Combining Methods to Stabilize and Increase Performance of Neural Network-Based Classifiers

Fabricio A. Breve, Moacir Jr. P. Ponti, Nelson D. A. Mascarenhas
Departamento de Computação – Universidade Federal de São Carlos, São Paulo, SP, Brasil
{fabricio, moacir, nelson}@dc.ufscar.br

Abstract

In this paper we present a set of experiments in order to recognize materials in multispectral images, which were obtained with a tomograph scanner. These images were classified by a neural network based classifier (Multilayer Perceptron) and classifier 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 Hold-Out technique) and the Kappa coefficient, and they showed performance stabilization.

1. Introduction

Multiple classifier systems based on the combination of several different classifiers are currently used to increase recognition performance. It had been observed that the sets of patterns misclassified by different classifiers would not necessarily overlap. For neural network based classifiers as Multilayer Perceptron this can be a way to increase and stabilize performance. An interesting issue on classifier combination research is the scheme by which they are combined.

A previous work used the Maximum Likehood, ICM and K-Means single classifiers to identify materials in tomography images with low noise [1]. Later, another work was presented, using images with high levels of noise, obtained with low exposure time. The behavior of Parzen, K-Nearest Neighbors, Logistic and Linear Maximum Likehood classifiers were observed, and also the behavior of the Product, Maximum and Minimum Combiners [2]. In this work we used the same noisy images used in the previous work [2] *(Figure 2)* in order to evaluate the behavior of a neural network (NN) based classifier in this kind of image, as well as to evaluate combination of single NN-based classifiers.

The main contribution of this paper is the investigation of classifier combination methods, not just on performance improvement like other recent works using linear combination [3], rough set reduction [4] and Dempster-Shafer Theory of Evidence [5], but also on their behavior when combining Multilayer Perceptron classifiers that are unstable due to the random initialization values. For this task, Bagging, Decision Templates and Dempster-Shafer combiners are used. The observation is carried out using different settings for the hidden layer allowing to analyze in which level of complexity of the NN classifiers the combination can assure better performance.

This paper is organized as follows. Section 2 presents the acquisition of the image used in the experiments. Section 3 is concerned about the classification and combination methods. Section 4 explains the performance evaluation method. The experiments and results are described on Section 5. Finally, Section 6 has the conclusion and final remarks.

2. Image Acquisition

The computerized tomograph (CT) scanner used to acquire the images is a first generation equipment developed by Embrapa[1] in order to explore applications in soil science. It has the X and γ-ray sources fixed while the object being studied is rotated and translated. All the system is controlled by hardware and software developed by Embrapa. [6]

In this work we have images of a phantom that was built with materials found in soil. The phantom used has a support of plexiglass (polymer) and has 4 cylinders containing: aluminum (left), water (top), phosphorus (right) and calcium (bottom), as shown in Figure 1.

[1] Brazilian Agricultural Research Corporation – institute for scientific reseach on agriculture, providing feasible solutions for the developmente of Brazilian agribusiness.

