Theoretical and Experimental Analysis of a Two-Stage System for Classification

N. Giusti[2] F. Masulli[1] A. Sperduti[1]

(1) Dipartimento di Informatica - Università di Pisa

Corso Italia 40, 56125 Pisa, ITALY

E-mail: masulli@disi.unige.it, perso@di.unipi.it

(2) Micronix Computer

Via dei Colombi, 2 - 51016 Montecatini Terme (PT), ITALY

E-mail: ngiusti@micronix.net

Abstract

We consider a popular approach to multi-category classification tasks: a Two-Stage system based on a first (global) classifier with rejection followed by a (local) Nearest Neighbor classifier. Patterns which are not rejected by the first classifier are classified according to its output. Rejected patterns are passed to the Nearest Neighbor classifier together with the top-h ranking classes returned by the first classifier. The Nearest Neighbor classifier, looking at patterns in the top-h classes, classifies the rejected pattern. An editing strategy for the Nearest Neighbor reference database, controlled by the first classifier, is also considered.

We analyze this system, showing that even if the first level and Nearest Neighbor classifiers are not optimal in a Bayes sense, the system as a whole may be optimal. Moreover, we formally relate the response time of the system to the rejection rate of the first classifier and to the other system parameters. The error-response time trade-off is also discussed.

Finally, we experimentally study two instances of the system applied to the recognition of handwritten digits. In one system the first classifier is a Fuzzy Basis Functions Network, while in the second system it is a feed-forward Neural Network. Classification results as well as response times for different settings of the system parameters are reported for both systems.

index terms: multi-category classification, rejection, global and local classification, hierarchical classifier, Bayes classifier.

1 Introduction

In complex multi-category classification tasks it is widely used the approach where a multistage or hierarchical system is used in order to find the right trade-off between accuracy and resources allocation (e.g., response time, size of system, error costs). A typical example is a system where the set of classes is organized in a hierarchical way, and different classifiers are trained in order to drive the input pattern towards the most specific classifier to be used for the final classification (see, for example, [22, 31, 9, 24, 30]). Another example, is a system where only a subset of the input features, i.e., the less expensive to compute, are given as input to a first level classifier, and further input features are eventually used at further classification stages if the final classification cannot be performed with a sufficient level of confidence (e.g., reject-based approaches using an incremental set of input features [14, 23, 13, 31, 25]).

In this paper, we focus on a scheme where a first fast classifier with rejection is used to classify patterns with high confidence. Rejected patterns are forwarded to a more complex and slower second level classifier for a final classification (or further rejection). Typically, the system as a whole holds better classification performance with respect to the first classifier at the cost of a slower response time. Alternatively, improved classification performance can also be obtained by resorting to a committee of classifiers [5, 28, 10, 20, 26, 32, 21, 29, 7, 17, 16, 11, 27], however, the above hierarchical system turns out to be more flexible if constraints on the mean response time are imposed by the operating environment. In fact, by tuning the rejection criterion for the first classifier is possible to reach the best trade-off between error and response time.

Specifically, we study a Two-Stage system where rejected patterns are forwarded

to the Nearest Neighbor classifier together with the top-h ranking classes returned by the first classifier. Only patterns in the reference database belonging to the top-h classes are used by the Nearest Neighbor classifier to classify the rejected pattern. Moreover, to further speed-up the response time of the Nearest Neighbor classifier, an editing of the Nearest Neighbor reference database can be performed by collecting patterns rejected by the first level classifier according to a narrower rejection criterion.

It is worth noting that this type of system is consistent with a view of the classification process which tries to conciliate a global approach with a local one. In fact, while the aim of global estimation is to estimate a function for all possible values of input, it is typically very hard, especially for multi-classification problems, to get a good estimate for inputs which are very close to the decision boundary. For these inputs, it is more effective to perform a local estimation (see [2, 33]), which focuses on a specific estimation point. Of course, the natural choice for performing local estimation is to use memory-based methods, such as Nearest Neighbor.

The probability of error of the above system can be studied theoretically and it can be demonstrated that even if the two classifiers in the system are not optimal in a Bayes sense, the system as a whole, under specific conditions, may be optimal. Moreover, it is not difficult to formally relate the response time of the system to the rejection rate of the first classifier and to the other system parameters, and thus to discuss the error-response time trade-off.

We consider two specific instances of this scheme and we show that the obtained empirical results agree with the proposed framework, in the sense that the accuracy can be improved without significantly increase the average response time of the system.

