LEARNING HYBRID NEURO-FUZZY CLASSIFIER MODELS FROM DATA: TO COMBINE OR NOT TO COMBINE?

Bogdan Gabrys
Applied Computational Intelligence Research Unit
Division of Computing and Information Systems
University of Paisley, High Street, Paisley PA1 2BE, United Kingdom
Tel: +44 (0) 141 848 3752, Fax: +44 (0) 141 848 3542
E-mail: gabr-ci0@paisley.ac.uk

ABSTRACT: To combine or not to combine? Though not a question of the same gravity as the Shakespeare's to be or not to be, it is examined in this paper in the context of a hybrid neuro-fuzzy pattern classifier design process. A general fuzzy min-max neural network with its basic learning procedure is used within six different algorithm independent learning schemes. Various versions of cross-validation, resampling techniques and data editing approaches, leading to a generation of a single classifier or a multiple classifier system, are scrutinised and compared. The classification performance on unseen data, commonly used as a criterion for comparing different competing designs, is augmented by further four criteria attempting to capture various additional characteristics of classifier generation schemes. These include: the ability to estimate the true classification error rate, the classifier transparency, the computational complexity of the learning scheme and the potential for adaptation to changing environments and new classes of data. One of the main questions examined is whether and when to use a single classifier or a combination of a number of component classifiers within a multiple classifier system.

KEYWORDS: Neuro-fuzzy Classifier, Pattern Recognition, Ensembles of Classifiers, Resampling Techniques, Classifier Combination, Cross-validation.

INTRODUCTION

With an increasing computer power available at affordable prices and availability of vast amount of data there is an increasing need for robust methods and systems, which can take advantage of all available information. Automatic model building directly from data with a minimal or no human supervision is already absolutely crucial in order to stay competitive and maximally exploit the data in quickly changing business environments. However, the methodology for ensuring that created models (i.e. classifiers, predictors etc.) are as good as possible should be in place before using them with confidence.

No human supervision in model building also implies that one should use powerful enough techniques which can learn the data to any degree of accuracy. There are currently a lot of methods from soft computing, machine learning, and statistics domains which, in principal, satisfy this requirement. In the pattern recognition domain the examples include the nearest neighbour classifiers [18],[20], decision trees [18], neural networks with sufficient number of hidden nodes [4],[14],[18], fuzzy if then rules systems which are built directly from data [1]-[3],[15]-[17], neuro-fuzzy techniques based on hyperbox fuzzy sets [8]-[13], Bayesian networks or logical rule bases. From the statistical point of view most of these methods could be classified as non-parametric models. The main challenge in such cases is to design a model building strategy which would guard against overfitting of the training data or, in other words, would lead to a good generalisation performance.

Over the last few years, there has been a great amount of interest in the combination of the learning capability and computational efficiency of neural networks with the fuzzy sets ability to cope with uncertain or ambiguous data. This has led to a development of various hybrid neuro-fuzzy techniques [1]-[3],[15]-[17], including a general fuzzy min-max (GFMM) neural network for clustering and classification [8]-[13]. The development of the GFMM originated from our investigation into uncertain information processing in the context of the decision support for the operational control of industrial processes. This generic pattern recognition method based on hyperbox fuzzy sets combines supervised and unsupervised learning within a single learning algorithm, can grow to meet the demands of the problem, has the ability to incorporate new information without a need for complete retraining, learns on-line and has the ability to process inputs in the form of real (confidence) intervals. It has been successfully applied to a very challenging problem of leakage detection and identification in water distribution systems where a hierarchical system based on the GFMM has been used [12].

Since proposing the original GFMM neural network a number of extensions have been proposed in an attempt to produce a flexible pattern recognition framework which could accommodate various problems when generating models directly from high dimensional, real-world data. These extensions included a development of agglomerative learning algorithms for GFMM resulting in a generation of hierarchical structures [11], various approaches to dealing with missing data [8] and the use of statistical resampling techniques in the process of generating classifiers with good generalisation performance through: the use of data editing techniques [10], combining multiple copies of the GFMM classifier at the decision and model levels [9], and estimating the parameters controlling the complexity of the final GFMM classifier [11].

