflicting objectives aimed for a problem and there is no unique solution that is best for all objectives [9]. In this case, one possible tactic is to find the set of all non-dominant solutions, referred to as the Pareto set approach [20].

In the multi-objective optimization problem shown in the formulation (1), the goal is to find the best vector \vec{x} of N variables, to minimize M objective functions. The feasible searching space is limited by lower and upper bounds for each variable x_i , and also J inequality and K equality constraints [6].

$$\begin{aligned} &\text{Min} \\ &f_m(\vec{x}), \quad m \in \{1, 2, \dots, M\} \\ &\text{Subject to} \\ &g_j(\vec{x}) \geq 0, \quad j \in \{1, 2, \dots, J\} \\ &h_k(\vec{x}) = 0, \quad k \in \{1, 2, \dots, K\} \\ &x_i^L \leq x_i \leq x_i^U, \quad i \in \{1, 2, \dots, N\} \end{aligned} \quad (1)$$

Using the Pareto set approach, the aim is to find S non-dominant solutions \vec{x}_s , $S \geq 1$. Dominance is defined as the relation \preceq and a solution $x_{s1}^{\vec{}}$ is said to dominate another solution $x_{s2}^{\vec{}}$ if both conditions (2) and (3) are true.

$$(\forall m)(f_m(x_{s1}^{\vec{}}) \leq f_m(x_{s2}^{\vec{}})) \quad (2)$$

$$(\exists m)(f_m(x_{s1}^{\vec{}}) < f_m(x_{s2}^{\vec{}})) \quad (3)$$

Condition (2) states that there is no objective function where $x_{s1}^{\vec{}}$ is worse than $x_{s2}^{\vec{}}$ and condition (3) states that $x_{s1}^{\vec{}}$ is strictly better than $x_{s2}^{\vec{}}$ in at least one objective [3]. If there is at least one pair $(m1, m2)$ for which $f_{m1}(x_{s1}^{\vec{}}) < f_{m1}(x_{s2}^{\vec{}})$ and $f_{m2}(x_{s1}^{\vec{}}) > f_{m2}(x_{s2}^{\vec{}})$ then neither of the solutions dominates the other, because it is not possible to say that one solution is better than the other for both objectives. Since it is a minimization problem, $x_{s1}^{\vec{}}$ is a better solution for objective function f_{m1} , while $x_{s2}^{\vec{}}$ is better for f_{m2} . The Pareto set is a set of viable solutions where none of them dominates any of the others [6] and is formally defined in expression (4).

$$P^* = \{\vec{x}_1 \mid \nexists \vec{x}_2 : \vec{x}_2 \preceq \vec{x}_1\} \quad (4)$$

In the Pareto set, the elements are solutions, or values for the minimization problem variables. The set of the objective functions values for the Pareto set is called Pareto frontier and is given by expression (5).

$$PF^* = \{\vec{f}(\vec{x}) \mid \vec{x} \in P^*\} \quad (5)$$

The image classification task can be seen as a multi-objective problem, because choosing more specific classes usually results in lower accuracy, so it makes sense to find compromising solutions. Particularly in remote sensing applications, one typical use of classification is to derive a land cover or land use map [16]. How specific this map can be, given the available data, is a question frequently faced by human analysts. Choosing to discriminate only “agriculture” instead of “corn” and “sugarcane” usually requires less attributes, and leads to a more accurate map, with less confusion between classes. Descending in class hierarchy makes accuracy worse, but depending on how worse that is, it can be considered a better map, because it is more specific; and all this is affected by the number of features used.

When we try to minimize the number of features while maximizing the classification accuracy and descend as low as possible in the class taxonomy, we are doing multi-objective optimization. They are conflicting objectives because optimizing one of them degrades at least one of the others [19], [16]. There is no unique optimal solution, but a set of solutions representing a trade-off surface among several objectives. None of these solutions can be considered better than the others with respect to all goals simultaneously. It is the user, with some higher level information, who can choose the one to be adopted [6]. In the end of the process of this proposed method, a report with the evaluation of the scenarios which produced the trade-off results is generated. Therefore, the final report contains the description and the classification result analysis of the scenarios which have the best compromise of the three goals.

