Evolutionary Learning using a Sensitivity-Accuracy Approach for Classification

Javier Sánchez Monedero[1], C. Hervás-Martínez[1], F.J. Martínez-Estudillo[2], Mariano Carbonero Ruz[2], M. C. Ramírez Moreno[2], M. Cruz-Ramírez[1]

i02samoj@uco.es

[1]Dept. of Computer Science and Numerical Analysis, University of Córdoba.
[2]Dept. of Management and Quantitative Methods, ETEA, Spain.

23rd June 2010

Introduction

How to evaluate a classifier?

Purpose of evaluation:

- To compare two classifiers performance
- To design fitness functions for evolutionary algorithms

Approaches:

- Accuracy (Traditional): one dimensional ordering
  - Global performance
  - "I can classify all the healthy people with 99% Accuracy... but 0% of the ill people class"
- Accuracy and Sensitivity: two dimensional ordering
  - Accuracy (C): Global performance
  - Sensitivity (S): worse classified class

Alternatives for measuring a classifier performance

Binary-classification problems

- Correct Classification Rate (CCR) $\rightarrow$ threshold metric
- Root mean square error (RMSE) $\rightarrow$ probabilistic metric
- Area under Curve (AUC) $\rightarrow$ range metric

Multi-classification problems

- Extension of AUC to multi-class: minimize the $Q(Q-1)$ misclassification rates $\rightarrow$hight computational cost

Accuracy and Sensitivity (I)

Classification problem

Lets consider a classification problem with Q classes and N training or testing patterns with a classifier g, the contingency or confusion matrix is:

$$M(g) = \left\{ n_{ij};\ \sum_{i,j=1}^{Q} n_{ij} = N \right\} \tag{1}$$

where n_{ij} represents the number of times the patterns are predicted by classifier g to be in class j when they really belong to class i.

Accuracy and Sensitivity (II)

Definitions

- the number of patterns associated with class i by $f_i = \sum_{j=1}^{Q} n_{ij}, \;\; i = 1, \ldots, Q$.
- Let $S_i = n_{ii}/f_i$ the number of patterns correctly predicted to be in class i with respect to the total number of patterns in i (sensitivity for class i).
- $S = \min\{S_i;\; i = 1, \ldots, Q\}$
- Correct Classification Rate or Accuracy, $C = (1/N)\sum_{j=1}^{Q} n_{jj}$

Accuracy and Sensitivity plot

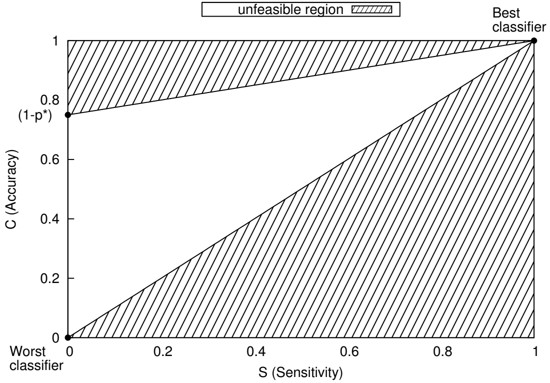


Figure: Unfeasible region in the two-dimensional space for a concrete classification problem.

Extreme Learning Machine for ANNs

ELM's [1] Three-Step Learning Model

Given a training set N samples $D = \{(\mathbf{x}_j, \mathbf{y}_j) : \mathbf{x}_j \in R^K, \mathbf{y}_j \in R^Q, j = 1, 2, \ldots, N\}$, where $\mathbf{x}_j$ is an $k \times 1$ input vector and $\mathbf{y}_j$ is an $Q \times 1$ target vector, activation function g and the number of hidden nodes L,

1. Assign randomly input weight vectors or centres a_i and hidden node bias or impact factor $b_i, i = 1, \ldots, L$
2. Calculate the hidden layer output matrix $\mathbf{H}$
3. Calculate the output weight $\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{Y}$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose (MP) generalized inverse of matrix $\mathbf{H}$

Extreme Learning Machine for ANNs

ELM's features

1. The learning speed of ELM is extremely fast.
2. The ELM tends to reach the solutions straightforward without problems such as local minima, improper learning rate and over fitting
3. ELM is a simple tuning-free three-step algorithm.