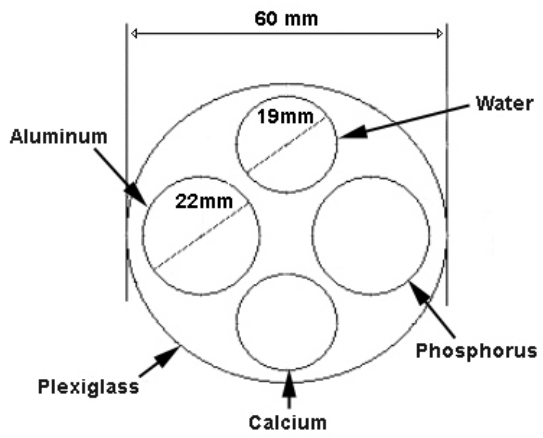

Figure 1. Phantom construction diagram with dimensions and materials

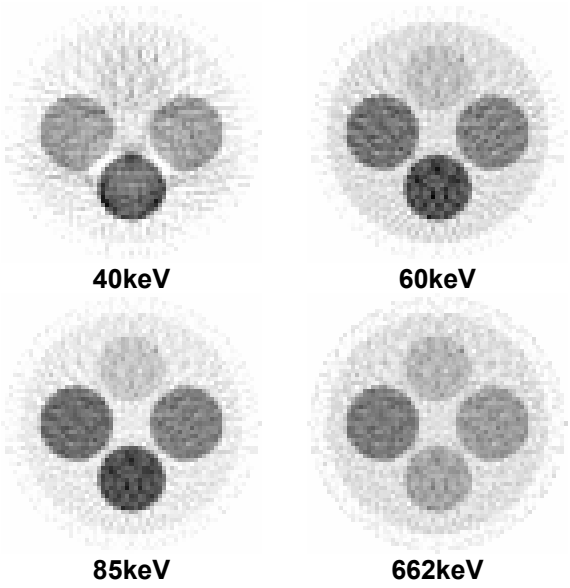

Figure 2. Multispectral image bands acquired by an X and γ-ray CT scanner with multiple energies: 40keV, 60keV, 85keV and 662keV

To obtain the images two X-ray sources and two γ-ray sources (Cesium and Americium) were used. The X-ray energies were 40keV and 85keV. The Γ-ray were 662keV (Cesium) and 60keV (Americium).

Images with size 65x65 pixels of this phantom were obtained from each radioactive source using 3 seconds of exposure. After the reconstruction with the filtered backprojection algorithm the images were normalized to 256 levels of gray, which are proportional to the values of the physically observed linear attenuation coefficients.

3. Classification Methods

The classification was performed using the Multilayer Perceptron (MLP) classifier and some classifiers combiners.

3.1. Multilayer Perceptron

The Multilayer Perceptron is a kind of network that is composed by a set of sensorial units in the *input layer*, one or more *hidden layers* and an *output layer* of computational units. The signal is set in the input layer and propagates through the network until it gets to the output layer.

This kind of network has being used with success to solve difficult problems through its training by using the *error backpropagation algorithm*, which basically consists of two steps: a step forward where the signal propagates through the computational units until it gets to the output layer; and a step backwards where all the synaptic weights are adjusted accordingly to an error correction rule. [7]

In a typical MLP network all the units from a layer are connected with every unit from the previous and from the next layer. There are no connections between units in the same layer; neither there are connections from non-adjacent layers. The function in the input layer units is an identity function and these units do not perform computational tasks. The hidden and the output layers have Sigmoid functions like this: [8]

$$g(a) \equiv \frac{1}{1+\exp(-a)}$$

and their output are usually in the [0-1] interval.

3.2 Bagging

The term Bagging was created by Breiman [9] and is an acronym for *Bootstrap AGGregatING*. This method consists of building bootstrap replicas of the training set and then training each one alone. The outputs from each classifier are then combined by 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, a classifier where small changes in the training set lead to big changes in the classifier output. Not to follow this rule means that we will have a set of almost identical classifiers. Examples of unstable classifiers are the neural network based, while

the k-nearest neighbor is an example of a stable classifier.

3.3 Decision Templates

When using classifiers that give us continuous-valued outputs (like the Multilayer Perceptron) 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 $\Omega = \{\omega_1, \omega_2, ..., \omega_c\}$ the set of labels from classes. Each classifier D_i from the set $D = \{D_1, ..., D_L\}$ provides *c* degrees of support. We can assume that all the *c* degrees are in [0, 1] interval, that is, $D_i : \Re^n \rightarrow [0,1]^c$. The notation $d_{i,j}(x)$ represents the support degree that the classifier D_i gives to *x* being from ω_j. The *L* outputs from the classifiers for a given input *x* can be organized in a *decision profile* (*DP*(*x*)) as in the matrix:

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

Decision Templates (DT) is a kind of combiner where the idea is to remember the most typical decision profiles for each class, called *decision templates*, and then compare them with the current decision profile using some similarity measure (like the Euclidean distance). The closest match will label the sample. [10]

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:

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

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

After training, given an input, we construct its decision profile $DP(x)$ and calculate the similarity *S* between $DP(x)$ and each DT_j:

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

3.4 Dempster-Shafer

This method is based on the Evidence theory, introduced by Glenn 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. [11]

The Dempster-Shafer combiner (DS) training is like DT training, the *c* decision templates are found from the data. However, instead of calculating the similarity between the decision template and the decision profile we calculate the proximity between the decision template and the output of each classifier. These are used to calculate the belief degree for every class. At last the final degrees of support for each class are calculated from the belief degrees. These steps are described below: [10]

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**:

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

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:

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

The final support degrees are given by

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

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

4. Evaluation of the Classification

The performances of the methods were evaluated by using the estimated error by the Hold-Out technique and the Kappa coefficient.