In this work, the implementation of the multi-objective optimization solution was based on the M_GEO algorithm [7], a multi-objective version of the Generalized Extremal Optimization (GEO) algorithm [8]. GEO is an evolutionary algorithm where the species are represented by a string of bits that encodes the variables of the optimization problem. Each bit is forced to “evolve” with a probability that is proportional to its fitness. The M_GEO algorithm is shown in Figure 2.

The only free parameter to adjust, the τ in Figure 2, step 11, controls the determinism level of the search, from completely random ($\tau = 0$) to a deterministic search ($\tau \rightarrow \infty$). For each specific problem, there is a value for τ that maximizes the search efficiency [7].

The species were implemented as a bit string representing two variables, the feature and the class set, according to the scenario definition presented in Section 3.

```

1: initialize randomly one  $L$ -bits string that codify  $N$ 
   design variables;
2: while stopping criteria not reached do
3:   for  $i \leftarrow 1$  to  $L$  do
4:     compute the value of bit  $i$ , generating a new
       string  $\vec{x}_i$ ;
5:     calculate the value of all objective functions,
        $\vec{f}(\vec{x}_i) = [f_1(\vec{x}_i), \dots, f_M(\vec{x}_i)]$ ;
6:     test and save Pareto frontier and Pareto set;
   end for
7:   choose randomly one objective function as  $f_{Ic}$ 
8:   foreach string  $\vec{x}_i$  do
9:     assign a fitness value proportional to the gain
       or loss that  $f_{Ic}(\vec{x}_i)$  has if the bit mutates,
       compared to the best  $f_{Ic}$  value found so far;
10:    rank the bits according to their fitness values,
        being 1 the position of the worst value;
11:    mutate one bit of the population with
        probability  $P_k \propto k^{-\tau}$ ,  $k \in \{1, 2, \dots, L\}$ ,
         $k$  is the position of the bit in the rank;
   end foreach
end while
12: return Pareto frontier and Pareto set;

```

Figure 2: M_GEO algorithm

3 Scenario Definition

In the context of this work, a scenario consists of a feature and a class set, as in Figure 1. These variables values are altered by the scenario generator. A classified image is generated when a classifier uses the scenario components to classify an image (Figure 3). The idea in this system is to evaluate different scenarios and find the compromising ones according to two or more objectives sought for the classification result.



Figure 3: Classification task using scenario

Formally, we define scenario as a pair $S = (\Phi, \Omega)$, where Φ is the attribute or feature set and Ω is the class set. When S is applied to image I , the classified image I_S is generated.

The proposed method supposes the existence of labeled samples and an initial class hierarchy (Figure 4), built or loaded by the system user. This class tree represents a vision of the scene, a rough description of what

the user expects the classifier to find in the image. The ideal is that the hierarchy is based on semantic information, not on attributes values or classes spectral behavior. However, if this information is not available, it is possible to generate the hierarchy automatically, using information from the available features [14].

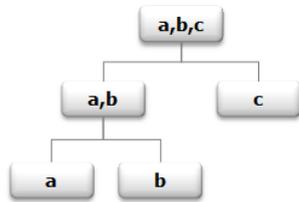


Figure 4: Example of a class tree

The labeled samples provide information about available features, allowing the scenario generator to build different attribute combinations, the different sets Φ . Any subset of the initial set of all attributes can be generated, except the empty set, since it is not possible to perform a classification with no features.

