Multilayer Perceptron Classifier Combination for Identification of Materials on Noisy Soil Science Multispectral Images

Fabricio A. Breve[1] Moacir P. Ponti-Junior Nelson D. A. Mascarenhas
Computer Department – UFSCar – Federal University of São Carlos – São Carlos-SP, Brazil
{fabricio,moacir,nelson}@dc.ufscar.br

Abstract

Classifier combination experiments using the Multilayer Perceptron (MLP) were carried out using noisy soil science multispectral images, which were obtained using a Tomograph scanner. Using few units in the MLP hidden layer, images were classified using a single classifier. Later we used classifier combining techniques as Bagging, Decision Templates (DT) and Dempster-Shafer (DS), in order to improve the performance of the single classifiers and also stabilize the performance of the Multilayer Perceptron. The classification results were evaluated using Cross-Validation. The results showed stabilization of Multilayer Perceptron and improved results were achieved with fewer units in the MLP hidden layer.

1. Introduction

There are many techniques for combining multiple classifiers. They appeared on literature mainly in the past 20 years. The idea behind combiners is that different individual classifiers can offer complementary information about the objects to be classified. Instead of using just one classifier, a safer option would be to use many classifiers and combine their outputs [1]. The combination of classifiers has the intuitive purpose to improve performance, specially on challenging problems like handwriting recognition and others.

In previous works, material analysis on soil science multispectral images were performed using statistical classifiers and simple combining rules with good results [2][3]. In this paper we present a set of experiments with a neural-network based classifier in order to recognize materials in noisier soil science multispectral images, obtained with less exposure time. These images were initially classified by a single Multilayer Perceptron classifier and later some combining techniques (Bagging, Decision Templates and Dempster-Shafer) were investigated. We also present a performance comparison between the individual classifiers and the combiners. The results were evaluated by the estimated error, obtained using the Cross-Validation technique. This paper also extends previous works [4][5] that presented some preliminary neural networks improvements using classifier combination.

2. Image Acquisition

The computerized tomograph (CT) scanner used to acquire the images is a first generation equipment developed by Embrapa in order to explore applications in soil science. It has fixed X and γ-ray sources, while the object being studied is rotated and translated. All the system hardware and software was developed by Embrapa [6].

In this work we used images of a phantom containing four materials commonly found in soil: calcium, phosphorus, water and aluminum. The phantom has a cylindrical base of plexiglass (polymer), and has four cylinders inside, each one containing one of the materials, as shown in Figure 1.

The images were obtained using two X-ray sources and two Γ-ray sources (Cesium and Americium). The X-ray energies were 40keV and 85keV. The Γ-ray sources were 662keV (Cesium) and 60keV (Americium). The images have a resolution of 65x65 pixels, and were obtained using only 3 seconds of exposure. After the reconstruction with the filtered backprojection algorithm, they were normalized to 256 levels of gray, which are proportional to the values of the physically observed linear attenuation coefficients. The four images are shown in Figure 2. Together they

[1] Fabricio A. Breve is currently with Institute of Mathematics and Computer Science, USP – University of São Paulo, São Carlos-SP, Brazil. E-mail: fabricio@icmc.usp.br.

1530-1834/07 $25.00 © 2007 IEEE
DOI 10.1109/SIBGRAPI.2007.10

compose a multispectral image, which is the object of study of this work.

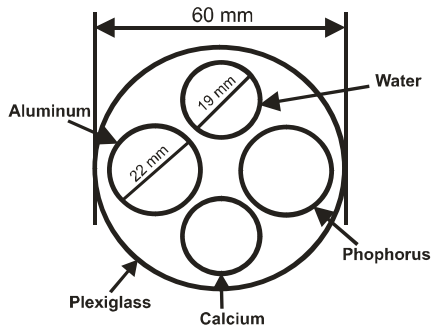

Figure 1. Phantom construction diagram with dimensions and materials

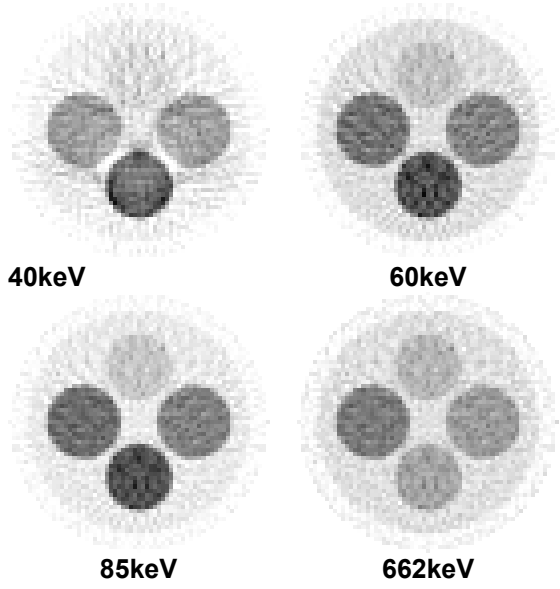

Figure 2. Multispectral image bands acquired by an X and γ-ray CT scanner with multiple energies: 40keV, 60keV, 85keV and 662keV

