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

Fabricio A. Breve[1]	fabricio@dc.ufscar.br
Moacir P. Ponti Jr.	moacir@dc.ufscar.br
Nelson D. A. Mascarenhas	nelson@dc.ufscar.br

DC – Departamento de Computação

UFSCar - Universidade Federal de São Carlos, São Paulo, SP, Brasil

[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.

Goals

- Recognize materials in multispectral images (obtained with a tomograph scanner) using a Neural Network based classifier (Multilayer Perceptron)
- Investigate classifier combination techniques in order to improve and stabilize the performance of the Multilayer Perceptron

Summary

- Image Acquisition
- Review of the Combination Methods
- Experiments Setup
- Evaluation
- Combination
- Results
- Conclusions

Image Acquisition

- First generation Computerized Tomograph developed by Embrapa in order to explore applications in soil science
  - X-Ray and γ-ray fixed sources
  - Object being studied is rotated and translated

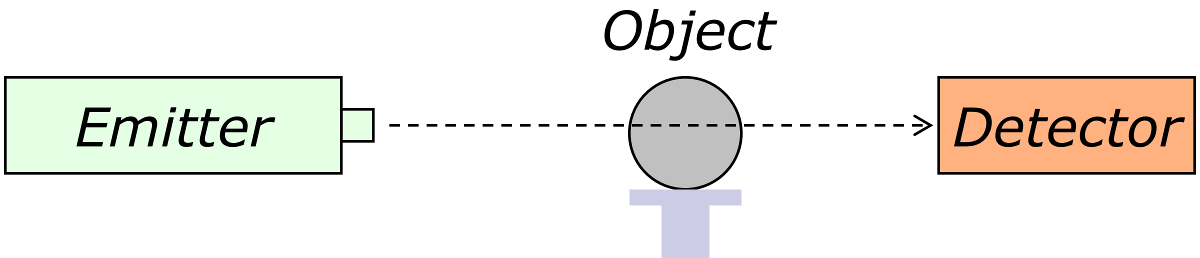


Image Acquisition

- Phantom built with materials found in soil
- Plexiglass support
- 4 Cylinders containing:
  - Aluminum
  - Water
  - Phosphorus
  - Calcium

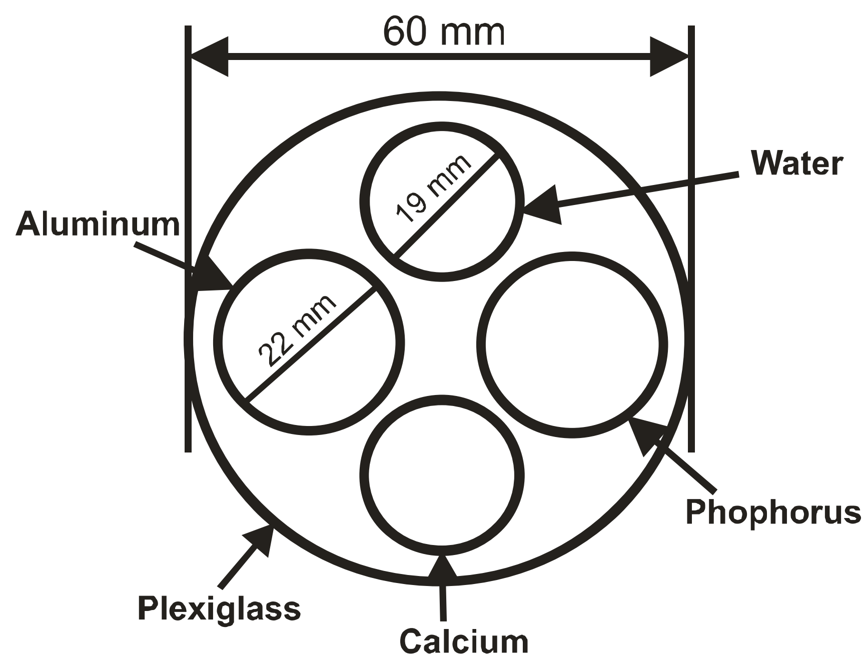


Image Acquisition

- Resolution: 65x65 pixels
- 256 gray levels
- Negative images for better visualization
- Exposure: 3 seconds
  - High noise level

40 keV
X-Ray

60 keV
γ-ray
(Americium)

85 keV
X-Ray

662 keV
γ-ray
(Cesium)

Combination Methods

- Bagging
  - Bootstrap AGGregatING
  - Bootstrap sets built randomly from the original training set using substitution
  - Each bootstrap set trains a classifier
  - Outputs are combined using majority voting
  - Requires the base classifier to be unstable
    - minor differences in the training set can lead to major changes in the classifier

Combination Methods

- *Decision Profile* (*DP*(**x**))