The generation of class sets Ω is guided by a class tree. Let Γ be the set of all the most specific classes, those that would be the class tree leaves. If no class tree was used, any partition of Γ could be used as a set Ω except the one where all classes are included in the same subset (the root of the class tree). If we used this partition, no classification would take place, since there would be only one class including all samples. But in this work, we use the class tree to exclude other semantically less meaningful partitions. For example, consider the class tree in Figure 4, built from $\Gamma = \{a, b, c\}$. There are five possible partitions for Γ : $\{\{a, b, c\}\}, \{\{a, b\}, \{c\}\}, \{\{a, c\}, \{b\}\}, \{\{a\}, \{b, c\}\}$ and $\{\{a\}, \{b\}, \{c\}\}$. The first one is the set of all classes, therefore cannot be used as Ω . And only $\{\{a, b\}, \{c\}\}$ and $\{\{a\}, \{b\}, \{c\}\}$ are allowed by the class tree in Figure 4. It is not possible to merge classes b and c in the same class, leaving a alone, because this is not how the tree is structured.

4 Scenario Analyst

The scenario analyst uses a multi-objective optimization algorithm to find compromising scenarios, according to the intended objectives. Figure 5 shows the methodology steps implemented by the system.

The implementation of the scenario generation followed the model stated in Section 3. The string that is used to represent these design variables is a L -bit string, where $L = F + C$, F is the number of available features

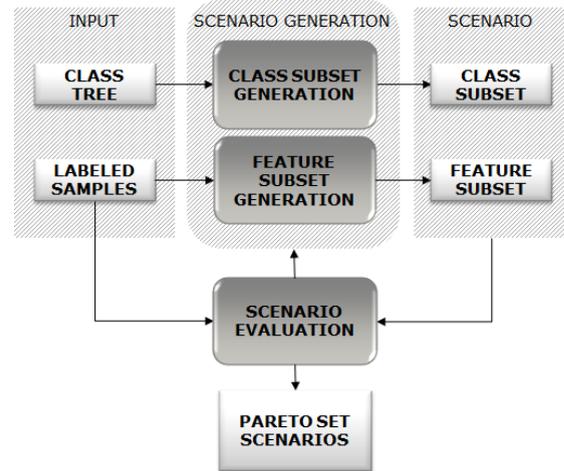


Figure 5: System architecture

and C is the total number of all classes considered, including all leaves and internal nodes of the class tree, except the root.

The only attribute set constraint is to be non-empty, so, if the scenario generator is altering the value of the first F string bits and an all-zero F -bits substring is generated, it is ignored, and another one is produced. Any other combination is allowed.

As for the last C bits, the constraints are more complex, because they consider the class hierarchy that guides the class set generation. There are two constraints that must be true for all classes in the hierarchy:

- If a class is included in a class set, then none of its ascendants or descendants can be in the set at the same time.
- If a class is not included in a class set, then either one of its ascendants or descendants must be in the set.

With these two constraints, the class set is guaranteed to follow the proposed formal model. In practice, it guarantees that no class is ever removed from the scenario; it either joined its siblings forming a superclass or was divided into subclasses.

After the reference data input, M_GEO algorithm is used to control the loop where different scenarios are generated and evaluated. The labeled samples are divided into three subsets: half of the samples are used for training the classifier, 1/4 for the evaluation of the objective function on step 5 of M_GEO (Figure 2) and the remaining 1/4 is used to evaluate the final classification. Differently from step 1 of the original M_GEO,

we initialize the system with the complete scenario, the one where all features are used, and the most specific classes are discriminated.

5 Case Study

As a system application, the land cover was extracted from a Landsat-5 image [1]. The study area is located in the Amazon region of Brazil, in Para state. The optical image is from sensor TM (Thematic Mapper) and was acquired on June, 29th, 2010. We use 6 bands: B1-B5 and B7. The image has 30 meters of spatial resolution, 8 bits of radiometric resolution and 1672000 pixels, structured in 1900 lines and 880 columns.

The original image can be obtained from INPE Image Catalog [12]. The field samples were assigned to a previously segmented image, and only a spatial subset was used, in order to fit other data source. The complete processing of the input data is described in [15]. The channels B3 and B4 from the whole used image can be seen in Figure 6 and a more detailed view can be observed in the subset shown in Figure 7.