The paper is organized as follows. The proposed two-stage classification system and its theoretical analysis (fully described in Appendix A) are presented in Section 2. Experimental results in this context are reported in Section 3, where two instances of the Two-Stage system are applied to handwritten digits recognition: in one instance the first level classifier is a Fuzzy Basis Functions Network (described in details in Appendix B), while in another instance a feed-forward Neural Network is used. The performances of the two systems for different instantiations of the parameters' values are computed, and the experimental analysis shows that the two-stage approach can actually reach a higher performance than the ones obtained by its single component classifiers. Moreover, the analysis on the response time of the system can be used to select the different settings of parameters with the same response time, allowing the user to easily select the one which holds the best classification capability. Section 4 reports the conclusions.

2 A Two-Stage System for Multi-Category Classification Tasks

Let us to consider a multi-category classification framework, where an input pattern is represented as a random n-dimensional feature vector $\boldsymbol{X}$ belonging to one of N classes, $\omega_1, \ldots, \omega_N$. In this context we define a two-stage pattern recognition scheme consisting of a hierarchy made up by a multi-category (global) classifier $\mathcal{C}$ with rejection, followed by a Nearest-Neighbor Rule (NR) classifier working on the patterns rejected by $\mathcal{C}$. Specifically, a rejection rule is implemented as a *rejection threshold* on the level of the higher output, i.e., if no output of $\mathcal{C}$ is greater than the threshold, the pattern is rejected from $\mathcal{C}$. The basic idea is that patterns rejected

The Two-Stage System

input: first level classifier $\mathcal{C}$, rejection threshold γ, NR database threshold λ, number of classes h to use for NR, reference database T;

Compute $T(\lambda)$, the subset of T containing patterns $\boldsymbol{Y}$ rejected by $\mathcal{C}$ using threshold λ, i.e., $max_j\ \boldsymbol{o}_j(\boldsymbol{Y}) \leq \lambda$;

repeat

Read input pattern $\boldsymbol{X}$;

Compute $\boldsymbol{o}(\boldsymbol{X})$, i.e., the output of $\mathcal{C}$ for $\boldsymbol{X}$;

if $max_j\ \boldsymbol{o}_j(\boldsymbol{X}) > \gamma$ **then** return ω_q, where $q = arg\ max_j\ \boldsymbol{o}_j(\boldsymbol{X})$;

else

Compute $Cl(h, \boldsymbol{X})$, the set of the first h classes with higher $\boldsymbol{o}_j(\boldsymbol{X})$;
Compute $T(\lambda)_h$, the subset of patterns in $T(\lambda)$ and class in $Cl(h, \boldsymbol{X})$;
Return for $\boldsymbol{X}$ the class computed by NR($T(\lambda)_h$,$\boldsymbol{X}$);

Figure 1: The algorithm for the Two-Stage System.

by the rejection rule with threshold value γ are then classified by the Nearest-Neighbor Rule with reference database made up by patterns rejected by the same classifier $\mathcal{C}$, but using a threshold value $\lambda \geq \gamma$. By using the recognition threshold γ, $\mathcal{C}$ classifies very fast most of the patterns with small classification error, while a minority of patterns are forwarded to the NR for classification. Moreover, the quality of the NR database is controlled independently by the threshold λ. Of course, for rejected patterns, the recognition speed depends mainly on the dimension of the NR database. However, to speed-up the recognition time of the NR classifier, at classification time one can use an efficient *on-line editing strategy*, by considering for the NR only patterns belonging to classes that get the first h higher outputs by $\mathcal{C}$. The Two-Stage algorithm is defined in Figure 1.

Let us denote the Two-Stage system with H. It is not difficult to show (see Appendix A) that the probability of error $\boldsymbol{e}_{H,h}(\gamma)$ for the Two-Stage system, given a rejection threshold γ for the classifier at the first level, and selecting the top-h ranking classes for the second level classifier, can be expressed as the sum of four positive terms

$$\boldsymbol{e}_{H,h}(\gamma) = \boldsymbol{e}_{bayes} + \boldsymbol{e}_{\mathcal{C}}(\gamma) + \boldsymbol{e}_{NR}(h) + \boldsymbol{e}_{top-h}, \tag{1}$$

where $\boldsymbol{e}_{bayes}$ is the optimal Bayes error over the input domain, $\boldsymbol{e}_{\mathcal{C}}(\gamma)$ is the error rate of the $\mathcal{C}$ classifier for the given value of the rejection threshold γ, $\boldsymbol{e}_{NR}(h)$ is the error induced by the application of the Nearest Neighbor Rule on the patterns rejected by $\mathcal{C}$ and depending on the value of h, and $\boldsymbol{e}_{top-h}$ is the error due the fact that the right class is not included (by $\mathcal{C}$) in the top-h classes. Notice that $\boldsymbol{e}_{top-h}$ is null if $h = N$. Moreover, under this condition, the Two-Stage system may reach the optimal Bayes error if and only if all the patterns which are not rejected by $\mathcal{C}$ are classified correctly while the rejected patterns are classified correctly by the Nearest Neighbor Rule. Of course, there is no guarantee that the Two-Stage system will reach the optimal Bayes error, however, the above decomposition of the error shows that, in principle, we do not loose the possibility to reach the optimal Bayes error even if both $\mathcal{C}$ and NR are not optimal.