3. Classification Methods

Classification was first performed using individual Multilayer Perceptron classifiers. Later, Bagging technique, Decision Templates and Dempster-Shafer classifier combiners were used in order to improve the performance of the single classifiers.

3.1. The Multilayer Perceptron

The Multilayer Perceptron is composed by a set of sensorial units organized in three or more layers. The first layer is the input layer, which does not perform any computational task. Then there are one or more hidden (intermediate) layers and an output layer, all composed by computational nodes. In a typical MLP network all the nodes from a layer are connected with every node from the previous and from the next layer. There are no connections between nodes in the same layer, neither there are connections between nodes on non-adjacent layers. The non-computational nodes in the input layer use an identity function, while the computation nodes in the intermediate and the output layers use a sigmoid function [7].

This kind of neural network has been used with success to solve difficult problems through its training by using the error backpropagation algorithm, which basically consists of two steps: a forward step where the signal propagates through the computational units until it gets to the output layer; and a backwards step where all the synaptic weights are adjusted accordingly to an error correction rule [8].

3.2. Bagging

Bagging was created by Breiman [9] and is an acronym for Bootstrap AGGregatING. This method consists of creating bootstrap replicas of the training set and then training a different classifier with each replica. The outputs from each classifier are combined using majority voting. The bootstrap sets are built randomly from the original training set using substitution. To take advantage of this method it is essential that the base classifier be unstable, where minor changes in the training set can lead to major changes in the classifier output. Otherwise, we will have just a set of almost identical classifiers and combining them would lead to little or no improvement in the classification at all. An example of stable classifier is the K-Nearest Neighbor, while the Multilayer Perceptron is an example of an unstable classifier.

3.3. Decision Templates

When using classifiers that give us continuous-valued outputs (like MLP) we can treat the outputs as confidences in proposed labels and estimates of the posterior probabilities for each class. Let $x \in \Re^n$ be the feature vector and $\Omega = \{\omega_1, \omega_2, ..., \omega_c\}$ be the

set of labels from each class. Each classifier D_i from a given set $D = \{D_1,...,D_L\}$ provides *c* degrees of support. We can assume that all the *c* degrees are in the [0, 1] interval, that means $D_i : \Re^n \rightarrow [0,1]^c$. The notation $d_{i,j}(x)$ represents the support degree that the classifier D_i gives to *x* being from the class ω_j. The *L* outputs from the classifiers for a given input *x* can be organized in a matrix called *decision profile* (*DP*(*x*)), organized as follows:

$$DP(x) = \begin{bmatrix} d_{1,1}(x) & \cdots & d_{1,j}(x) & \cdots & d_{1,c}(x) \\ \vdots & & \vdots & & \vdots \\ d_{i,1}(x) & \cdots & d_{i,j}(x) & \cdots & d_{i,c}(x) \\ \vdots & & \vdots & & \vdots \\ d_{L,1}(x) & \cdots & d_{L,j}(x) & \cdots & d_{L,c}(x) \end{bmatrix}. \quad (1)$$

In this combiner the idea is to remember the most typical Decision Profile (DP) for each class, and call them the Decision Template (DT) of each class. Then, when we want to classify a given sample we build its DP and compare it with the DT of each class using some measure distance (like the Euclidean distance). The closest match will label the sample [1].

To calculate a decision template for the *j* classes we take the mean of the decision profiles $DP(z_k)$ from all the members of ω_j from the training data set *Z*:

$$DT_j = \frac{1}{N_j} \sum_{\substack{z_k \in \omega_j \\ z_k \in Z}} DP(z_k), \quad (2)$$

where N_j is the number of elements from *Z* that belongs to ω_j.