Disadvantages / Limitations

- ELM need a high number of hidden layer nodes
- Some improvements of ELM:
  - **Evolutionary ELM (E-ELM)** [2]
  - Optimally-Pruned ELM (OP-ELM) [3]

Evolutionary ELM (E-ELM)

Evolutionary ELM

1. Uses Storn's [1] differential evolutionary algorithm
2. The population is $\boldsymbol{\theta} = [\mathbf{w}_1, \ldots, \mathbf{w}_k, b_1, \ldots, b_k]$
3. E-ELM Evolves the input layer to hidden layer connection weights w_i
4. Uses original ELM for obtaining the *optimal* output weights $\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{Y}$
5. The misclassification rate $(\frac{1}{Accuracy})$ is used as fitness function

E-ELM considering C and S

Objective: to improve both the Accuracy and Sensitivity of the ANNs classifiers obtained by E-ELM

E-ELM considering C and S (E-ELM-CS)

Multi-objective optimization

- Multi-objective approach: not always C and S are cooperative objectives
- Linear multi-objective: efficient approach for multi-objective optimization

E-ELM-CS

- E-ELM-CS fitness function (to minimize):

$$\phi_\lambda = \frac{1}{(1-\lambda)C + \lambda S} \qquad (2)$$

- $\lambda \in [0, 1]$ is a user parameter obtained by experimental validation

C and S as competitive objectives

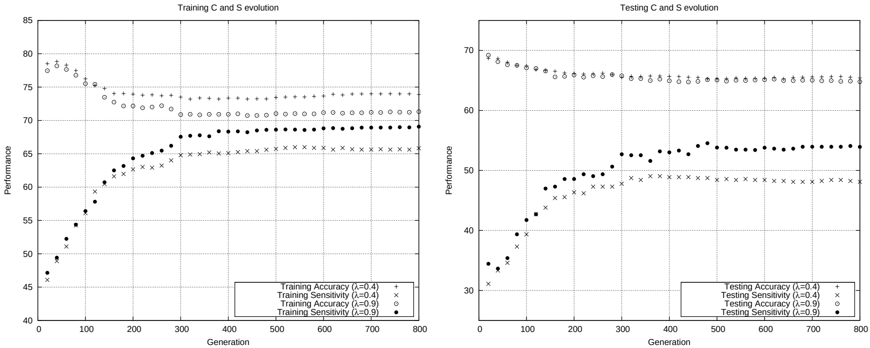
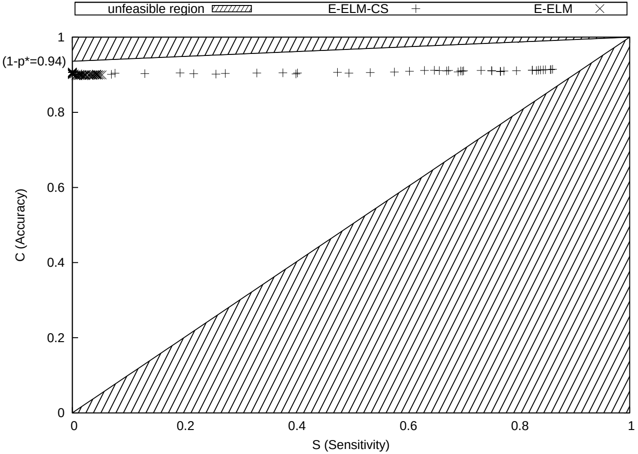


Figure: E-ELM-CS *C* and *S* evolution for BreastC database.