Finally, it should be noted that only under special conditions, i.e., using the k-Nearest-Neighbor Rule with large values for k and an infinite amount of training examples, the optimal Bayes error can be reached: the NR by itself may only achive asymptotic error rate which is sub-optimal (only about twice the Bayes error [6]). However, the use of a different second level classifier turns out to be quite problematic. In fact, just training a global classifier on patterns rejected by the

first level classifier does not lead to good performance, since the complexity of the classification problem is exactly "coded" by the rejected patterns. This can be better understood when considering the support vectors of a (kernel) Support Vector Machine [33]: the support vectors are the patterns closest to the decision boundary and training from scratch the classifier by just keeping these patterns will not change the outcome of learning.

Following this reasoning, it is clear that only a local classifier may be able to get a good performance. Given a testing pattern, an alternative to the use of NR would be to train a classifier with the training examples located in a small neighborhood around the testing pattern and then to apply the trained classifier to the testing pattern itself [2]. This approach, however, would excessively increase the response time of the system. Alternatively, a predefined decomposition of the input space in regions could be used to train once for all local classifiers assigned to each region [13]. This solution, however, could be inadequate since the a priori decomposition could turn out to be sub-optimal.

2.1 Error-Response Time Trade-off

For some applications, the response time of a system may be as much important as the performance in generalization. The Two-Stage system is flexible enough to allow the balance of these two aspects by tuning the rejection threshold γ.

First of all, note that the classifier at level 1 has a constant response time, independently of the pattern in input. Let C_1 be the computational cost of running it on a pattern. On the other hand, the response time of the Nearest Neighbor classifier using the top-h ranking selection rule depends on the number of prototypes stored for each of the top-h ranking classes. Let $\xi_i^h(\gamma) = P(\omega_i \in Cl(h)|\gamma)$, i.e., the

probability of class ω_i to be in the top-h ranking classes given a rejection threshold γ, then the expected number of prototypes $P_{2,h}(\gamma)$ used by the Nearest Neighbor classifier is

$$P_{2,h}(\gamma) = h \sum_{i=1}^{N} c_i(\lambda) \xi_i^h(\gamma), \tag{2}$$

where $c_i(\lambda)$ is the number of prototypes of class ω_i in the reference database obtained by λ. Note that, if the prototypes are balanced across classes, i.e., $\forall i \; c_i(\lambda) = c_\lambda$, then $P_{2,h}(\gamma) = hc_\lambda$. The computational cost $C_{2,h}(\gamma)$ can then be defined as

$$C_{2,h}(\gamma) = \mathcal{I}(P_{2,h}(\gamma)), \tag{3}$$

where $\mathcal{I}(\cdot)$ is a function which depends on the way the Nearest Neighbor classifier is implemented. Finally, the response time $C_{H,h}(\gamma)$ for the Two-Stage system is

$$C_H(\gamma) = C_1 + r(\gamma) C_{2,h}(\gamma), \tag{4}$$

where

$$r(\gamma) = \int_R p(\boldsymbol{X}) d\boldsymbol{X}. \tag{5}$$

Note that, in general, a local classifier needs much more time to respond than a global classifier. In our setting, the response time for the classifier at the second level can be reduced by using small values for both h and λ. This reduction in response time, however, is paid with a loss in generalization. Thus, the problem is to find a good trade-off between $C_H(\gamma)$ and $\boldsymbol{e}_{H,h}(\gamma)$. Since the relationship between the different components and parameters of the Two-Stage system are not linear, in this paper we have studied this problem from an experimental point of view.

3 Experimental Results

As test-bed for our experiments we used the classification of handwritten digits. Specifically, we used a training set, a validation set, and a test set extracted from the NIST-3 data-base [8]. Both the training set and the validation set were made up of 10,000 associative pairs of segmented handwritten digits each, obtained from disjoint groups of writers, while the test set consisted of 5,000 independent digits.

The preprocessing of the digits included the following steps:

1. digit image extraction from the CD-ROM and normalization to a 32×32 binary matrix;

2. low-pass filtering in order to remove some small spots and holes from the image;

3. application of a shear transform to the digit image to straighten the axis joining the first upper-left point of the digit image to the last lower-right point;

4. image skeletonization by using a thinning algorithm;

5. finally, transformation of the digit representation into a 64-element vector, each vector element representing the number of black pixels contained in adjacent 4×4 squares (local counting).

