Image classification using bagging and decision tree in SACI system

Carlos F. S. Volotão[1], Guaraci J. Erthal[2], Luciano V. Dutra[3], Rafael D. C. Santos[4]

Instituto Nacional de Pesquisas Espaciais – INPE, Caixa Postal 515 - 12245-970 - São José dos Campos - SP, Brasil

{volotao,gaia,dutra}@dpi.inpe.br, rafael@lac.inpe.br

Abstract. *This paper presents the availability of two new features in SACI system: bagging and a new decision tree algorithm, the user options and results.*

1. Introduction

From the classification perspective, the goal of effective mining from data streams is to achieve the best possible classification performance for the task at hand (Zhu et al., 2004). Decision trees (to be referred to as "trees" throughout) in this paper are used to classify remote sensing images and in this case the term "classification trees" is often used. Trees work by sorting instances down a branch of the tree, from root to leaf, each node on the way questioning a specific attribute of that instance. Each branch leaving each node represents a value of the attribute that was tested at that node (Rogers, 2003). Usually the trees are represented upside down, with the root at the top and the leaves at the bottom.

Basically the goal of the procedure being presented is to classify a multi-band satellite image in a number of classes defined in regions of interest (ROIs). Trees are widely used algorithms and therefore there is a big number of literature concerning to this, which is an advantage in the implementation of this classifier.

The over-fitting of the training samples may occur if both the stop criteria and the pruning are not effective to "generalize" the tree. To enhance the classification performance the bagging is an alternative. In this case we construct several trees by randomly sorting the training samples to obtain different subsamples called bootstraps. Each tree provide a classification result and the final result is the class frequency majority chosen by vote. This makes the final classification more generic and can enhance the overall accuracy. The fact behind the merit of bagging is from the following underlying assumption: Each participating classifier has a particular subdomain from which it is most reliable, especially when different classifiers are built using different subsets of features and different subsets of the data.

The Image Processing Department (DPI) from the Brazilian National Institute for Space Research (INPE) developed an academical system for research purposes called "Sistema de Análise e Classificação de Imagens" (SACI) by the effort of researchers and students. It is able to aggregate independent implementation modules (Rosa et al., 2005). This environment provides the basic operations for choosing the files, defining the regions of interest and displaying the results by charts and images. The old version was called SCID and this was renamed to SACI.

2. Usage description

Building a decision tree from a training set consists of two phases. The initial decision tree derived in the construction phase may not be the best generalization due to over-fitting.

So the bagging and the stopping criteria avoid nodes from being created but the parameters offered are able to improve the accuracy, but can reduce the accuracy if "wrong" values are given. This means the algorithm is sensible to some parameters. They need to be tested and used because the algorithm still lacks the post-pruning phase

The bagging used with the trees was written in a way to combine not only several trees but multiple classifier systems such as the ones reported by Rogers (2003).

Among many partition merit criteria it was selected the most common way to judge the goodness of a split: entropy or impurity measure. In fact it was implemented two of the more often used: "gini index" split gain and "entropy split gain", as they have been proved accurate. Breiman et al. (1984), Quinlan (1986), Mingers (1989), Safavian and Landgrebe (1991), Buntine and Niblett (1992), and Fayyad and Irani (1992) discuss and compare different partition merit criteria. (Brodley and Utgoff, 1992)

In the construction phase the data is scanned multiple times and this can make the process slow. The bagging is responsible to make several trees and one of the parameters is the number "B" of bootstraps. If B is, say, 20 this means the classification will be theoretically about 20 times slower than with B=1 or without bagging (B=0).