To classify a sample, we construct its decision profile $DP(x)$ and calculate the similarity *S* between $DP(x)$ and each DT_j:

$$u_j(x) = S(DP(x), DT_j) \qquad j = 1,\ldots,c. \quad (3)$$

3.4. Dempster-Shafer

Dempster-Shafer (DS) is based on the Evidence Theory, proposed by Glen Shafer as a way to represent cognitive knowledge. In this formalism the best probability representation is a belief function, rather than a Bayesian distribution. Probability values are assigned to a set of possibilities instead of unique events. Its appeal is in the fact that they code evidences rather than propositions. It provides a simple method of combining evidences from different sources (Dempster rule) without any a priori distribution [10].

The training algorithm for DS is the same algorithm used to train the DT combiner, where the DT´s for each class are found from the training data. The difference here is that instead of calculating the similarity between the DP of a given sample and each DT, we calculate the closeness between the DT and the output of each classifier. These closeness values are used to calculate a belief degree for each classifier for each one of the classes. The final degrees of support for each class are calculated from the belief degrees [1]. These steps are described below:

Let DT_j^i be the *i*th row of the decision template DT_j and $D_i(x)$ be the output of D_i, that is, $D_i(x) = [d_{i,1}(x),...,d_{i,c}(x)]^T$: the *i*th row of the decision profile *DP*(*x*). We calculate the "proximity" ϕ between DT_j^i and the output of the D_i classifier for some input **x**:

$$\phi_{j,i}(x) = \frac{(1 + \left\| DT_j^i - D_i(x) \right\|^2)^{-1}}{\sum_{k=1}^{c} (1 + \left\| DT_k^i - D_i(x) \right\|^2)^{-1}}, \quad (4)$$

where || . || is any matrix norm. For example, we can use the Euclidean distance between the two vectors. So, for each decision template we will have *L* proximities.

Using the last equation we can calculate for every class, *j* = 1,...,*c*; and for every classifier, *i* = 1,...,*L*, the following belief degrees:

$$b_j(D_i(x)) = \frac{\phi_{j,i}(x) \prod_{k \neq j} 1 - \phi_{k,i}(x))}{1 - \phi_{j,i}(x)[1 - \prod_{k \neq j}(1 - \phi_{k,i}(x))]}. \quad (5)$$

The final support degrees are given by

$$\mu_j(x) = K \prod_{i=1}^{L} b_j(D_i(x)) \qquad j = 1,...,c. \quad (6)$$

where *K* is a normalizing constant to keep the output in the [0-1] interval.

4. Evaluation

The performance of the classifiers and combiners were evaluated by the Error Estimated by the Cross-Validation technique, in which we took N pre-labeled samples, choose an integer K and randomly divide the N samples into K subsets of size N/K. Then we could use one subset to test the performance of a classifier trained on the union of the remaining K-1 subsets. This procedure was repeated K times, choosing a different subset for testing each time [11]. In this paper we used N=480 and K=48, so each subset had 10 samples.

Cross-Validation has high computational costs, specially because we are dealing with the Multilayer Perceptron (slow training) and classifier combination tasks (multiples classifiers to be trained), so it was avoided in previous works [4][5], in which the much faster hold-out technique was applied. In this paper we used Cross-Validation, training a total of more than 22 thousands classifiers, so we can expect these results to be more accurate than previous ones using the hold-out technique.

5. Experiments

In order to test the classifiers, we have taken 80 samples in 10x8 pixels windows (like illustrated in Figure 3) from each of the 6 classes (water, aluminum, phosphorus, calcium, plexiglass and background) in a total of 480 samples. From this set, we separated 48 subsets with 10 samples each, following the Cross-Validation technique, so we could use each of these subsets to test the performance of a classifier trained with the remaining 47 subsets. This procedure was repeated 47 times in order to test every selected sample. We used the four available bands, so we have 4 features, which also means that the input layer of our networks have 4 nodes.

The MLP networks we trained had from 2 to 10 units in one single hidden layer. These numbers were selected because there is no foolproof way to tell a priori how many units in the hidden layer would be the best choice [12]. The Nguyen-Widrow [13] initialization algorithm was used to setup the parameters in the MLP networks. Adaptive learning rates were used with 0.01 as the initial value for learning rate, 1.05 as the multiplier for increasing learning rate, 0.7 as the multiplier for decreasing learning rate, 0.95 as the momentum constant and 1.04 as the error ratio.