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

Decision Templates

- Continuous-valued outputs from each classifier for a given sample are used to build a decision profile
  - Classifiers have different initializations
- The Decision Templates are the mean over all the decision profile from each training sample for each class

$$DT_j = \frac{1}{N_j} \sum_{\substack{\mathbf{z}_k \in \omega_j \\ \mathbf{z}_k \in \mathbf{Z}}} DP(\mathbf{z}_k)$$

Decision Templates

- The label of a test sample is chosen by comparing its decision profile with each decision template and choosing the most similar one
- This technique also takes advantage of classification mistakes

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

Dempster-Shafer

- Based on the Evidence Theory, a way to represent cognitive knowledge
- It's like the Decision Templates method, but for each test sample we calculate the "proximity" between each row of the DT and the output of each classifier for a given sample **x**

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

Dempster-Shafer

- “Proximities” are used to calculate the belief degree for every class.

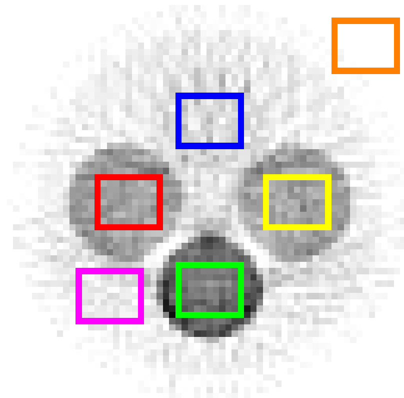
$$b_j(D_i(\mathbf{x})) = \frac{\phi_{j,i}(\mathbf{x}) \prod_{k \neq j} (1 - \phi_{k,i}(\mathbf{x}))}{1 - \phi_{j,i}(\mathbf{x}) \left[1 - \prod_{k \neq j} (1 - \phi_{k,i}(\mathbf{x})) \right]}$$

- At last the final degrees of support of each test sample for each class are calculated from the belief degrees.

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

Experiments Setup

- 480 samples (80 samples from each of the 6 class):
  - Aluminum
  - Water
  - Phosphorus
  - Calcium
  - Plexiglass
  - Background



40keV

60keV

85keV

662keV

Experiments Setup

- 4 band $\Rightarrow$ 4 features $\Rightarrow$ 4 units in the input layer
- Networks with 2 to 10 units in one single hidden layer
  - there is no foolproof way to tell a priori how many units in the hidden layer would be the best choice
- 6 classes $\Rightarrow$ 6 units in the output layer
- Free parameters setup by Nguyen-Widrow initialization algorithm
- Adaptive learning rates

Evaluation

- Cross-Validation
  - Set of 480 samples
    - Split in 48 subsets of 10 samples each
  - For each subset:
    - Train the classifier with the samples from the other 47 subsets
    - Test the classifier with the remaining subset
  - High Computational Costs
    - Multilayer Perceptron $\Rightarrow$ slow training
    - Classifier Combination $\Rightarrow$ multiple classifiers to be trained
  - 22.000 Multilayer Perceptron classifiers were trained for this paper
    - So we expect the results to be quite reliable
    - More accurate than *Hold-Out* used in previous works

Combination

- Bagging
  - Combination using majority voting rule was replaced by the mean rule
    - To take advantage of the continuous-valued output (soft labels)
  - For each cross-validation iteration (1 to 48)
    - Combine 10 base classifiers with different initialization parameters and different bootstrap training samples
      - Taken from the 470 samples of the 47 training subsets
    - Test the combination with the remaining subset
      - 10 samples

Combination

- Decision Templates (DT) and Dempster-Shafer (DS)
  - Euclidean distance to measure similarity
  - For each cross-validation iteration (1 to 48)
    - Combine 10 base classifiers with different initialization parameters
      - Using the 470 samples from the 47 training subsets
    - Test the combination with the remaining subset
      - 10 samples