Though commonly the main objective of the classifier design process is constructing a classifier with as good a performance as possible, in many pattern classification applications there are some additional aspects which are regarded as equally important. For instance, it may be a part of the requirements that the classifier model is transparent and has the ability to provide an easily interpretable explanation of the suggested classification decision to a non-technical user. On the other hand, there may be applications where the speed of generation of the model is of primary concern or there are restrictions on the size of the classification model that can be stored. Yet another example could be an application where due to the non-stationary environment the emphasis should be put on the potential adaptability of the classifier model while in operation.

Bearing this in mind, in this paper we will concentrate on the analysis of various algorithm independent classifier model generation approaches for designing a GFMM classifier [8]-[13] or a GFMM based multiple classifier system. Various model generation approaches, which could be used together with the basic GFMM learning algorithms, will be assessed and discussed in the context of the following four criteria: a) the ability to estimate the performance of the model on unseen data; b) the generated model's power to explain the suggested decisions which can be interpreted by the user; c) the computational complexity involved in the classifier model building process; and d) the potential for adaptation of the classifier model to changing environments and new classes of data.

Specifically the above criteria will be applied to six GFMM classifier model building schemes including generating classifiers on the basis of the full training data set, using a k-fold and multiple 2-fold cross-validation procedures, data editing, various pruning approaches and combining multiple copies of the GFMM classifier.

Though in terms of pure classification performance the ensemble/combination methods [5]-[7],[9],[19] have been frequently shown to offer high classification performance gains in comparison to individual classifiers, when they are considered in the context of other criteria like the ones mentioned earlier the choice of the classifier generation scheme is no longer so clear. In this sense one of the main questions investigated will be that of whether to use a single model or a combination of a number of components forming the final classifier. As it will be illustrated each of the discussed model generation approaches has some advantages and disadvantages which make them more suitable for certain applications.

The remaining of this paper is organised as follows. The next section presents a summary of the GFMM neural network with definitions of hyperbox fuzzy sets and associated fuzzy membership function. It also provides a short description of one of the learning algorithms which can be used to place and adjust hyperboxes in the input space. In the following section the six different classifier generation schemes are discussed. This is followed by some simulation results illustrating their properties. And finally, conclusions are presented in the last section.

GENERAL FUZZY MIN-MAX NEURAL NETWORK DESCRIPTION

The GFMM neural network for classification constitutes a pattern recognition approach that is based on hyperbox fuzzy sets. A hyperbox defines a region of the *n*-dimensional pattern space, and all patterns contained within the hyperbox have full class membership. A hyperbox is completely defined by its min-point and its max-point. The combination of the min-max points and the hyperbox membership function defines a fuzzy set. Learning in the GFMM neural network for classification consists of creating and adjusting hyperboxes in the pattern space. Once the network is trained the input space is covered with hyperbox fuzzy sets. Individual hyperboxes representing the same class are aggregated to form a single fuzzy set class. Hyperboxes belonging to the same class are allowed to overlap while hyperboxes belonging to different classes are not allowed to overlap therefore avoiding the ambiguity of an input having full membership in more than one class. The input to the GFMM can be itself a hyperbox (thus representing features given in a form of upper and lower limits) and is defined as follows:

$$\boldsymbol{X}_h = [\boldsymbol{X}_h^l \;\; \boldsymbol{X}_h^u] \tag{1}$$

where $\boldsymbol{X}_h^l$ and $\boldsymbol{X}_h^u$ are the lower and the upper limit vectors for the *h*-th input pattern. Inputs are contained within the *n*-dimensional unit cube I^n. When $\boldsymbol{X}_h^l = \boldsymbol{X}_h^u$ the input represents a point in the pattern space.