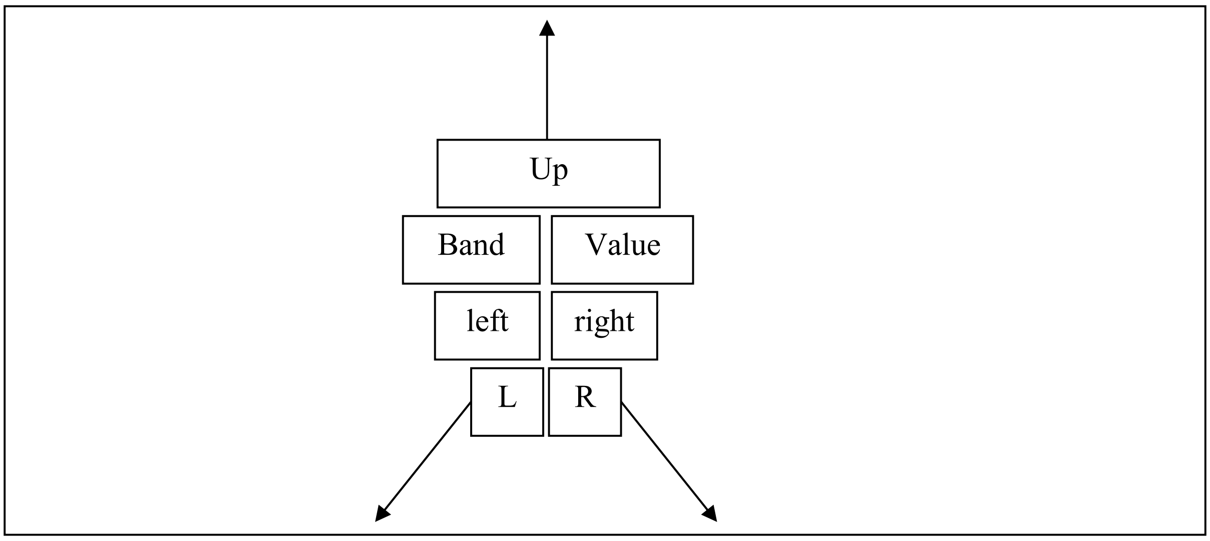
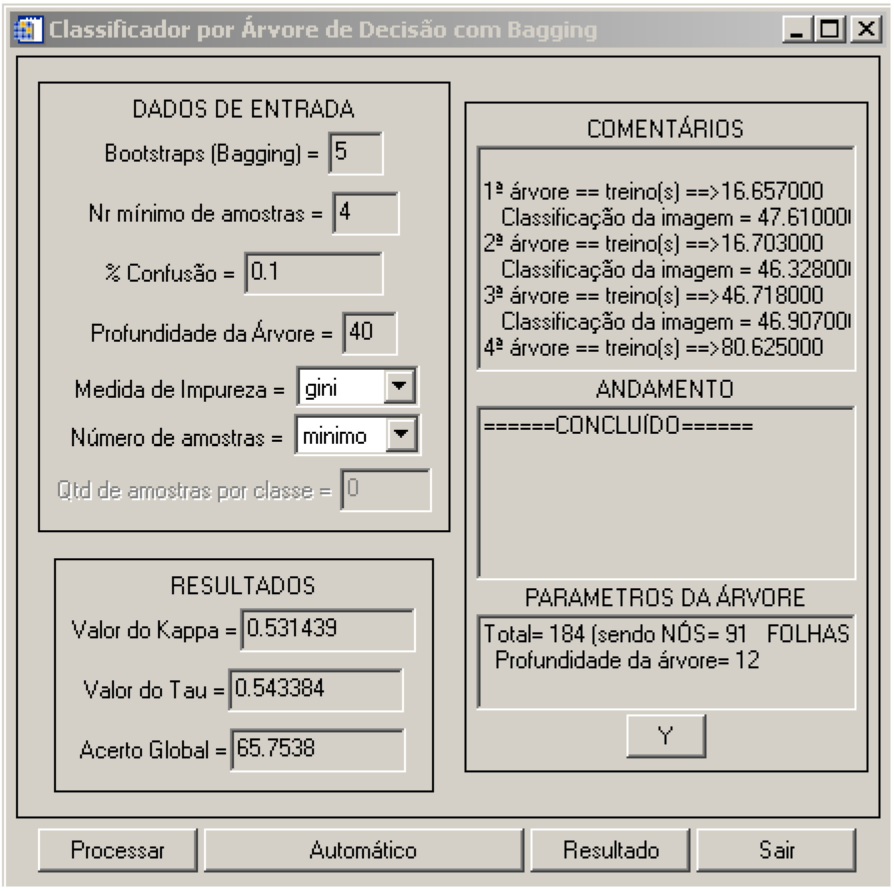


Figure 1. The internal structure of the decision tree: three pointers (up, L, R), what Band, what value, if it is a leaf then use (left, right) instead of the pointers (L,R).

Following the order presented in Figure 1, the parameters present in the main window are: the number of bootstraps (B); the minimum number of pixels used to define a leaf; the class purity tolerance; the tree depth limit; the impurity measure method; and the number of pixels per class used.

The impurity method can be chosen as 'gini' or 'entropy'. Some other methods may be available in the future to allow comparison of the tree layout.

The number of pixels in each bootstrap is flexible and can be chosen as: ·maintaining each ROI original size in number of pixels; ·sorting without making any restrictions in the class sizes; chosen from the minimum, the mean or the maximum class sample size; or defining the exact class size in pixels (the last input box then becomes available).