4.1 Hold-Out

In this method, usually we split the set of available data in two halves. The first one is used to train the classifier, and the second one is used to test it, obtaining the error rate. [10]

This method is pessimistic because it uses only a portion of the data available for training [12]. However, when testing Multilayer Perceptron based classifiers, a fast testing scheme is often required because this kind of network takes much more time to train than statistical based classifiers. Hold-Out method can train only a single classifier to obtain the estimated error rate, therefore this method is a good choice to evaluate Multilayer Perceptron based classifiers.

4.2 Kappa Coefficient

The Kappa coefficient can be used to measure the agreement rating between two classifiers. To evaluate the performance of a classifier we can use Kappa to compare the output of a classifier with the pre-labeled samples [13].

The values given by Kappa coefficient (K) are in the [-1 1] interval. When the agreement is no higher than expected in a random classification K will be 0. K will output 1 when there is a total agreement between the sample labels and the classifier output.

The higher the value of K, the best is the performance of the classifier. The interpretation of the Kappa coefficient is subjective and depends on the level of correctness required by the problem. [14][15]

5. Experiments

In the experiments we took a set of 80 samples in 10x8 pixels windows from each of the 6 classes – water, aluminum, calcium, phosphorus, plexiglass and background – in a total of 480 samples. Then we split this set in two groups with 240 samples each (40 samples from each class). The first one was used for training and the second one for testing. We used all the 4 bands available, so the input layer size is 4.

Since there is no fail proof way to determine how many units in the hidden layer would be the best choice [16], we trained classifiers with 2 to 15 units in one single hidden layer.

MLP based classifiers tend to present different results due to its random initialization parameters, so every experiment for all classification methods were executed 100 times, and the results in this paper are the mean value in those 100 times. The Nguyen-Widrow initialization algorithm was used to initialize the MLP networks. Adaptive learning rate were used.

In the experiments with the Bagging technique we replaced the combination using majority vote for the combination using means, so we could take advantage of the soft labels (continuous-valued outputs) provided by MLP. For Decision Templates combiner we used the Euclidean distance. All the experiments with classifiers combiners trained 10 base classifiers for the combination.

The classification results can be viewed in Table 1 (Estimated Error) and Table 2 (Kappa Coefficient). The best results are in boldface. Thematic images for the best classifier in each group are available in Figure 5 and 6. Figure 5 shows thematic images for the best case in the 100 experiments while Figure 6 shows the worst case.

6. Conclusions

In the experiments using a single classifier we noticed that the classification becomes better as the number of units in the hidden layer increases, however even the best case (15 units) still produces bad results sometimes, due to the nature of MLP, as we can see in Figure 6. The use of classifier combination led to more stable classifiers, where even the worst case still delivers a good classification. It is also important to notice that the best results with all the combiners were achieved with few units in the hidden layer.

Dempster-Shafer and Decision Templates combiners showed relatively good results no matter how many units there were in the hidden layer, improving the performance, especially when using few neurons on hidden layer. Therefore, we could say they would be good choices of combiners in situations where is not viable to conduct experiments to find the optimal number of units in the hidden layer for a particular problem.

Some future work could include experiments with more than 10 base classifiers to observe if the performance improvements could worth the extra time to train all the base classifiers. Also it would be interesting to observe the results of a combination using Bagging techniques, but using Dempster-Shafer or Decision Templates as the combiner, instead of the mean combiner we used in our experiments.

The results shows that using Multilayer Perceptron based classifiers to identify materials on CT images is viable, even in images with high noise levels. The use of classifiers combiners led to more stable systems and minimized the effects of 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 the experiments, CAPES and FAPESP (grant n. 04/05316-7) for student scholarship. This work was also partially supported by FAPESP Thematic Project 2002/07153-2.