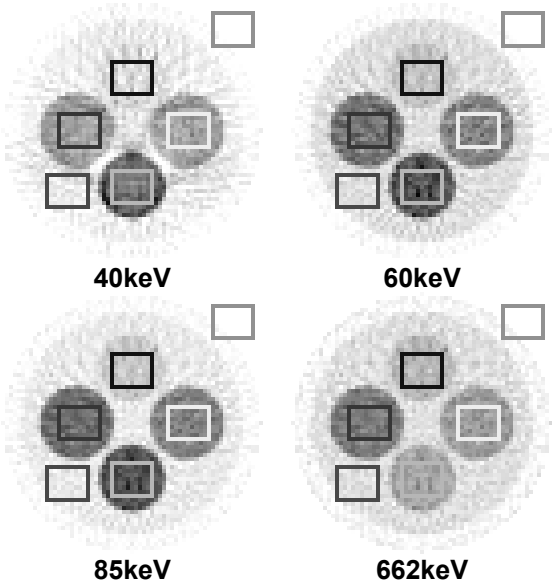

Figure 3. Pixel windows selected as samples in each image

The experiments using classifier combination were also evaluated using Cross-Validation techniques and the same number of subsets, so for each of the 48 subsets ten different classifiers were trained and combined as follows.

In the experiments using the Bagging technique we replaced the combination using majority voting rule by the mean rule, so we could take advantage of the continuous-valued output (soft labels) provided by our neural network-based classifiers. We combined 10 base classifiers with different bootstrap training samples and different initialization parameters.

In the experiments with the Decision Templates and Dempster-Shafer combiners we used the Euclidean distance to measure similarity. All the experiments with DT and DS trained 10 base classifiers for the combination with different initialization parameters.

After experimenting Bagging, DT and DS combiners, we decided to mix these techniques, in order to test if we could achieve any improvement. So, we repeated the experiments with MLP using Bagging technique again, but with the Decision Templates (BAGDT) and Dempster-Shafer (BAGDS) as the combiners, instead of the simple mean rule.

The results can be viewed in Table 1 and Figure 4. They show the error estimated for a single classifier and each combination scheme for each MLP configuration of units in the hidden layer. The best results in each column are in boldface and the best results in each line are in italics.

Table 1. Estimated Error for each combination scheme with different number of MLP units in the hidden layer

Units	Single	Bagging	DT	DS	BAGDT	BAGDS
2	0.6417	0.4812	*0.0271*	0.0521	0.0292	0.0542
3	0.2812	0.0208	*0.0167*	0.0188	0.0208	0.0292
4	0.1542	0.0208	*0.0146*	*0.0146*	0.0229	**0.0188**
5	0.0438	0.0167	***0.0125***	***0.0125***	0.0167	0.0208
6	0.0750	**0.0146**	0.0167	***0.0125***	**0.0146**	**0.0188**
7	0.0646	0.0167	*0.0146*	0.0167	0.0167	**0.0188**
8	0.0521	0.0188	0.0167	0.0167	***0.0146***	**0.0188**
9	0.0375	0.0188	0.0167	0.0167	***0.0146***	0.0208
10	**0.0292**	0.0188	*0.0167*	*0.0167*	0.0188	0.0188
Mean	0.1533	0.0697	*0.0169*	0.0197	0.0188	0.0243

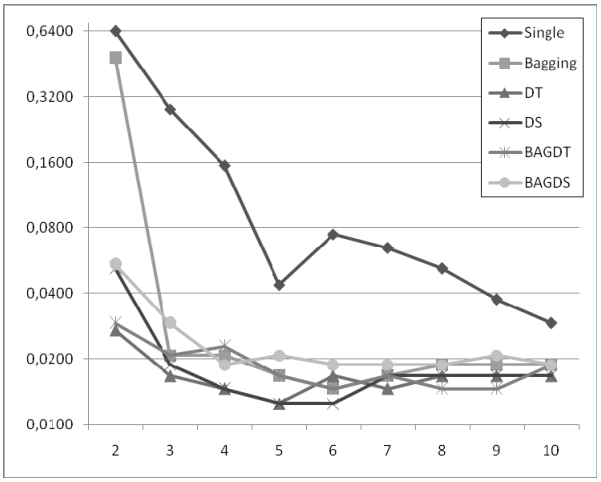

Figure 4. Estimated Error for each combination scheme with different number of MLP units in the hidden layer

6. Conclusions

In the experiments with the MLP single classifier we noticed that the results got better as we added units to the hidden layer. Also we have really bad results using only 2 or 3 units, and that is probably due to the unstable nature of the MLP and its lack of ability to escape from local minima depending on its initialization parameters. The use of classifier combiners overcomes this problem, because with ten different classifiers (and ten different initializations) chances are that some of them will reach the global minima. That is easy to perceive by analyzing the results with the combiners where the best results were achieved using fewer units in the hidden layer.