It is worth noting that the resulting digit representation exhibits sufficient degrees of invariance to both scale and small image shifts or rotations.

3.1 Performance

The hardness of the classification problem was evaluated by studying the performance of the k-Nearest-Neighbor Rule (k-NR) for different values of k (see Figure 2). Due to sparseness of data, the best performance was obtained for $k = 1$ (92.89%). In order to have a more extensive comparison against other classification techniques based on supervised training, we report in Table 1 the performance of the best Fuzzy Basis Functions Network (FBFN) (see [3] and Appendix B for details on the model) with 48, 12, and 10 hidden nodes we were able to obtain on the data. It must be pointed out that the FBFN holds universal approximation capabilities and, under the same training conditions we adopted, it can approximate the Bayes optimal classifier [18]. Moreover, all the Multilayer Neural Networks (NNs) we were able to train showed a slightly inferior performance with respect to FBFNs with a similar number of free parameters. As reference for NN we will use a network with one hidden layer with 48 hidden units (NN_{48}), which achieved a performance of 93.42% on the test set.

In Table 1, we have also reported the results obtained on a simplified version of the FBFN (see Appendix B for details), namely a ESFBFN with 20 hidden units. This network can be trained in less than half the time required by the $FBFN_{48}$ as well as its response time is less than half of the $FBFN_{48}$, while still preserving a similar performance in generalization.

In order to avoid over-fitting, the training for all the networks (including the neural network) was stopped using the validation set (early stopping).

Due to the better trade-off between response time and generalization performance, among the neuro-fuzzy models we selected the $ESFBFN_{20}$ as classifier at the first level of the first instance of the Two-Stage system. We also considered

a system where the first classifier was the neural network NN_{48} with normalized output. Moreover, accordingly with the results reported in Figure 2, we decided to use $1 - NR$ as second level classifier for both systems.

In Figure 3, we have reported the distribution for classes of the reference database for the NR ($k = 1$) (to be used in the Two-Stage system) generated by $ESFBFN_{20}$ for different values of λ, while the effect of the rejection threshold on the training and test set for the $ESFBFN_{20}$ is shown in Figure 4. Similar curves are obtained for NN_{48}.

Let us now turn to the performance of the whole Two-Stage systems. In Figure 5 we have reported the test curves for $\gamma = 0.96$ and different values of λ for both systems. As expected the performance of both systems improved with the dimension of the reference database for the NR (higher values of λ). Mixed results are instead obtained when considering the number of classes used for the NR (on-line editing): the system based on NN_{48} got the worst performance with just 2 classes, while the system based on $ESFBFN_{20}$ got in this case its best performance. Thus, it seems that NN_{48} is not able to get a very good estimate of the a posteriori probability of the first two best classes.

To assess the "best" values for γ and h (i.e., the number of classes to be used by

MODEL	%S-Test	Epochs	Epochs Duration (sec)	Training Cost
$FBFN_{48}$	94.09	13	1925	25025
$FBFN_{12}$	92.26	36	307	11052
$FBFN_{10}$	92.23	55	180	9900
$ESFBFN_{20}$	93.80	250	49	12250

Table 1: Comparison among FBF networks (with 48, 12, and 10 hidden units), and a ESFBF network with 20 hidden units (2 for each class). %S-Test is the success rate on the test set. The epoch duration is measured on a Sun 10.

the NR) we performed experiments with $\lambda = 1.0$ and different values for γ and h for both systems. The results obtained for the test set for both systems are reported in Figure 6. On the sampled values for the parameters, the system based on NN_{48} got the lowest error (5.08%) with $\gamma = 0.68$ and $h = 5$, while the system based on ESFBFN_{20} got the lowest error (4.98%) with $\gamma = 0.88$ and $h = 2$.

In Figure 7, we have reported the performance on the test set of both the NR and NN_{48} on the set of patterns rejected by NN_{48} for different values of γ. From these curves it is clear that the NR outperforms NN_{48} on patterns which are close to the decision boudary of NN_{48}. Similar results are obtained for ESFBFN_{20}.

3.2 Response Time

In order to study the response time of the system we focus on the Two-Stage system where the first classifier is ESFBFN_{20}. For this system, we have estimated the quantities defined in Eq. (4), i.e., C_1, $C_{2,h}$, and $r(\gamma)$. Both C_1 and $C_{2,h}$ have been estimated by considering the most relevant mathematical operations performed by $\mathcal{C}$ and NR. We have considered additions, subtractions, multiplications, divisions, and the computation of the exponential function. The cost of performing 100 millions of each operation on a SUN 20 workstation has been computed experimentally. We obtained the same cost (let say unitary cost) for addition, subtraction, and multiplication, while the division costed 2.208 and the computation of the exponential function 6.305. Using these weights we estimated the computational cost for computing the output of $\mathcal{C}$ and NR[1]. The rejection rate $r(\gamma)$ was experimentally computed over different values of γ close to the optimum value for the validation set, i.e., $\gamma = 0.87$, $\gamma = 0.88$, and $\gamma = 0.89$. The estimation of $C_H(\gamma)$ for different values of h and λ

