Neural-Network Combination for Noisy Data Classification

Fabricio A. Breve 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

Goals

- Recognize materials in multispectral images (obtained with a tomograph scanner) using Neural Network based classifiers
- Investigate classifier combining techniques in order to improve performance

Summary

- Image Acquisition
- Classification Methods
- Combination Methods
- Evaluation
- Experiments
- 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

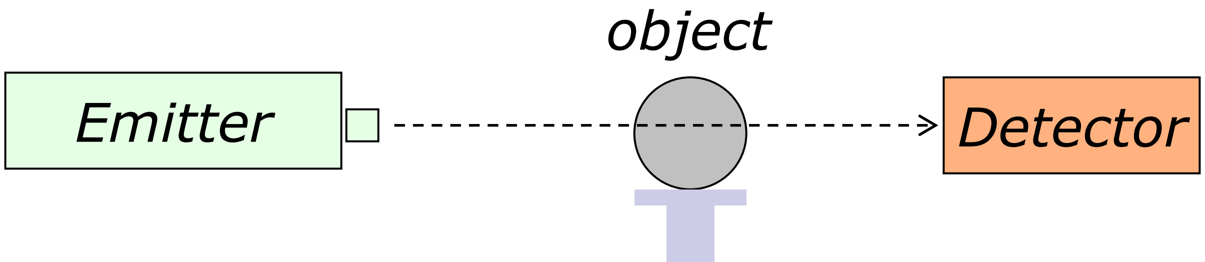


Image Acquisition

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

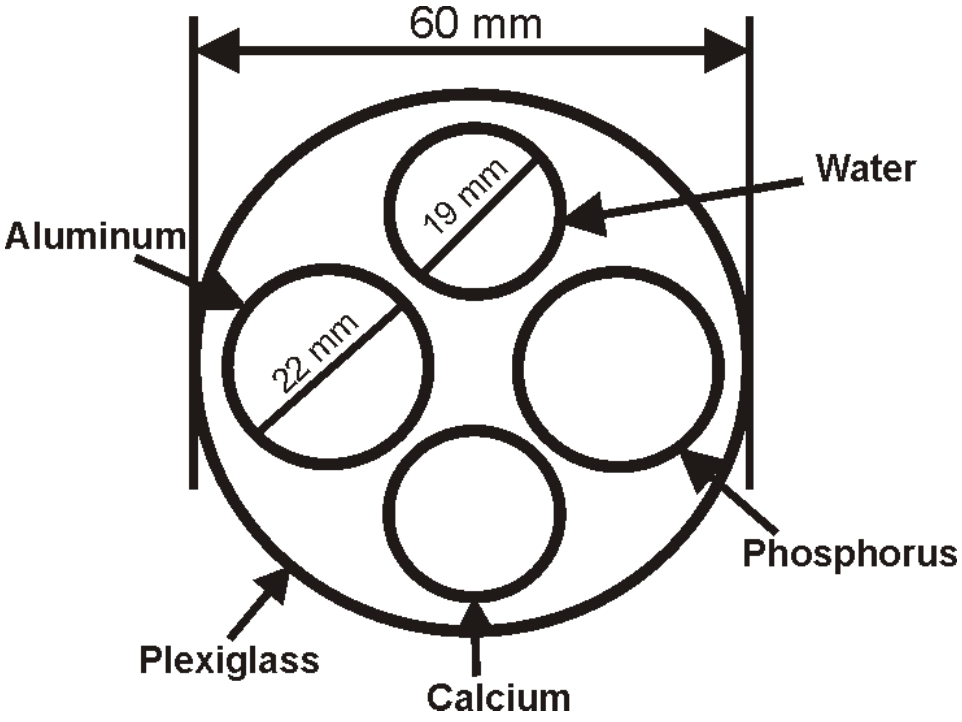


Image Acquisition


40keV


60keV


85keV


662keV

- X-ray sources:
  - 40 keV
  - 85 keV
- γ-ray sources
  - 60 keV (Americium)
  - 662 keV (Cesium)
- 65x65 pixels
- 256 levels of gray
- 3 seconds of exposure
  - High level of noise

Classification Methods

- Multilayer Perceptron
  - Composed by a set of sensorial units organized in three or more layers
    - *Input layer:* does not perform any computational task
    - *Hidden (intermediate) layers* and *output layer*: composed by computational nodes (sigmoid functions)
  - Backpropagation Algorithm

Classification Methods

- Radial Basis Function
  - three layers with totally different roles
    - *Input layer*: doesn't perform computational task
    - *Second layer (hidden layer):* performs a non-linear transformation from the entry-space to a high-dimensional hidden-space
    - *Output layer*: linear and provides the network answer to an input signal

Combination Methods

- Decision Templates
- Dempster-Shafer

Decision Templates

- Continuous-valued outputs from each classifier with a different initialization for a given sample are used to build a decision profile (DP)
- The Decision Templates (DT) are the mean over all the decision profile from each training sample for each class

Decision profile for sample $\mathbf{x}$

Output for L classifiers and c classes

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

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

Decision template for class j

Number of elements from class j

sample

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

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

Similarity function between $DP(\mathbf{x})$ and DT_j
like Euclidean distance

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 the decision template and the output of each classifier.

$$\Phi_{j,i}(x) = \frac{\left(1 + \left\| DT_j^i - D_i(x) \right\|^2\right)^{-1}}{\sum_{k=1}^{c} \left(1 + \left\| DT_k^i - \underbrace{D_i(x)}_{i^{\text{th}} \text{ line of } DP(\mathbf{x})} \right\|^2\right)^{-1}}$$

Dempster-Shafer

- These are used to calculate the belief degree for every class.
- At last the final degrees of support of each test sample for each class are calculated from the belief degrees.