Bootstraps (Bagging) = 5
Nr mínimo de amostras = 4
% Confusão = 0.1
Profundidade da Árvore = 40
Medida de Impureza = gini
Número de amostras = minimo
Qtd de amostras por classe = 0

Valor do Kappa = 0.531439
Valor do Tau = 0.543384
Acerto Global = 65.7538

```
1ª árvore == treino(s) ==>16.657000
  Classificação da imagem = 47.61000
2ª árvore == treino(s) ==>16.703000
  Classificação da imagem = 46.32800
3ª árvore == treino(s) ==>46.718000
  Classificação da imagem = 46.90700
4ª árvore == treino(s) ==>80.625000
```

```
======CONCLUÍDO======
```

```
Total= 184 (sendo NÓS= 91   FOLHAS
 Profundidade da árvore= 12
```

Figure 2. Decision Tree Classifier with Bagging main window.

The main processing window have 5 buttons that can be translated as: "Y", "Process", "Automatic", "Result" and "Exit". The first is the Y flip button to rotate all display images. The "Processar" is the process button. The "Automatico" allows several different parameters combination, by changing the IDL font file without assistance.

3. Results

The Table 1 shows several test results for the same "Lena" test image and the same region of interest (ROI) having 5 classes. The following results are concerning these data sets. Figure 1 shows the classification using 3 bootstraps, 15 minimum samples per leaf, 30% tolerance about purity analysis, no cut in depth by setting depth to 100, "gini index" impurity method and a bootstrap size of 300 randomly chosen pixels.

The table results show that the bagging effect in the tree was about 2%. The best case was achieved by the tree without bagging that stopped at the depth of 40 and with 1248 nodes. For only 5 classes this means that the tree probably overfitted the training set to reduce the error but the classified image was much more heterogeneous than the 16 nodes tree in the 26th instance presented.

TABLE 1. Tests results using different parameters in 27 instances. The parameters are: item#, #bootstraps, minimum number of pixels, impurity tolerance, maximum depth, impurity method, pixels per ROI class, number of nodes achieved, depth of the tree, kappa, tau and global match.