[1]No optimized algorithm for the NR has been considered.

are reported in Figure 8. As expected, as soon as h and λ increase, $C_H(\gamma)$ increases exponentially. This is particularly true with the increase of h. Moreover, the rate of increase is controlled by the value of γ. For each plot in Figure 8 we have drawn on the cost surface curves with constant cost (i.e., $C_H(\gamma) = 100 \cdot 10^3$, $C_H(\gamma) = 200 \cdot 10^3$, and $C_H(\gamma) = 290 \cdot 10^3$). This was done to show that different couples of values for λ and h may end up to have the same computational cost and thus they have the same average response time.

If a bound on the average response time is given, these plots can be used to select the values of γ, λ, and h which are consistent with the target response time. Furthermore, the setting corresponding to the best performance can be chosen for the working system. In Figure 9 we give an example of how this selection can be done. We have chosen a cost of 10^5, which is used as cut point for the surfaces $C_H(\gamma)$ computed for sampled values of γ. The level curves obtained in this way are projected on the $\lambda - h$ plane and the performance of the Two-Stage system is evaluated on admissible points of the curves (i.e. $\lambda \geq \gamma$ and integer values for h) by using the validation set (Figure 9-(a)). The values for γ, λ, and h corresponding to the best performace (in this case, $\gamma = 0.8$, $\lambda = 0.988$, $h = 3$) are then used in the working system. From Figure 9-(b), it can be noted that the selected values, as well as the performance, are not far from the optimal values for the test set, i.e., $\gamma = 0.85$, $\lambda = 0.983$, $h = 2$.

4 Conclusion

In the context of multi-category classification tasks we have considered a Two-Stage system which combines a first level global classifier with the Nearest-Neighbor Rule.

This system is consistent with a view of the classification process which tries to conciliate a global approach with a local one in order to improve the trade-off between classification accuracy and response time. This trade-off is an important issue when considering classification tasks involving a high number of different classes.

For the proposed scheme it is possible to theoretically relate the error rate of the system with the optimal Bayes error, showing that it is actually possible to reach the optimal Bayes error even if the two classifiers in the system are not optimal in a Bayes sense. Moreover, we formally related the expected average response time of the system to the rejection rate of the first classifier and to the other system parameters. This allows the tuning of the system parameters in order to reach the desired error-response time trade-off.

On two specific instances of the system and on a specific classification task, we have demonstrated that the classification accuracy of the system can actually be higher than any of the single compounding classifiers. The system expected average response time is within[2] the range defined by the two compounding classifiers and it can be adjusted by tuning the system parameters.

We believe that for many other different instances of the system and many different classification tasks, a similar behavior, as predicted by the theoretical analysis, may be reproduced.

Acknowledgments

This work was supported by grants from CNR, INFM, and MURST. We thank Stefano Rovetta for helpful suggestions.

[2]Actually, it is closer to the extreme defined by the first level classifier.

A Theoretical Analysis of the Two-Stage System

In this appendix, we perform a theoretical analysis of the Two-Stage system error, showing in detail the decomposition of the error reported in Eq. (1).

Let π_{ω_i} to denote the *a priori* probability of observing class ω_i, while the *posterior* probability of class ω_i given an input vector $\boldsymbol{X}$ is denoted as

$$P(\omega_i \mid \boldsymbol{X}) = \frac{p(\boldsymbol{X} \mid \omega_i)\pi_{\omega_i}}{p(\boldsymbol{X})}, \tag{6}$$

where $p(\boldsymbol{X} \mid \omega_i)$ is the ω_i class conditional probability density function, and $p(\boldsymbol{X})$ is the input vector probability density function.

For our aims, it is important to sort the posterior probabilities $P(\omega_i \mid \boldsymbol{X})$ in decreasing order. With

$$P_1(\boldsymbol{X}) \geq P_2(\boldsymbol{X}) \geq \ldots \geq P_N(\boldsymbol{X}) \tag{7}$$

we denote the ordered sequence of posterior probabilities, where

$$P_1(\boldsymbol{X}) = max_{j \in [1,\ldots,N]} P(\omega_j \mid \boldsymbol{X}), \tag{8}$$

and so on. In the Two-Stage system, the output $\boldsymbol{o}(\boldsymbol{X})$ of the classifier $\mathcal{C}$ is actually providing an approximation of the posterior probabilities. However, since $\boldsymbol{o}(\boldsymbol{X})$ is only an approximation of the true posterior probabilities, the order induced by $\boldsymbol{o}(\boldsymbol{X})$ over the classes will, in general, be different from the one induced by the true posterior probabilities. Consequently, let