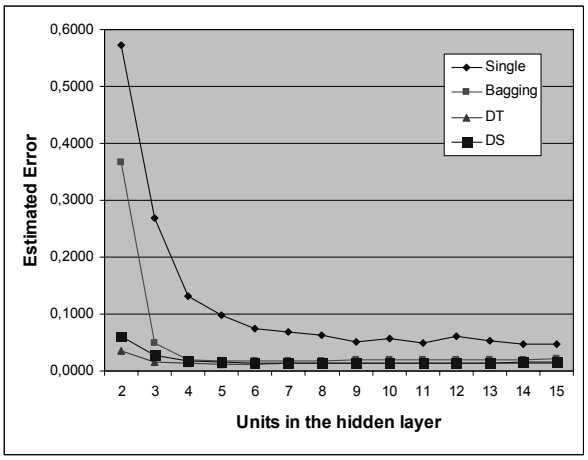

Figure 3. Estimated Error

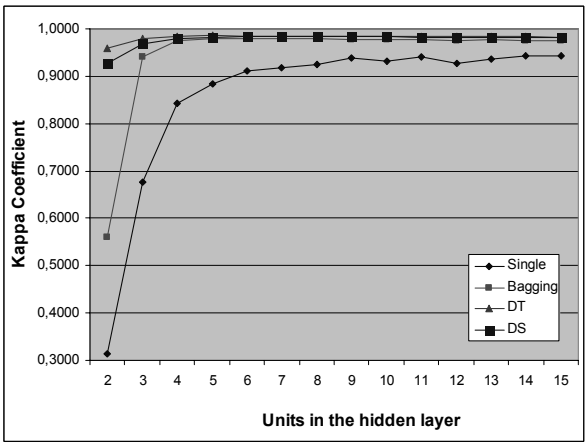

Figure 4. Kappa Coefficient

Table 1. Estimated Error

Units in the hidden layer	Single Classifier	Bagging	DT	DS
2	0.5720	0.3675	0.0349	0.0613
3	0.2689	0.0493	0.0163	0.0275
4	0.1318	0.0200	0.0141	0.0177
5	0.0976	0.0170	**0.0123**	0.0151
6	0.0741	**0.0168**	0.0127	0.0139
7	0.0681	0.0175	0.0129	0.0138
8	0.0636	0.0179	0.0130	**0.0137**
9	0.0511	0.0190	0.0134	0.0138
10	0.0570	0.0190	0.0135	0.0139
11	0.0497	0.0191	0.0136	0.0143
12	0.0603	0.0203	0.0136	0.0143
13	0.0525	0.0196	0.0137	0.0145
14	0.0477	0.0204	0.0140	0.0151
15	**0.0470**	0.0210	0.0143	0.0150

Table 2. Kappa Coefficient

Units in the hidden layer	Single Classifier	Bagging	DT	DS
2	0.3137	0.5591	0.9581	0.9265
3	0.6773	0.9409	0.9805	0.9671
4	0.8419	0.9760	0.9831	0.9788
5	0.8829	0.9796	**0.9853**	0.9819
6	0.9111	**0.9799**	0.9848	0.9833
7	0.9183	0.9790	0.9845	0.9835
8	0.9237	0.9786	0.9844	**0.9836**
9	0.9387	0.9773	0.9840	0.9835
10	0.9316	0.9773	0.9838	0.9833
11	0.9404	0.9771	0.9837	0.9829
12	0.9277	0.9757	0.9837	0.9829
13	0.9371	0.9765	0.9836	0.9827
14	0.9428	0.9756	0.9832	0.9819
15	**0.9436**	0.9748	0.9828	0.9821