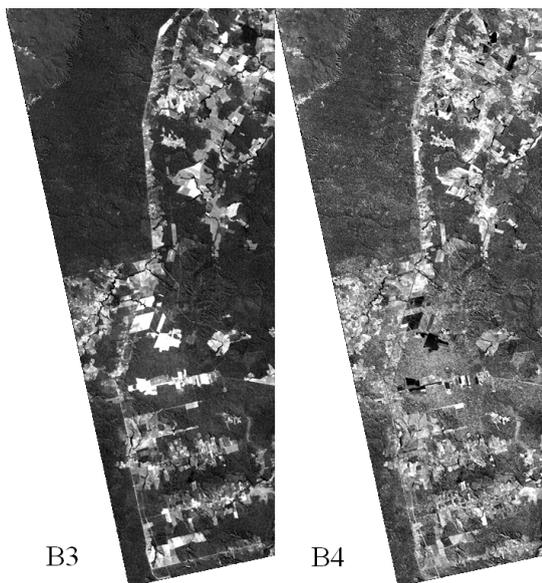


Figure 6: Bands 3 and 4 from TM image

The classes to be discriminated originated from several field works performed in 2010 and include some types of forests, agriculture and pasture. A class tree was built by a remote sensing specialist and can be seen in Figure 8. Table 1 shows the meaning of the classes and the size of the available samples in pixels.

Three objective functions were considered: the accuracy of the classification and both sizes of the class

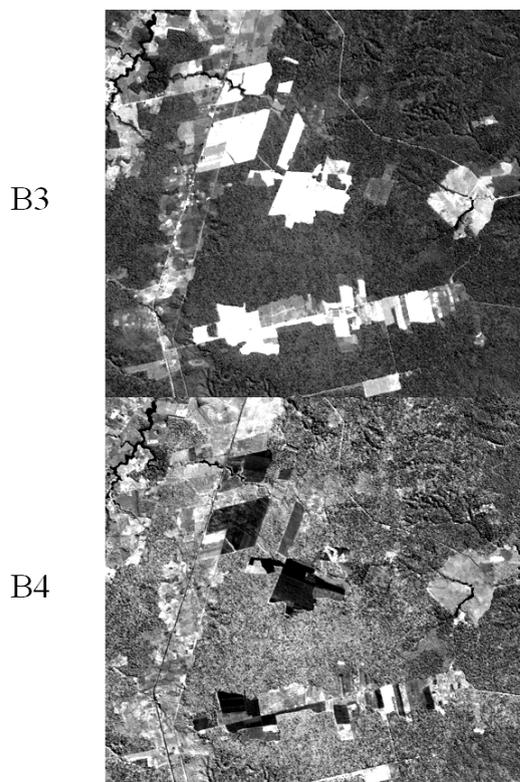


Figure 7: Detail of bands 3 and 4 from TM image

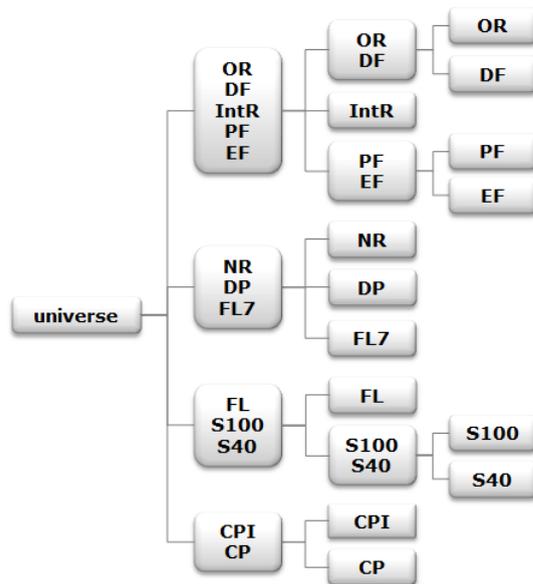


Figure 8: Class tree for the application

Table 1: Meaning of the classes from the class tree

<i>class</i>	<i>meaning</i>	<i>sample size</i>
OR	old regeneration	2518
DF	degraded forest	2306
IntR	intermediate regeneration	453
PF	primary forest	7035
EF	exploitation forest	3635
NR	new regeneration	794
DP	dirty pasture	828
FL7	fallow land 7-24 months	269
FL	fallow land	1306
S40	soybean 40 days	134
S100	soybean 100 days	581
CPI	clean pasture and inaja	753
CP	clean pasture	906

and the feature set. The Maximum Likelihood (ML) classifier was used to perform the pixel wise, supervised classifications.