$$F_1(\boldsymbol{X}), F_2(\boldsymbol{X}), \ldots, F_N(\boldsymbol{X}) \tag{9}$$

be sequence (7) reordered according to $\boldsymbol{o}(\boldsymbol{X})$, i.e.,

$$F_1(\boldsymbol{X}) = P(\omega_k \mid \boldsymbol{X}) \tag{10}$$

with

$$k = argmax_{j\in[1,\ldots,N]} o_j(\boldsymbol{X}), \tag{11}$$

and so on. We can now define the discrepancy values induced by the classifier $\mathcal{C}$ as

$$\Delta^F(\boldsymbol{X}) = P_1(\boldsymbol{X}) - F_1(\boldsymbol{X}). \tag{12}$$

The same definitions can similarly be devised for the classifier at the second level of the Two-Stage system (Nearest Neighbor):

$$G_1(\boldsymbol{X}), G_2(\boldsymbol{X}), \ldots, G_N(\boldsymbol{X}), \tag{13}$$

and

$$\Delta^G(\boldsymbol{X}) = P_1(\boldsymbol{X}) - G_1(\boldsymbol{X}). \tag{14}$$

Note that $\Delta^F(\boldsymbol{X}) \geq 0$ and $\Delta^G(\boldsymbol{X}) \geq 0$.

Let us denote the Two-Stage system with H. The probability of error $\boldsymbol{e}_{H,h}(\gamma)$ for the Two-Stage system, when using a rejection threshold γ for the classifier at the first level, and selecting the top-h ranking classes for the second level classifier, can be expressed as

$$\boldsymbol{e}_{H,h}(\gamma) = \boldsymbol{e}_1(\gamma) + \boldsymbol{e}_{2,top-h}(\gamma), \tag{15}$$

where $\boldsymbol{e}_1(\gamma)$ is the error rate of the $\mathcal{C}$ classifier, and $\boldsymbol{e}_{2,top-h}(\gamma)$ is the error induced by the application of the second level classifier selecting only the top-h ranking classes

suggested by the $\mathcal{C}$ classifier.

The first level error rate can be written as

$$\boldsymbol{e}_1(\gamma) = \int_A (1 - F_1(\boldsymbol{X}))p(\boldsymbol{X})d\boldsymbol{X}, \tag{16}$$

where A is the input subspace accepted for classification by the classifier, i.e., $A = \{\boldsymbol{X} \mid max_{j\in[1,\ldots,N]} o_j(\boldsymbol{X}) \geq \gamma\}$. By using the discrepancy values, $\boldsymbol{e}_1(\gamma)$ can be cast in the following form

$$\begin{aligned} \boldsymbol{e}_1(\gamma) &= \int_A (1 - F_1(\boldsymbol{X}) + P_1(\boldsymbol{X}) - P_1(\boldsymbol{X}))p(\boldsymbol{X})d\boldsymbol{X} & (17)\\ &= \int_A (1 - P_1(\boldsymbol{X}))p(\boldsymbol{X})d\boldsymbol{X} + \int_{A\cap I^F} \Delta^F(\boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X}, & (18) \end{aligned}$$

where $I^F = \{\boldsymbol{X} \mid \Delta^F(\boldsymbol{X}) \neq 0\}$. On the other side, $\boldsymbol{e}_{2,top-h}(\gamma)$ can be written as

$$\boldsymbol{e}_{2,top-h}(\gamma) = \int_R (1 - G_1(\boldsymbol{X}) \sum_{j=1}^{h} F_j(\boldsymbol{X}))p(\boldsymbol{X})d\boldsymbol{X}, \tag{19}$$

where R is the input subspace rejected by the first level classifier, i.e., $R = \{\boldsymbol{X} \mid max_{j\in[1,\ldots,N]} o_j(\boldsymbol{X}) < \gamma\}$. After some algebra it turns out that

$$\begin{aligned} \boldsymbol{e}_{2,top-h}(\gamma) &= \int_R (1 - P_1(\boldsymbol{X}))p(\boldsymbol{X})d\boldsymbol{X} + \int_{R\cap I^G} \Delta^G(\boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X} & (20)\\ &\quad + \int_R G_1(\boldsymbol{X}) \sum_{j=h+1}^{N} F_j(\boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X}, \end{aligned}$$

where $I^G = \{\boldsymbol{X} \mid \Delta^G(\boldsymbol{X}) \neq 0\}$.