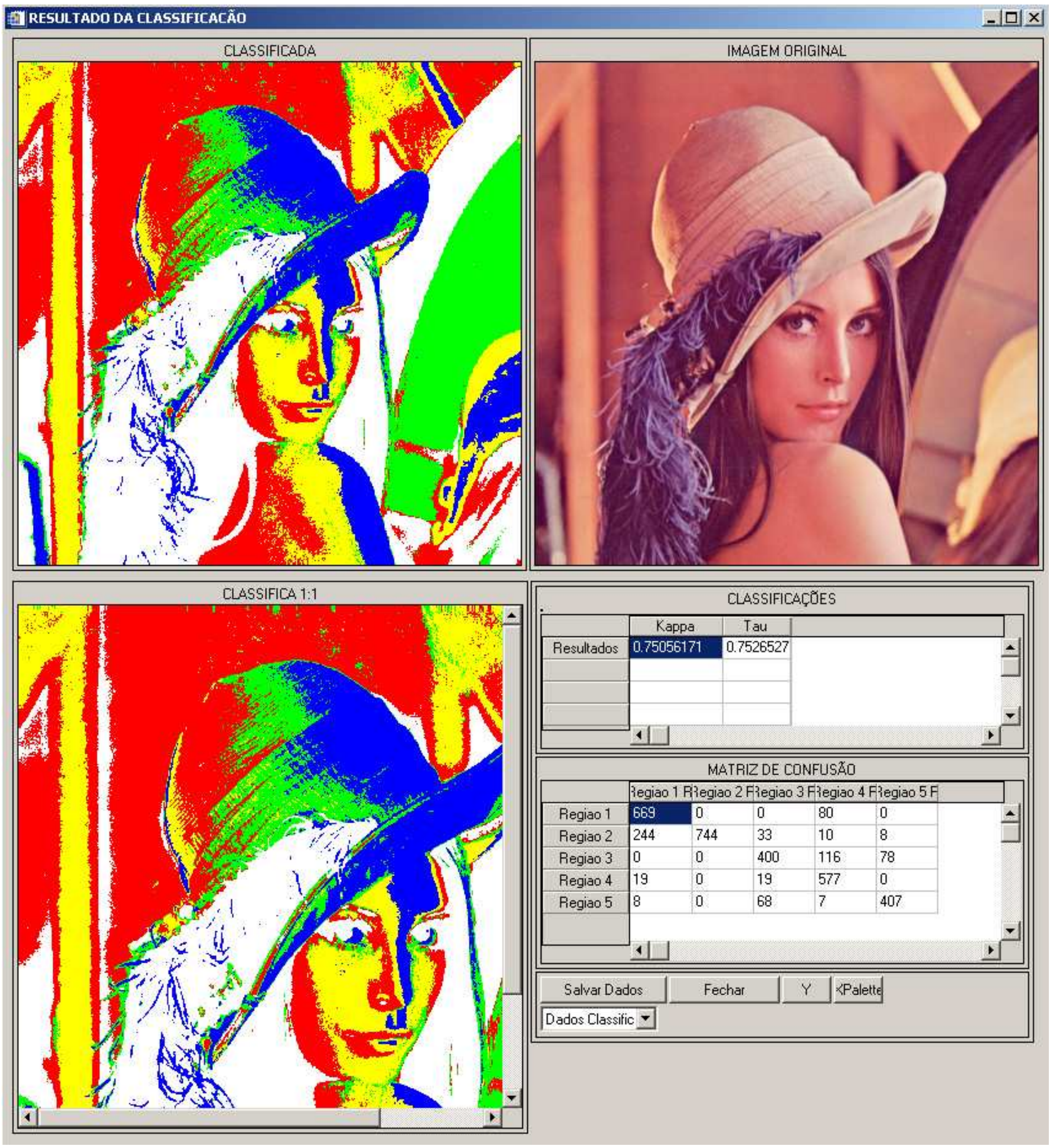
item	B	mpix	impurity	maxdepth	method	ppc	nodes	depth	kappa	tau	global
1	0	0	-	15	gini	original	820	16	0,923	0,925	94,0%
2	0	0	-	50	gini	original	1248	40	0,927	0,928	94,3%
3	0	0	-	50	entropia	original	1096	51	0,925	0,926	94,1%
4	1	0	-	50	gini	min	800	47	0,901	0,902	92,2%
5	10	0	-	50	gini	min	844	51	0,921	0,922	93,7%
6	1	0	-	50	gini	original	1168	51	0,903	0,904	92,3%
7	1	0	-	50	gini	aleatorio	1010	29	0,908	0,910	92,8%
8	1	0	-	50	gini	med	1056	51	0,912	0,913	93,1%
9	1	0	-	50	gini	max	1556	51	0,921	0,922	93,7%
10	1	0	-	50	gini	max	1450	51	0,915	0,916	93,3%
11	1	0	-	100	gini	max	1428	49	0,916	0,917	93,3%
12	1	2	-	100	gini	max	418	17	0,900	0,901	92,1%
13	1	3	-	100	gini	max	348	15	0,900	0,901	92,1%
14	1	0	0,05	100	gini	max	1076	52	0,903	0,904	92,3%
15	1	0	0,10	100	gini	max	420	43	0,886	0,888	91,0%
16	1	0	0,15	100	gini	max	310	37	0,874	0,876	90,1%
17	1	3	0,15	100	gini	max	146	14	0,861	0,863	89,0%
18	1	3	0,15	100	gini	min	58	12	0,858	0,860	88,8%
19	20	3	0,15	100	gini	min	78	10	0,873	0,875	90,0%
20	10	3	0,15	100	gini	200	48	8	0,868	0,870	89,6%
21	10	6	0,15	100	gini	200	40	9	0,860	0,862	89,0%
22	5	6	0,15	100	gini	200	38	9	0,863	0,865	89,2%
23	3	6	0,15	100	gini	200	34	8	0,857	0,858	88,7%
24	3	10	0,15	100	gini	200	34	7	0,843	0,845	87,6%
25	3	6	0,30	100	gini	200	20	8	0,732	0,735	78,8%
26	3	10	0,30	100	gini	200	16	6	0,747	0,749	79,9%
27	3	15	0,30	100	gini	300	15	5	0,751	0,753	80,2%

The best visual classification was achieved by the last instances although with the worse numeric results. The image results corresponding to the cases above 90% are very homogeneous and much useful than the best kappa of 0,93, which resulted in more "granulated" classes.

The tests pointed the best time achieved by the last instance, 25 seconds to train, classify and display the results, but about 5 minutes in the worst case.

4. Future Work

A pruning algorithm may be implemented to correct the over-fitting by reducing the number of nodes. This may introduce an additional generalization error as reduces the badly overfit. Helmbold and Schapire gave an efficient algorithm for post-pruning but it requires a different training data set. In the other hand, Kearns and Mansour (1998) present an efficient bottom-up algorithm for tree pruning that requires only a single pass through the tree and use the same data set used to construct the tree. They prove a strong performance guarantee for the generalization error of the resulting pruned tree. In other works these features can be used to enhance the robustness of the tree classifier. Alternatively the pruning algorithm presented by Rastogi and Shim (1998) can be applied during the generation phase of the decision tree.