The *j*-th hyperbox fuzzy set, B_j is defined as follows:

$$B_j = \{\boldsymbol{V}_j, \boldsymbol{W}_j, b_j(\boldsymbol{X}_h, \boldsymbol{V}_j, \boldsymbol{W}_j)\} \tag{2}$$

for all *j=1,2,...,m*, where $\boldsymbol{V}_j = (v_{j1}, v_{j2}, \ldots, v_{jn})$ is the min point for the *j*-th hyperbox, $\boldsymbol{W}_j = (w_{j1}, w_{j2}, \ldots, w_{jn})$ is the max point for the *j*-th hyperbox, and the membership function for the *j*-th hyperbox is:

$$b_j(\boldsymbol{X}_h) = \min_{i=1..n}(min([1 - f(x_{hi}^u - w_{ji}, \gamma_i)],[1 - f(v_{ji} - x_{hi}^l, \gamma_i)])) \tag{3}$$

where:

$f(x, \gamma) = \begin{cases} 1 & if \quad x\gamma > 1 \\ x\gamma & if \quad 0 \le x\gamma \le 1 \\ 0 & if \quad x\gamma < 0 \end{cases}$ - two parameter ramp threshold function; $\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_n]$ - sensitivity parameters governing how fast the membership values decrease; and $0 \le b_j(\boldsymbol{X}_h, \boldsymbol{V}_j, \boldsymbol{W}_j) \le 1$.

The membership values are used to decide whether the presented input pattern belongs to the class associated with the *j*-th hyperbox during the neural network operation stage.

The hyperbox membership values for each of the *p* classes are aggregated using the following formula:

$$c_k = \max_{j=1}^{m} b_j u_{jk} \tag{4}$$

where **U** is the binary matrix with values u_{jk} equal to 1 if the *j*-th hyperbox fuzzy set is a part of the *k*-th class and 0 otherwise; and $c_k \in [0, 1]$, *k=1..p*, represent the degrees of membership of the input pattern in the *k*-th class. A single winning class can be found by finding the maximum value of c_k .

AGGLOMERATIVE LEARNING ALGORITHM

The agglomerative learning for the GFMM [11] initializes the min $\boldsymbol{V}$ and max $\boldsymbol{W}$ matrices to the values of the training set patterns lower $\boldsymbol{X}^l$ and upper $\boldsymbol{X}^u$ limits respectively. The hyperboxes are then aggregated sequentially (one pair at a time) on the basis of the maximum similarity value calculated using a similarity measure adopted from the membership function (3). The hyperboxes with highest similarity value are only aggregated if:

a) newly formed hyperbox does not exceed the maximum allowable hyperbox size $0 \le \Theta \le 1$ and/or the hyperboxes similarity value is above certain minimum threshold value $0 \le s_{min} \le 1$; and

b) the aggregation does not result in an overlap with any of the hyperboxes representing other classes; and

c) the hyperboxes B_h and B_j form a part of the same class.

The above described process is repeated until there are no more hyperboxes that can be aggregated. For the formal detailed description of this process and definitions of various hyperbox similarity measures which can be used please refer to [11]. The on-line learning algorithm which could be used instead of the agglomerative version is described in [13].

ALGORITHM INDEPENDENT LEARNING APPROACHES

One of the most successful and commonly used approaches to estimating "true" errors and avoiding overfitting are based on various versions of cross-validation, bootstrap techniques and statistical resampling theory. While these resampling techniques have strong theoretical foundations they are also applicable to virtually any learning method. The resampling techniques have not only been used for error estimation but also for model building and improving classification performance. The description of six different model building schemes with the agglomerative algorithm used as a base learning algorithm follows.

GENERATING THE CLASSIFIER MODEL ON THE BASIS OF ALL TRAINING DATA WITHOUT PRUNING PROCEDURES

The simplest way to generate a GFMM classifier is to apply the agglomerative algorithm to the whole training set. After the training is completed the training set is learnt perfectly i.e. with the zero resubstitution error rate. However, the resubstitution error rate is a very poor estimate of the error which one can expect when testing on unseen data. Another problem with this approach is that the training data is very likely to be overfitted and a poor generalisation performance will ensue. Some of the generated hyperboxes can be overspecialised and representing only noisy data or outliers. This in turn can make the interpretation of the classification results slightly more difficult.