Several ways to measure accuracy are available, but there is no agreement about which one is the best. Most quantitative methods use the confusion matrix [18] derived from classification data and reference data. The kappa coefficient, presented by Cohen [4], is a widely used measure in remote sensing applications.

In this application, we use kappa coefficient (κ) to assess the agreement between the result of the classification and the reference data (the labeled samples). Our three objective functions are shown in the set of equations (6).

$$\begin{aligned}
 f_1 &= -|\Omega| \\
 f_2 &= |\Phi| \\
 f_3 &= -\kappa
 \end{aligned} \tag{6}$$

The negative signs in f_1 and f_3 are needed because we solve it as a minimization problem, and we want to maximize these values. We have only one restriction: the value of κ must be at least 0.5 so the classification can be considered satisfactory.

The resulting Pareto set and frontier can be observed in Table 2. The elements of the Ω_i sets, i from 1 to 6, are in Table 3. The size of the Ω sets was included in Table 2 for emphasis. The notation *PF.EF* means that the samples from the two classes *PF* and *EF* were all relabeled to one single class, forming their parent class shown in the class tree.

Six different class sets were included by the system in the Pareto set, and five different feature sets. The best feature set, that is, the smallest one, is 1, 2, 5, 6, which suggests these bands are the most powerful to discriminate the user classes in Ω_3 . The scenario with all classes and all features has a low kappa, 0.5295, and removing one feature (thus improving f_2) does not make it better.

A much better kappa is obtained when the user is satisfied with discriminating only the top four classes from the class tree (Ω_6).

Table 2: Selected scenarios

<i>bands</i>	<i>classes</i>	$ \Phi $	$ \Omega $	κ
1,2,3,4,5,6	Ω_1	6	13	0.5295
1,3,4,5,6	Ω_1	5	13	0.5181
1,2,3,5,6	Ω_1	5	13	0.5181
1,2,3,4,6	Ω_1	5	13	0.5181
1,2,3,4,5,6	Ω_2	6	12	0.5694
1,2,5,6	Ω_3	4	12	0.5249
1,2,3,4,5,6	Ω_6	6	4	0.7986
1,2,4,5	Ω_6	4	4	0.7957
1,2,3,4,5,6	Ω_4	6	10	0.5865
1,2,4,5,6	Ω_5	5	7	0.6401

Table 3: Ω elements

<i>set</i>	<i>elements</i>
Ω_1	AP,AP7,RA,FP,PL,RI,S40,S100,FD,FPE,PLI,RInt,PS
Ω_2	AP,AP7,RA,PL,RI,S40,S100,FD,FP,FPE,PLI,RInt,PS
Ω_3	AP,AP7,RA,FD,PL,RI,S40,S100,FP,FPE,PLI,RInt,PS
Ω_4	RA,FD,RInt,FP,FPE,RI,PS,AP7,S40,S100,AP,PL,PLI
Ω_5	RA,FD,RInt,FP,FPE,RI,PS,AP7,S40,S100,AP,PL,PLI
Ω_6	RA,RInt,FD,FP,FPE,RI,PS,AP7,S40,S100,AP,PL,PLI

The image was classified using the scenarios of Table 2. Figure 9 shows the detailed view of three classified images. In order to obtain the classification in Figure 9a, all 13 classes were used, and all 6 image bands. The accuracy for this classification is $\kappa = 0.5295$ and the classified image has a lot of isolated pixels, suggesting great confusion between classes. Figure 9b was generated with the class set Ω_5 , with 7 classes, and 5 bands (1,2,4,5,6) were used. Even with fewer bands, the accuracy of this classification was better ($\kappa = 0.6401$), which can be explained by the use of fewer classes. Figure 9c used class set Ω_6 (4 classes) and only 4 bands, obtaining $\kappa = 0.7957$. This last classified image shows a better definition of the discriminated classes, with less isolated pixels.