CLASSIFICAÇÕES

	Kappa	Tau
Resultados	0.75056171	0.7526527

MATRIZ DE CONFUSÃO

	Regiao 1	Regiao 2	Regiao 3	Regiao 4	Regiao 5
Regiao 1	669	0	0	80	0
Regiao 2	244	744	33	10	8
Regiao 3	0	0	400	116	78
Regiao 4	19	0	19	577	0
Regiao 5	8	0	68	7	407

Figure 3. The result is shown in this layout, with the classification at the side of the original image and with a 1:1 zoomed image and the confusion matrix at the bottom.

Two other methods to avoid over-fitting are the addition of noise in the training data and the method described by Ho (1998). Both can easily integrate this work in the future but is not yet implemented. Ho (1998) don´t uses all the features but subsets of the features, which are the bands in image case, should be randomly selected in order to train trees. With several trees the data is classified and the final result presents the majority voting pixel-by-pixel.

5. Conclusion

This work presents the bagging and decision tree implementation in IDL, with several full controllable variables. The goal was met and was to provide SCID with tree classificatory method with bagging.

The parameters are useful to reduce tree complexity and can make huge generalization effect increasing the classification quality although reducing slightly the accuracy indexes.

The software is working and can be easily enhanced and updated as it was written in IDL language and is part of an academic image processing classification system.

7. References

Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulic and Renault 1991]; or dates in parentheses, e.g. Knuth (1984), Smith and Jones (1999).

The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.

References

Breiman, L.; Friedman, J. H.; Olshen, R. A.; Stone, C. J. Classification and regression trees. Belmont, CA: Wadsworth International Group. 1984

Brodley, C. E.; Utgoff, P. E. Multivariate decision trees. COINS Technical Report 92-82, University of Massachussets, Dec, 1992. (online: 04-mar-2008) http://www.cs.iastate.edu/~honavar/brodley-dtree.pdf

Buntine, W.; Niblett, T. A further comparison of splitting rules for decision-tree induction. In: Machine Learning, 8, pp.75-85. 1992.

Fayyad, U. M.; Irani, K. B. The attribute selection problem in decision tree generation. In: Proceedings of the Tenth National Conference on Artificial Intelligence, pp. 104-110. San Jose, CA: MIT Press, 1992.

Helmbold, D. P.; Schapire, R. E. Predicting nearly as well as the best pruning of a decision tree. In: Proceedings of the Eighth Annual Conference on Computational Learning Theory, ACM Press, pages 61-68, 1995.

Ho, T. K. The random subspace method for constructiong decision forests. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, 20, 8-aug-1998, pp.832-844.

Kearns, M.; Mansour, Y. A fast, bottom-up decision tree pruning algorithm with near-optimal generalization. In: Proc. 15th International Conf. On Machine Learning, 1998. (online: 04-mar-2008) http://citeseer.ist.psu.edu/82070.html

Mingers, J. An empirical comparison of selection measures for decision tree induction. In: Machine Learning, 3, pp. 319-342. 1989.

Quinlan, J. R. Induction of decision trees. In: Machine Learning, 1, pp.81-106. 1986.

Rastogi, R.; Shim, K. PUBLIC: A Decision Tree Classifier that Integrates Building and Pruning. In: Proceeding of the 24 th VLDB Conference, New York, 1998.

Rogers, S. Non linear transformations for the creation of diverse classifier ensembles. Dissertation Project, University of Sheffield, England, 2003. (Online: 04-mar-2008) http://www.dcs.shef.ac.uk/intranet/teaching/projects/archive/ug2003/pdf/u8sr.pdf

Rosa, G.; Dutra, L. V.; Freitas, C. C. Extração de atributos via modelos Arma 1D/2D, avaliação estatística da distância JM e classificador condicional contextual. In: Anais do V Worcap, INPE, São José dos Campos, 2005.

Safavian, S. R.; Langrebe, D. A survey of decisión tree classifier methodology. In: IEEE Transactions on Systems, Man and Cybernetics, 21, pp.660-674. 1991.

Zhu, X.; Wu, X.; Yang, Y. Effective classification of noisy data streams with attribute-oriented dynamic classifier selection. In: Proceedings of the 4th IEEE International Conference on Data Mining, pp. 305-312, Brighton, UK, 2004.