Thus, the error rate for the Two-Stage system can be written as

$$\boldsymbol{e}_{H,h}(\gamma) = \boldsymbol{e}_{bayes} + \int_{A\cap I^F} \Delta^F(\boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X} + \int_{R\cap I^G} \Delta^G(\boldsymbol{X})p(\boldsymbol{X})d\boldsymbol{X} \tag{21}$$

$$+\int_R G_1(\boldsymbol{X}) \sum_{j=h+1}^{N} F_j(\boldsymbol{X}) p(\boldsymbol{X}) d\boldsymbol{X},$$

where $\boldsymbol{e}_{bayes}$ is the optimal Bayes error over the input domain, and all the remaining terms are non-negative. Note that, in order for the Two-Stage system to reach the optimal Bayes error, we must have $h = N$, $A \cap I^F = \emptyset$, and $R \cap I^G = \emptyset$. So, even if the classifiers at the first and second level are not optimal in the Bayes sense[3], the Two-Stage system can still approach the optimal error rate. What is important is that the classifier at the first level must misclassify only vectors in R, while the classifier at the second level must misclassify only vectors in A.

Of course, it is not reasonable to expect these kind of behavior from the classifiers in the Two-Stage system. However, it must be noted that the error of the classifier at the first level decreases with the increase of γ. In fact, when considering a bayesian classifier, the adopted rejection rule[4] guarantees that it is possible to express the error rate directly as a function of the reject rate via the following Stieltjes integral [4]

$$\boldsymbol{e}_1(\bar{t}) = -\int_0^{\bar{t}} t d\rho(t), \tag{22}$$

where $\gamma = 1 - t$,

$$\rho(t) = \int_{R_{bayes}} p(\boldsymbol{X}) d\boldsymbol{X}, \tag{23}$$

and $R_{bayes} = \{\boldsymbol{X} \mid max_{j\in[1,\ldots,N]} P_j(\boldsymbol{X}) < 1 - t\}$

The classifier at first level will have, in general, a small error probability, which means that the set $A \cap I^F$ is small in size. Moreover, the pattern rejected by it will

[3]Note that I^F or I^G are not required to be empty.

[4]The rejection rule we adopted corresponds to the Chow rule [4]. Chow has shown that this rule is optimal in the sense that for a given error rate (error probability) the rejection rate (reject probability) is minimized.

be located, in general, close to the boundaries of the classes. As a consequence, the rejected patterns are expected to be rather sparse, and a further classification by using a global classifier is not going to return a satisfactory classification. In order to correctly classify these patterns, it is more productive to use a local classifier [33], such as the Nearest Neighbor classifier. Because of the above considerations, the classifier $\mathcal{C}$ will tend to minimize the term

$$\boldsymbol{e}_{\mathcal{C}}(\gamma) = \int_{A \cap I^F} \Delta^F(\boldsymbol{X}) p(\boldsymbol{X}) d\boldsymbol{X}, \tag{24}$$

while the classifier at the second level, because of its locality, should reduce considerably the term

$$\boldsymbol{e}_{NR}(h) = \int_{R \cap I^G} \Delta^G(\boldsymbol{X}) p(\boldsymbol{X}) d\boldsymbol{X}. \tag{25}$$

The remaining term

$$\boldsymbol{e}_{top-h} = \int_R G_1(\boldsymbol{X}) \sum_{j=h+1}^{N} F_j(\boldsymbol{X}) p(\boldsymbol{X}) d\boldsymbol{X}, \tag{26}$$

is minimized by setting $h = N$, however, for speeding up the response time of the system, it may be convenient to have $h < N$.

B The Extended Simplified Fuzzy Basis Function Network

A Fuzzy Logic System with *singleton* fuzzification, *max-product* composition, *product inference* and *height* defuzzification can be represented as [19]

$$y = f(\mathbf{x}) = \sum_{l=1}^{M} \overline{y}^{\,l} \phi_l(\mathbf{x}) \tag{27}$$

with

$$\phi_l(\mathbf{x}) = \frac{\prod_{i=1}^{p} \mu_{F_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{p} \mu_{F_i^l}(x_i)} \tag{28}$$

where $\overline{y}^{\,l}$ denotes the center of gravity of the output fuzzy set, and $\phi_l(\mathbf{x})$, $l = 1, 2, ..., M$, are called *fuzzy basis functions*. We can refer to those FLS as *fuzzy basis expansions* or *networks of fuzzy basis functions* (FBF network)[5].

The relationships between fuzzy basis expansions and other basis functions have been extensively studied in [12]. It is worth noting that the FLS with universal function property studied by Mendel and Wang [35, 34] (i.e., a singleton FLS using product inference, product implication, Gaussian membership and height defuzzification) can be rewritten as a FBF network expansion.