Combination

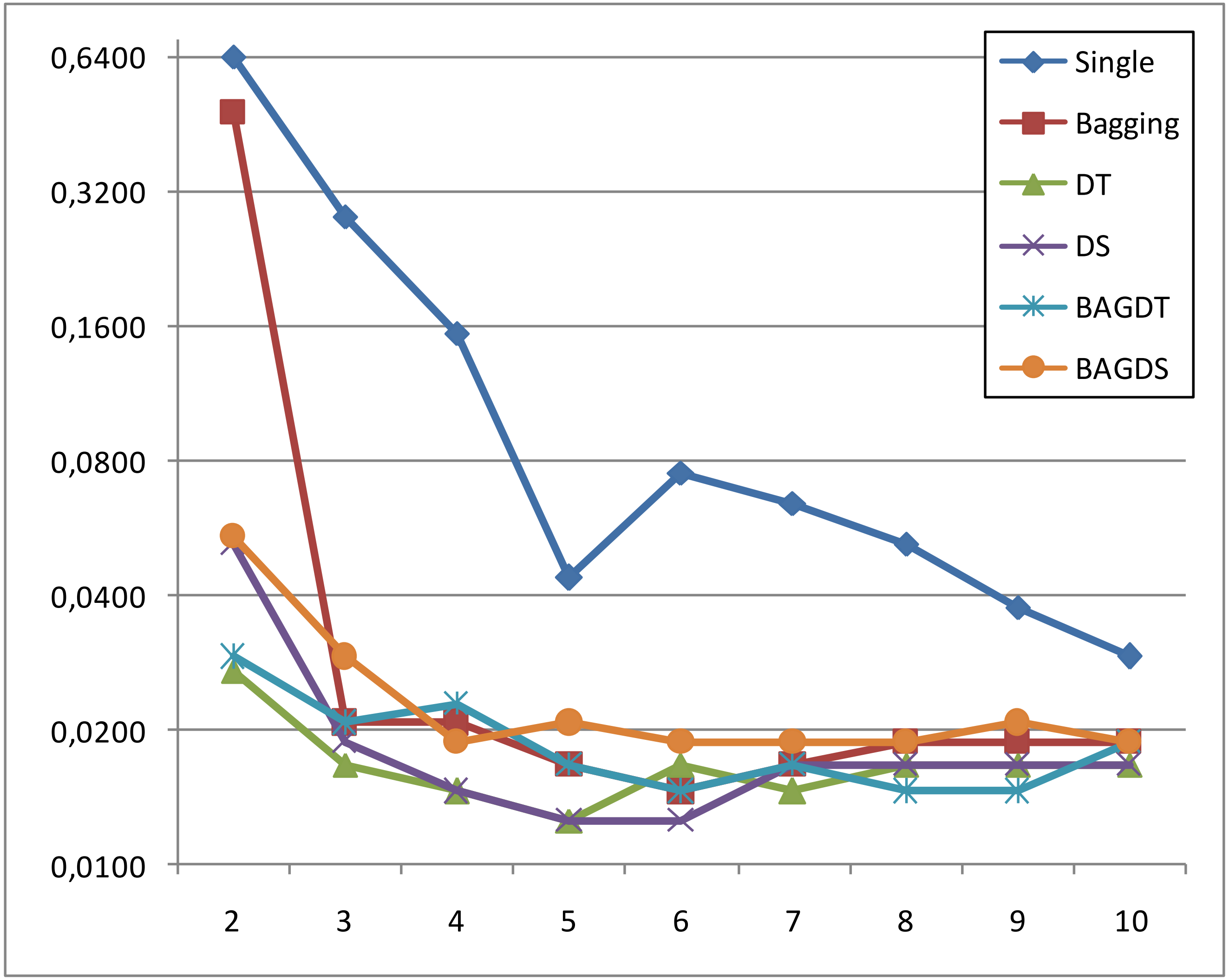
- Mixed techniques:
  - Bagging + Decision Templates (BAGDT)
  - Bagging + Dempster-Shafer (BAGDS)
- DT and DS as the combiners
  - Instead of the voting or simple mean rule

Results

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

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

Results


0,6400
0,3200
0,1600
0,0800
0,0400
0,0200
0,0100
2
3
4
5
6
7
8
9
10
Single
Bagging
DT
DS
BAGDT
BAGDS

Conclusions

- MLP single classifier
  - Results got better as we added units to the hidden layer
  - Really bad results using only 2 or 3 units
    - Unstable nature of the MLP and its lack of ability to escape from local minima depending on its initialization parameters.
  - Classifier combiners overcomes this problem
    - 10 different classifiers and 10 different initializations
      - Chances are that some of them or at least one of them will reach the global minima

Conclusions

- Decision Templates
  - Good results no matter how many units there were in the hidden layer.
    - Good performance if at least one of the base classifiers performs a good classification
    - Even if we have only average classifiers, DT still can perform good combination.
  - Should be a good choice of combiner when
    - It is hard to find the parameters to train a classifier that escapes from local minima
    - 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

Conclusions

- BAGDT and BAGDS
  - Seem to perform slightly worse than DT or DS alone
    - Bagging takes advantage of unstable classifiers
      - minor changes in the training samples lead to major changes in the classification.
    - MLP is unstable by itself
      - changing only the initialization of the parameters is enough to produce entirely different classifications
    - The extra “disorder” placed by the bagging technique is unnecessary

Conclusions

- Multilayer Perceptron is viable to identify materials on CT images, 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
    - including the number of units in the hidden layer
  - The unstable nature of the individual MLP classifiers

Acknowledgements

- 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

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

Fabricio A. Breve[1] fabricio@dc.ufscar.br
Moacir P. Ponti Jr. moacir@dc.ufscar.br
Nelson D. A. Mascarenhas nelson@dc.ufscar.br

DC – Departamento de Computação
UFSCar - Universidade Federal de São Carlos, São Paulo, SP, Brasil

[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.