The advantage of this approach is that the model generation is very quick and does not require any additional procedures for hyperbox pruning. Since the GFMM can grow to accommodate new data, the adaptation which would result in creating new and adjusting the existing hyperboxes can be carried out using the base algorithm.

K-FOLD CROSS-VALIDATION WITH PRUNING PROCEDURES [8],[11]

In order to avoid overfitting and provide a better estimation of the generalisation error a k-fold cross validation procedure can be used. Apart from a better estimate of the true error the complexity of the GFMM classifier can be controlled through using a pruning procedure. The basic pruning procedure used in this paper removes hyperbox fuzzy sets which cause more misclassifications than correct classifications on the validation set. In this way a simpler classifier can be generated with larger hyperboxes which can be more easily interpreted.

The disadvantage of this method is that not all the training data are used for the classifier design and that one cannot say which of the k generated classifiers should be delivered as the final model. The model adaptation, as well as in the methods described below, can be problematic. On one hand, if the new data was to be included in the model immediately when it is presented, the on-line learning algorithm could be applied but the classifier would quickly degenerate to the case where no pruning was used. On the other hand, an evolving validation set accounting for new data could be used for periodical pruning and model updating.

MULTIPLE 2-FOLD CROSS-VALIDATION WITH PRUNING PROCEDURES [9],[11]

While k-fold cross validation provides good true error estimates for medium sized problems, repeated 2-fold cross-validation can provide yet more accurate true error estimates especially for small number of training samples. Additional advantage of using multiple 2-fold cross validation is the opportunity to estimate some parameters for controlling the complexity of the final model. In case of the GFMM such a parameter, which was estimated in this way, was the minimum cardinality (number of samples) of a hyperbox fuzzy set for which the hyperbox fuzzy set should be still retained in the final GFMM classifier model. Once the minimum cardinality is estimated the training is performed for the whole training set and hyperbox fuzzy sets representing a number of input patterns smaller than this minimum cardinality are pruned. The model generated is of the similar complexity as in the case of a single k-fold cross validation but all training data are used in the training process. The transparency of the model is retained but the adaptation to new data could be more difficult if the whole process of multiple cross-validation was to be repeated on a regular basis.

DATA EDITING BASED ON MULTIPLE 2-FOLD CROSS-VALIDATION [10]

A similar idea to the above which allows to use as many of the samples from the training set as possible but reducing a risk of overfitting the training set is based on the training data editing procedure. This approach is based on estimating the probability of every single sample in the original training set to be used in the generation of the hyperboxes during the multiple cross-validation. This probability is simply a ratio of the number of times a training sample $\boldsymbol{X}_h$ has been used in generation of a hyperbox which is retained in the classifier model after the pruning to the total number of repetitions of the 2-fold cross-validation. The training samples with small probability values are removed from the training set and the base learning algorithm is applied to the remaining training samples.

The properties of a GFMM classifier generated in this way are similar to the one discussed above.

ENSEMBLE OF CLASSIFIERS OBTAINED FROM REPEATED 2-FOLD SPLITTING OF THE TRAINING DATA SET AND AVERAGING THE OUTPUTS OF INDIVIDUAL CLASSIFIERS [11]

Even with a model complexity control in place, the variance component of the classification error for such highly flexible classifiers as the GFMM can be high. One way of reducing the variance and frequently improving the classifier performance is through “bagging” classifiers. Or in other words by combining the outputs of a number of classifiers generated during repeated 2-fold cross-validation. In case of the GFMM classifier the class support values c_k can be averaged. The improved classification performance of an ensemble classifiers comes at a cost of vastly increased complexity of the classification system and often increased time of achieving the classification results. The transparency of the classification decisions is also lost. The adaptation of the model to changing environments can be even more difficult since a number of copies of the GFMM classifier would have to be adapted at the same time.