The graph in Figure 10 shows the variation of the three objectives for the eight different solutions in the Pareto set. The optimal solutions of each function are marked. The values of *kappa* were multiplied by 10 so the variation could be placed in the same range as the other objective functions.

As can be observed in the graph, the optimal points are never together in the same solution. After examining the analysis result, the user can choose the one that will best fit his needs. If the expectations are not reached with a low kappa, a higher scale map will be derived, with less specific classes. If there is need of more specific classes, the available features are not enough to

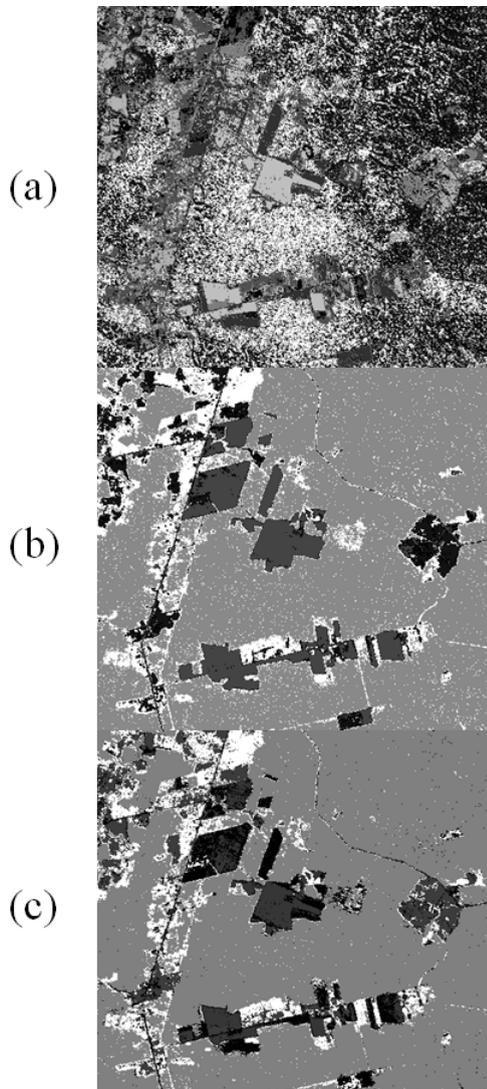


Figure 9: Detail of classified images for scenarios with (a) 13 classes, 6 bands; (b) 7 classes, 5 bands; and (c) 4 classes, 4 bands

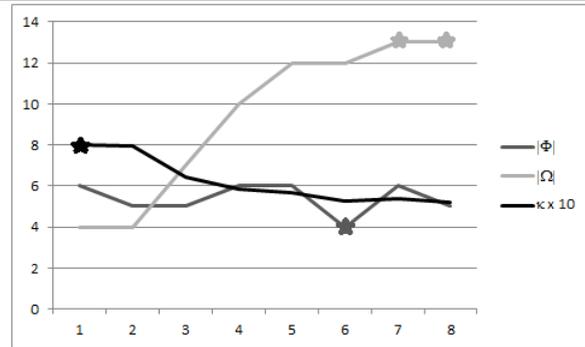


Figure 10: Variation of the objective functions

give a good kappa.

6 Comments and Future Work

The image classification task can be improved by a choice of the best classification scenario. An automated analyst makes it easier to consider the possible options for generating different scenarios. The multi-objective optimization proved to be a good method to solve this issue. Finding the best Pareto set of compromising scenarios will help the user to consider the best possibilities. The final choice of the best result will depend on the user application, resources availability and model restrictions. Although this method was created to be used in remote sensing applications, it may also be applied to other pattern recognition problems.