E-ELM vs. E-ELM-CS

Dataset	Algorithm	$C(\%)$ Mean±SD	$S(\%)$Mean±SD
BreastC	E-ELMCS$_{\lambda=0.4}$	**68.97±3.19**	**33.97±6.82**
	E-ELM	68.36±1.98	23.33±6.42
BreastCW	E-ELM-CS$_{\lambda=0.4}$	**96.32±0.86**	**93.87±2.28**
	E-ELM	95.68±1.19	92.61±3.21
Balance	E-ELM-CS$_{\lambda=0.7}$	**91.48**±1.50	**86.74±10.01**
	E-ELM	90.56±1.38	14.00±17.73
Gene	E-ELM-CS$_{\lambda=0.1}$	**83.72±1.93**	**81.10±2.94**
	E-ELM	83.48±1.90	78.89±4.97
Iris	E-ELM-CS$_{\lambda=0.9}$	**97.41±1.76**	**94.53±11.24**
	E-ELM	97.04±2.21	92.18±4.98
Newthy	E-ELM-CS$_{\lambda=0.9}$	**96.23±2.31**	**80.85±11.88**
	E-ELM	94.26±2.35	75.77±10.16

E-ELM vs. E-ELM-CS



E-ELM vs. E-ELM-CS

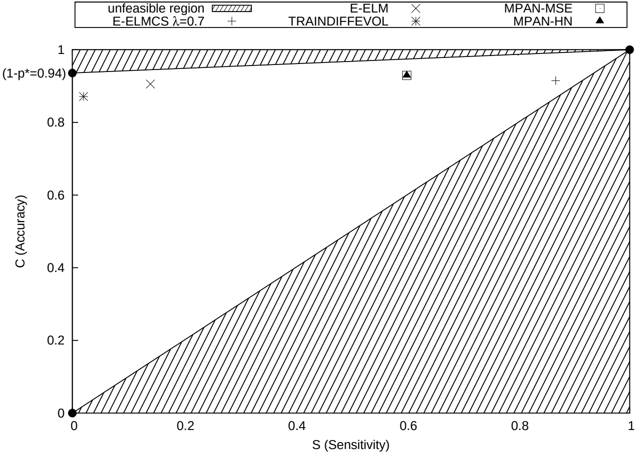


Figure: Comparison of E-ELM-CS, E-ELM, TDIF, MPAN-MSE and MPAN-HN methods for Balance database

E-ELM-CS Conclusions

- To consider the classifier training process as a multi-objective approach improves both C and S, but more significantly S
- S is improved for imbalanced databases
- Apparently, it is not clear with weight should be assigned to each objective. It heavily depends on the dataset

Future work

- Multiclassification problems with high number of classes and imbalanced datasets
- Look for efficient algorithms for building classifiers based on ANNs (¿extending E-ELM-CS?):
  - Optimally-Pruned ELM
  - Re-sampling, hibridation, etc.
  - Other neural networks types such as Product Unit, generalized Gaussian, qGaussian...
- ...

References

[1] Huang, G.B., Chen, L., Siew, C.K.:
Universal approximation using incremental constructive feedforward networks with random hidden nodes.
Neural Networks, IEEE Transactions on **17**(4) (July 2006) 879–892

[2] Zhu, Q.Y., Qin, A., Suganthan, P., Huang, G.B.:
Evolutionary extreme learning machine.
Pattern Recognition **38**(10) (2005) 1759 – 1763

[3] Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C., Lendasse, A.:
Op-elm: Optimally pruned extreme learning machine.
Neural Networks, IEEE Transactions on **21**(1) (December 2009) 158–162

References

[1] Storn, R., Price, K.:
Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces.
J. of Global Optimization **11**(4) (1997) 341–359

[2] Fernández, J., Martínez-Estudillo, F., Hervás, C., Gutiérrez, P.:
Sensitivity versus accuracy in multi-class problems using memetic pareto evolutionary neural networks.
IEEE Transacctions on Neural Networks **Accepted** (2010)

Evolutionary Learning using a Sensitivity-Accuracy Approach for Classification

Javier Sánchez Monedero[1], C. Hervás-Martínez[1], F.J. Martínez-Estudillo[2], Mariano Carbonero Ruz[2], M. C. Ramírez Moreno[2], M. Cruz-Ramírez[1]
i02samoj@uco.es

[1]Dept. of Computer Science and Numerical Analysis, University of Córdoba.
[2]Dept. of Management and Quantitative Methods, ETEA, Spain.

23rd June 2010