COMBINATION OF INDIVIDUAL MODELS OBTAINED FROM REPEATED 2-FOLD SPLITTING OF THE TRAINING DATA SET [11]

An alternative to combining at the decision level (i.e. combining outputs of multiple copies of the GFMM classifier) is a combination at the model level (i.e. combining the hyperbox fuzzy sets from different copies of the GFMM). In case of GFMM it can be achieved in a straightforward manner just by using the hyperbox fuzzy sets from different models to be combined as inputs to the base learning algorithm. In this way the resulting GFMM classifier complexity and transparency is comparable with classifiers generated during a single cross-validation procedure while the improved classification performance and reduced variance is comparable to the ensemble of classifiers with combined decisions. This, however, comes at a cost of increased training time since additional training cycle is added. The adaptation of the model could be carried out as for any other single GFMM model but the benefits of the combination of the models could be quickly lost.

SIMULATION EXAMPLE

The above analysis of different GFMM classifier model generation schemes will now be illustrated on the basis of a 2-dimensional, non-trivial pattern recognition problem. The data set represents a normal mixtures data which has been introduced by Ripley [18]. The training data consists of 2 classes with 125 points in each class. Each of the two classes has bimodal distribution and the classes were chosen in such a way as to allow the best-possible error rate of about 8%. The training set and an independent testing set of 1000 samples drawn from the same distribution are available at http://www.stats.ox.ac.uk/~ripley/PRNN/.

A graphical illustration of the GFMM model for a single 2-fold cross validation procedure is shown in Figure 1. As we can see each of the hyperboxes covers a part of the input space which can represent a specific class of data. The hyperbox min-max points can be easily converted into a set of rules understandable for non-technical user. The classification boundaries created can be of arbitrary shape. The hyperboxes can be quite easily expanded and shrunk to accommodate new data.

The simulation results for each of the six model generation schemes are shown in Table 1. As we can see from the table the approaches which are based on multiple cross-validation offer significant improvement in the performance in comparison to the other model generation schemes. The performance improvement of the ensemble of classifiers (i.e. combination at the decision level) comes at a cost of vastly increased model complexity (i.e. number of hyperboxes in the final model) which dramatically increases with the increase of the component models combined. The model complexity of the classifier generated on the basis of the full training set without pruning is also significantly higher in comparison to the other schemes generating a single GFMM model. This is due to overfitting of the training data which is also reflected

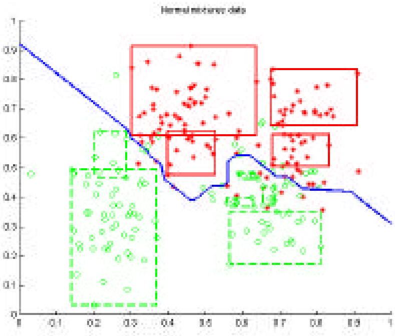

Figure 1: Normal mixtures data set: the training data, decision boundary, and the hyperboxes created during a single 2-fold cross-validation

Classifier generation procedure	Average no. of hyperboxes in the final model	Training set error rate [%]		Testing set error rate[%]	
		Mean error	Standard deviation	Mean error	Standard deviation
No pruning, training on the full data set	37	0	-	12.1	-
2-fold cross validation	6.78	13.48	1.54	9.64	1.03
Cardinality based pruning	10	11.6	-	8.2	-
Data editing	15	8	-	8.3	-
Combination at the decision level (40 combined classifiers)	270.25	13.2	0.57	8.55	0.25
Combination at the model level (40 combined classifiers)	15.75	9.8	1.48	8.15	0.24

Table 1: GFMM classification results for normal mixtures data set.

in the classification performance. On the other hand, the hyperbox cardinality based pruning when applied to the same model generated on the basis of the full training set can significantly improve the performance while reducing the model complexity.