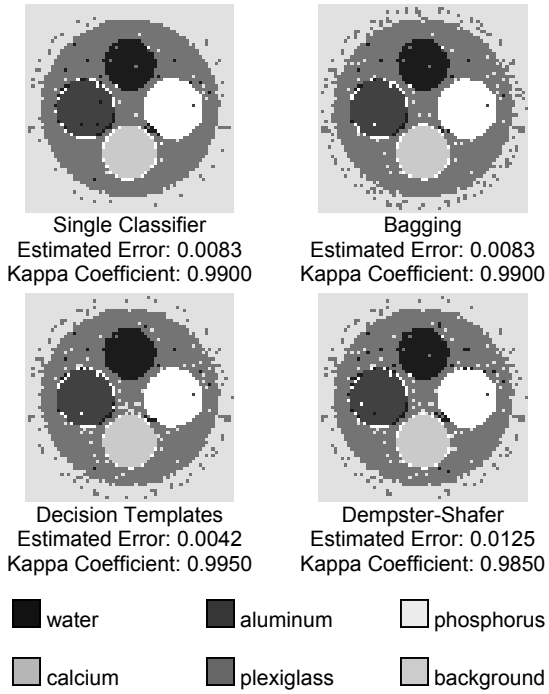

Figure 5. Thematic images for the best classifier of each group (best case)

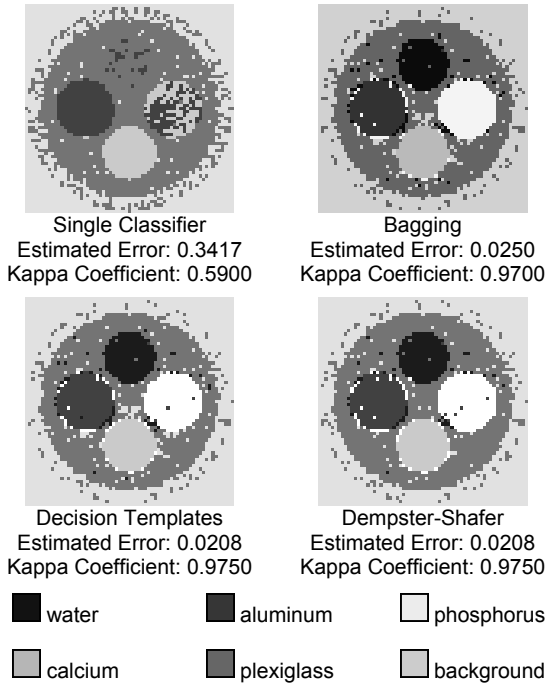

Figure 6. Thematic images for the best classifier of each group (worst case)

8. References

[1] 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. 452, 2000, pp. 351-360.

[2] M.P.-Jr. Ponti., and N.D.A. Mascarenhas. "Material Analysis on Noisy Multispectral Images Using Classifier Combination", *Proc. IEEE Southwest Symposium on Image Analysis and Interpretation*, 6th, 28-30 mar.2004. pp. 1-5.

[3] N. Ueda, "Optimal Linear Combination of Neural Networks for Improving Classification Performance", *IEEE Trans. Pattern Analysis and Machine Intelligence,* v. 22, n.2, 2000, pp. 207-215.

[4] D. Yu, Q. Hu and W. Bao. "Combining Multiple Neural Networks for Classification Based on Rough Set Reduction", *Proc. IEEE 2001 Int. Conf. Neural Networks and Signal Processing*, v.1, dec. 2003, pp. 543-548.

[5] A. A1-Ani and M. Deriche. "A Dempter-Shafer Theory of Evidence Approach for Combining Trained Neural Networks", *Proc. IEEE Int. Symp. Circuits and Systems*, v.3, may. 2001, pp. 703-706..

[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, 1990. p. 745-750.

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

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

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

[10] L. Kuncheva. "Combining Pattern Classifiers", *Wiley-Interscience*, Hoboken, NJ, 2004.

[11] 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, 2000, IEEE International, pp. 2429-2431.

[12] 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.*., Vol. 2, Canada, 1995, pp. 1137-1143.

[13] J. Carletta. "Assessing Agreement on Classification Taks: the Kappa Statistic", *Computational Linguistics*, v. 22(2), 1996, pp. 249-254.

[14] R.G. Congalton, “A Review of Assessing the Accuracy of Classifications of Remotely Sensed Data”, *Remote Sensing Environment,* v.37, 1991, pp. 35-46.

[15] J. Cohen. “A Coefficient of Agreement for Nominal Scales”. *Educational and Psychological Measurement*. v.20, n.1, 1960, p. 37-46.

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