The flexibility aimed for the system was reached with respect to the definition of new objectives to be optimized and the selection of the project variables among the ones already coded. The addition of new variables, however, requires more computational effort. New data structures and means to generate their values automatically need to be developed.

As future steps, we will add more variables to the scenario, such as the type of classifier and its parameters, and include other objective functions, like the generalization power. We also consider using a priority value to the classes, thus reducing the possibility of the system eliminate the ones that are more important to the user.

Acknowledgements

The authors are grateful to CAPES for the funding of this research.

References

- [1] Bishop, C. M. *Pattern recognition and machine learning*. Information Science and Statistics. Springer, New York, NY, 2006.
- [2] Chen, Y.-L., Hu, H.-W., and Tang, K. Constructing a decision tree from data with hierarchical class labels. *Expert Systems with Applications*, (36):4838–4847, 2009.
- [3] Coello, C. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems: An International Journal*, 1(3):269–308, 1999.
- [4] Cohen, J. A coefficient of agreement of nominal scales. *Educational and Psychological Measurement*, (20):37–46, 1960.
- [5] Dash, M. and Liu, H. Feature selection for classification. *Intelligent Data Analysis*, 1(3):131–156, 1997. Cited By (since 1996): 255.
- [6] Deb, K. *Multi-objective optimization using evolutionary algorithms*. John Wiley and Sons, Chichester, 1 edition, 2001.
- [7] Galski, R. L. *Desenvolvimento de versões aprimoradas híbridas, paralela e multiobjetivo do método da otimização extrema generalizada e sua aplicação no projeto de sistemas espaciais*. PhD thesis, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2006-09-22 2006.
- [8] Galski, R. L., Souza, F. L. d., Ramos, F. M., and Muraoka, I. Application of a new hybrid evolutionary strategy to spacecraft thermal design. In *Anales... Genetic and Evolutionary Computational Conference (GECCO)*, 2004.
- [9] Goicoechea, A., Hansen, D., and Duckstein, L. *Multiobjective decision analysis with engineering and business applications*. John Wiley and Sons, New York, 1 edition, 1982.
- [10] Gonzalez, R. C. and Woods, R. E. *Digital image processing*. Prentice Hall, Upper Saddle River, NJ, 2 edition, 2001.
- [11] Huang, C.-L. and Wang, C.-J. A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications*, 31(2):231–240, 2006.
- [12] INPE. *Image Catalog*. National Institute for Space Research, 2004. Brazil, <http://www.dgi.inpe.br/CDSR/>.
- [13] Jensen, J. R. *Remote sensing of the environment: an Earth resource perspective*. Prentice Hall, Upper Saddle River, NJ, 2 edition, 2007.
- [14] Negri, R. G. Avaliação de dados polarimétricos do sensor alopsar para classificação da cobertura da terra da amazônia. Master's thesis, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2009-05-19 2009.
- [15] Pereira, L. d. O. Avaliação de métodos de integração de imagens ópticas e de radar para a classificação do uso e cobertura da terra na região amazônica. Master's thesis, Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2012-08-27 2012.
- [16] Schowengerdt, R. A. *Remote sensing: models and methods for image processing*. Academic Press, San Diego, 3 edition, 2007.
- [17] Spolaôr, N. *Aplicação de algoritmos genéticos multiobjetivo ao problema de seleção de atributos*. Dissertação (mestrado em engenharia de informática), UFABC, Santo André, 2010.
- [18] Story, M. and Congalton, R. G. Accuracy assessment: A user's perspective. *Photogrammetric Engineering and Remote Sensing*, 52:397–399, 1986.
- [19] Theodoridis, S. and Koutroumbas, K. *Pattern recognition*. Academic Press, San Diego, 3 edition, 2006.
- [20] Van Veldhuizen, A. and Lamont, G. B. Multiobjective evolutionary algorithms: analyzing the state-of-the-art. *Evolutionary Computation*, 8(2):125–147, 2000.