CONCLUSIONS

Six different GFMM classifier model generation approaches have been presented and discussed. Depending on a classification problem each of the approaches can be shown to have some advantages over another. If the speed of model generation and quick adaptability to the changing environment are of primary concern then the base learning algorithm can be used without any hyperbox pruning procedures. On the other hand, if the model building can be carried out off line, the approaches based on multiple cross-validation either for estimating parameters controlling the complexity of the model or for building an ensemble of classifiers are likely to provide a better classification performance and the true error estimation. If the explanation of the classification decisions is to be provided, all the model generation schemes resulting in a single GFMM model can be used. This is with the exception of the ensemble of GFMM classifiers in which case the single GFMM model transparency is lost. While currently the advantages of the resampling techniques can be fully realised only during the initial model building process, an evolving validation set accounting for new data could be used for periodical model updating and hyperbox pruning. The frequency of such model validation would be dependent on the dynamics of a specific pattern classification problem.

REFERENCES

[1] Abe, S. and M.Lang, “A Method for Fuzzy Rules Extraction Directly from Numerical Data and Its Application to Pattern Classification”, *IEEE Trans. on Fuzzy Systems,* vol. 3, no. 1, pp.18-28, 1995

[2] Berthold, M., “Fuzzy Models and Potential Outliers”, in Proceedings of NAFIPS-99, pp. 532-535, IEEE Press, 1999

[3] Bezdek, J., J.Keller, R.Krisnapuram, N.R.Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing,* Kluwer Academic Publishers, 1999

[4] Bishop, C.M., *Neural Networks for Pattern Recognition*, Clarendon Press: Oxford, 1995

[5] Breiman, L., “Bagging predictors”, *Machine Learning*, vol. 24, no. 2, pp. 123-140, 1996

[6] Dietterich, T.G., “Machine Learning Research: Four Current Directions”, *AI Magazine*, vol. 18, no. 4, pp. 97-136, 1997

[7] Dietterich, T.G., “An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization”, *Machine Learning*, vol. 40, pp. 139-157, 2000

[8] Gabrys, B., “Neuro-Fuzzy Approach to Processing Inputs with Missing Values in Pattern Recognition Problems”, *submitted to Int. Journal of Approximate Reasoning*, 2001

[9] Gabrys, B., “Combining Neuro-Fuzzy Classifiers for Improved Generalisation and Reliability”, *submitted to the WCCI’2002 Congress*, 2001.

[10] Gabrys, B., “Data Editing for Neuro-Fuzzy Classifiers”, *Proceedings of the SOCO'2001 Conference*, Paisley, Scotland, 2001

[11] Gabrys, B., “Agglomerative Learning Algorithms for General Fuzzy Min-Max Neural Network”, accepted to the special issue of the *Journal of VLSI Signal Processing Systems*, 2001.

[12] Gabrys, B. and A.Bargiela, “Neural Networks Based Decision Support in Presence of Uncertainties”, *J. of Water Resources Planning and Management*, vol. 125, no. 5, pp.272-280, September/October 1999

[13] Gabrys, B. and A.Bargiela, “General Fuzzy Min-Max Neural Network for Clustering and Classification”, *IEEE Trans. on Neural Networks,* vol.11, no. 3, pp. 769-783, 2000

[14] Hassoun, M.H., *Fundamentals of artificial neural networks*, The MIT Press, 1995

[15] Kuncheva, L.I., *Fuzzy Classifier Design,* Physica-Verlag Heidelberg, 2000

[16] Nauck, D. and R. Kruse, "Learning in Neuro-Fuzzy Systems with Symbolic Attributes and Missing Values", *Proc. of International Conference on Neural Information Processing - ICONIP'99*, Perth, pp. 142-147, 1999.

[17] Nauck, D. and R.Kruse, “A neuro-fuzzy method to learn fuzzy classification rules from data”, *Fuzzy Sets and Systems*, vol. 89, no. 3, pp. 277-288, 1997

[18] Ripley, B.D. *Pattern Recognition and Neural Networks,* Cambridge University Press, 1996

[19] Ruta, D. and B.Gabrys, “An Overview of Classifier Fusion Methods”, *Computing and Information Systems* (Ed. Prof. M. Crowe), University of Paisley, vol. 7, no. 1, pp. 1-10, 2000

[20] Theodoridis, S. and K.Koutroumbas, *Pattern Recognition,* Academic Press, 1999