We can also realize that Decision Templates combiners showed good results no matter how many units there were in the hidden layer. The reason for this behavior is likely to be that decision templates are constructed based on the most common output of the classifiers for the training samples from each class, no matter if they are giving the right label for those samples or not. For example, if a single classifier always misclassifies samples of Aluminum as being Water, DT technique will still take advantage of it. In fact, if only one of the ten base classifiers performs a good classification, DT will probably perform a good combination. Even if we have only average classifiers, DT still can perform combination. So we can conclude that DT should be a good choice of combiner when it is hard to find the parameters to train a classifier that escapes from local minima or when it is not viable to conduct experiments to find out which is the optimal number of units in the hidden layer for a particular problem.

The techniques including the Bagging method (BAGDT and BAGDS) seem to perform slightly worse than DT or DS alone. Bagging takes advantage of unstable classifiers where minor changes in the training samples lead to major changes in the classification. But MLP classifiers are unstable by themselves, which means changing only the initialization of the parameters is enough to produce entirely different classifications. So it seems that the extra "disorder" placed by the bagging technique is unnecessary and does not improve the combination of MLP classifiers, at least in this particular case.

New advances in this field could be reached with use of mechanisms to eliminate redundant classifiers, constraint weak classifiers and adaptive combination. A contextual approach for the classifier combination methods is also another good research way.

In general, the results show that using Neural Network based classifiers, particularly the MLP, to identify materials on CT images is viable even in images with high noise levels. The use of classifiers combiners led to better classification and more stable MLP systems, minimizing the effects of bad choices of initialization parameters or configuration (mainly the number of units in the hidden layer) and the unstable nature of the individual MLP classifiers.

7. Acknowledgements

We would like to thank Dr. Paulo E. Cruvinel for providing the multispectral images used in this paper. This work was also partially supported by CAPES and FAPESP (grant n. 04/05316-7) and by FAPESP Thematic Project 02/07153-2.

8. References

[1] L. Kuncheva, Combining Pattern Classifiers, Wiley-Interscience, Hoboken, NJ, 2004.

[2] M. R. P. Homem, N. D. A. Mascarenhas, and P. E. Cruvinel, "The Linear Attenuation Coefficients as Features of Multiple Energy CT Image Classification", Nuclear Instruments and Methods in Physics Research, v.45, n. 2, pp. 351-360, 2000.

[3] M. P. Ponti Jr., and N. D. A. Mascarenhas, "Material Analysis on Noisy Multispectral Images Using Classifier Combination", Proc. IEEE Southwest Symposium on Image Analysis and Interpretation Computer Society Press, Lake Tahoe, Nevada, pp. 28-30, 2004.

[4] F. Breve, M. P. Ponti Jr., and N. D. A. Mascarenhas, "Combining Methods to Stabilize and Increase Performance of Neural-Network Based Classifiers", Proceedings of XVIII Brazilian Symposium on Computer Graphics and Image Processing, Natal, RN, pp. 105-111, 2005.

[5] F. Breve, M. P. Ponti Jr., and N. D. A. Mascarenhas. "Neural-Network Combination for Noisy Data Classification", Anais do II Workshop de Visão Computacional, São Carlos, SP, 2006.

[6] P.E. Cruvinel, R. Cesareo, and S. Mascarenhas. "X and γ-Rays Computerized Minitomograph Scanner for Soil Science", IEEE Transactions on Instrumentation and Measurementes, v. 39, n. 5, pp. 745-750, 1990.

[7] C. M. Bishop. Neural Networks for Pattern Recognition, Oxford, New York, 1995.

[8] S. Haykin. Redes Neurais – Princípios e Prática, Bookman, 2 ed., Porto Alegre, 2001.

[9] L. Breiman. "Bagging Predictors", Machine Learning, v. 26, n. 2, pp. 123-140, 1996.

[10] M.R. Ahmadzadeh, M. Petron, and K.R. Sasikala "The Depster-Shafer Combination Rule as a Tool to Classifier Combination". Geoscience and Remote Sesing Symposium. Proc. IGARSS, IEEE International, pp. 2429-2431, 2000.

[11] R. Kohavi. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection" Proc. of the 14th Int. Joint Conf. on A. I.., v. 2, Canada, pp. 1137-1143. 1995.

[12] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, Wiley, 2ed, New York, 2000.

[13] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," International Joint Conference of Neural Networks, V. 3, pp. 21-26, 1990.