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

Belief degree for class *j* and classifier *i*

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

Degree of support for class *j*

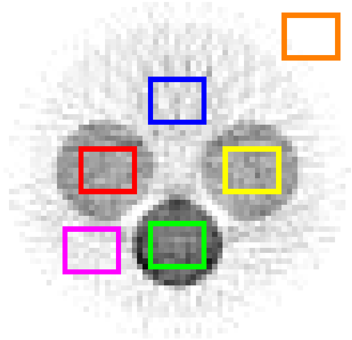
Normalization constant

Evaluation

- Hold-Out
  - Splits the set of available data in two halves:
    - Training set
    - Testing set
  - It's a fast testing scheme, as required by Neural Networks
- Kappa Coefficient
  - Measures the agreement rating between the classification of the test samples and their true class
    - $K = 1$ means full agreement
    - $K = 0$ means agreement is no higher than what is expected in a random classification

Experiments

- 480 samples (80 samples from each of the 6 class):
  - Aluminum
  - Water
  - Phosphorus
  - Calcium
  - Plexiglass
  - Background
- 240 samples (40 from each class) for training
- 240 samples for testing



40keV

60keV

85keV

662keV

Experiments

- MLP with 2 to 10 units in one single hidden layer
- RBF with 2 to 10 units in the hidden layer
- For the combination experiments we trained 10 different classifiers (changing the initialization parameters)
- Multilayer Perceptron is naturally unstable (it is sensitive to changes in the initialization parameters), so every experiment for all classification methods (the single ones and the combinations) were executed 100 times (changing the initialization parameters as well)

Results – Multilayer Perceptron

Estimated Error

Units	Single	DT	DS
2	0.5720	0.0349	0.0613
3	0.2689	0.0163	0.0275
4	0.1318	0.0141	0.0177
5	0.0976	**0.0123**	0.0151
6	0.0741	0.0127	0.0139
7	0.0681	0.0129	0.0138
8	0.0636	0.0130	**0.0137**
9	**0.0511**	0.0134	0.0138
10	0.0570	0.0135	0.0139

Kappa Coefficient

Units	Single	DT	DS
2	0,3137	0,9581	0,9265
3	0,6773	0,9805	0,9671
4	0,8419	0,9831	0,9788
5	0,8829	**0,9853**	0,9819
6	0,9111	0,9848	0,9833
7	0,9183	0,9845	0,9835
8	0,9237	0,9844	**0,9836**
9	**0,9387**	0,9840	0,9835
10	0,9316	0,9838	0,9833

Standard Deviation

Units	Single	DT	DS
2	0,2554	0,0286	0,0348
3	0,2892	0,0061	0,0133
4	0,2073	0,0044	0,0065
5	0,1504	0,0029	0,0046
6	0,1199	0,0020	0,0028
7	0,1065	**0,0013**	0,0024
8	0,0842	0,0014	**0,0020**
9	**0,0777**	0,0018	0,0021
10	0,0807	0,0018	0,0023

Results – Radial Basis Function

Estimated Error

Units	Single	DT	DS
2	0,5125	0,4458	0,4333
3	0,3958	0,3583	0,3583
4	0,2958	0,2583	0,2583
5	0,2292	0,2333	0,2333
6	0,1333	0,1292	0,1292
7	**0,0750**	**0,0708**	**0,0708**
8	0,0833	0,0833	0,0833
9	**0,0750**	0,0750	0,0750
10	0,0833	0,0875	0,0875

Kappa Coefficient

Units	Single	DT	DS
2	0,3850	0,4650	0,4800
3	0,5250	0,5700	0,5700
4	0,6450	0,6900	0,6900
5	0,7250	0,7200	0,7200
6	0,8400	0,8450	0,8450
7	**0,9100**	**0,9150**	**0,9150**
8	0,9000	0,9000	0,9000
9	**0,9100**	0,9100	0,9100
10	0,9000	0,8950	0,8950

Standard Deviation

Units	Single	DT	DS
2	0,0000	0,0000	0,0000
3	0,0000	0,0000	0,0000
4	0,0000	0,0000	0,0000
5	0,0000	0,0000	0,0000
6	0,0000	0,0000	0,0000
7	0,0000	0,0000	0,0000
8	0,0000	0,0000	0,0000
9	0,0000	0,0000	0,0000
10	0,0000	0,0000	0,0000

Conclusions

- Multilayer Perceptron
  - Using single classifier, results got better as we added units to the single layer
  - Using combination, the best results were achieved using few units on the hidden layer
  - Decision Templates and Dempster-Shafer show good results no matter how many units there is in the hidden layer

Conclusions

- Multilayer Perceptron
  - Differences between classifiers produced by different MLP initializations are enough to produce good combinations
  - Both methods showed improvements over the single classifier, but Decision Templates outperformed Dempster-Shafer with all the configurations, so we would highly recommend it for MLP-based classification systems

Conclusions

- Radial Basis Function
  - Both combining methods led to only slightly better classification
    - probably due to the more stable behavior of RBF
    - similar classifiers when we change only the initialization
    - classifiers do not differ from each other and it is difficult to obtain a good combination
  - DS method performed better than DT just in the experiments using 2 units in the hidden layer

Conclusions

- Neural Network based classifiers to identify materials on CT images is viable even when applied to images with high noise levels.
- The use of classifiers combiners led to better classification and more stable MLP systems, minimizing the effects of 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

Neural-Network Combination for Noisy Data Classification

Fabricio A. Breve 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