Here we are interested in a neuro-fuzzy logic system based on a multi-input-multi-output (MIMO) version of this FBF network. Specifically, if there are K units in the input layer, J fuzzy inference rules and I outputs, the rule activations can be expressed as:

$$r_j = \prod_k \mu_{jk}(x_k) \tag{29}$$

[5]In [19] fuzzy basis expansions for FLS with non-singleton fuzzification are also introduced.

$$\mu_{jk}(x_k) = \exp\left(-\frac{(x_k - m_{jk})^2}{2\sigma_{jk}^2}\right) \tag{30}$$

$$y_i = \frac{\sum_j r_j \overline{y}_{ij}}{\sum_j r_j} = \sum_j \overline{y}_{ij}\phi_j(\mathbf{x}) \tag{31}$$

$$\phi_j = \frac{\prod_k \mu_{jk}(x_k)}{\sum_j \prod_k \mu_{jk}(x_k)} \tag{32}$$

where the quantity $\mu_{jk}(x_k)$ represents the value of the membership function of the component x_k of the input vector for the jth rule, m_{jk} and σ_{jk}^2 are the means and the variances of the Gaussian membership functions, y_i are the values of the output units, $\overline{y}_{ij}$ is the center of gravity of the output fuzzy membership function of the jth rule associated with the output y_i, and ϕ_j is the fuzzy basis function associated to rule j, representing its normalized activation [6].

The FBF network can be regarded as a fuzzy system mapped on a network of RBF. The FBF network can be identified both by exploiting the linguistic knowledge available (*structure identification problem*)[15] and by using the information contained in a data set (*parameter estimation problem*) [15]. Learning rules based on Gradient Descent technique are discussed, e.g., in [35].

For pattern recognition applications, from this FBF network a *Simplified FBF network* (SFBF network) can be obtained by assuming, in accordance with *rule specialization* [1]:

$$\overline{y}_{ij} \equiv \delta_{ij} = \begin{cases} 1 & \text{if rule } j \text{ is associated to class } i, \\ 0 & \text{otherwise.} \end{cases} \tag{33}$$

This assumption leads to both a system with as many units as classes and a strong

[6] Without loss of generality, we could assume that the fuzzy membership functions are singletons ($\overline{y}_{ij} \equiv s_{ij}$).

simplification of the learning formulas, that become:

$$\Delta m_{jk} = \eta_m \phi_j U_{ij} [x_k - m_{jk}] / \sigma_{jk}^2 \tag{34}$$

$$\Delta \sigma_{jk} = \eta_\sigma \phi_j U_{ij} [x_k - m_{jk}]^2 / \sigma_{jk}^3 \tag{35}$$

with

$$U_{ij} = \begin{cases} (y_i - 1)^2 & \text{if } j = i \\ y_I^G - y_i & \text{if } j \neq i \end{cases} \tag{36}$$

It is worth noting that, from Eq. (33) and the form of the defuzzifier, $y \in (0, 1)$ follows, and consequently

$$U_{ij} = \begin{cases} \geq 0 & \text{if } j = i \\ \leq 0 & \text{if } j \neq i \end{cases} \tag{37}$$

holds.

Therefore, the learning rules of the SFBF network are competitive. During training, the means of the Gaussian membership functions of each rule move towards the patterns of the class associated to that rule, and escape from patterns belonging to other classes. At the same time, sigmas of Gaussian membership functions of each rule grow in order to increase the value of the membership function for patterns of the class associated to that rule, or shrink in order to reduce the value of the Gaussian membership function for patterns belonging to other classes. From a probabilistic poit of view, the SFBF can be seen as a mixture of Gaussians with diagonal covariance matrices.

Of course, since this system must have as many units as classification classes,

it cannot be used for complex classification tasks. To remove this constraint, a new level of competition among units can be introduced. The new defined network possesses n_j units associated to each class j, for a total of $J = \sum_{j=1}^{I} n_j$ units. During learning the output of each unit is computed and the best unit for each class is selected, i.e., for each class j the unit $i_j^* \in Idx_j = \{1, \ldots, n_j\}$ such that $i_j^* = arg \underset{i \in Idx_j}{max}\{\phi_i\}$ is selected. In that way the number of selected units is equal to the number of classes and the learning rules of the SFBF network can be applied. Thus, at each learning step, only the selected rules have the weights changed. During the operational phase, the input pattern is classified by the class label associated with the unit having maximum activity.

References

[1] D. Alfonso, F. Masulli, and A. Sperduti. Competitive learning in a classifier based on an adaptive fuzzy system. In P.G. Anderson and K. Warwick, editors, *Proceedings of the International ICSC Symposium on Industrial Intelligent Automation (IIA'96) and Soft Computing (SOCO'96), Reading, England*, pages C2–C8, Millet Alberta, Canada, 1996. ICSC.

[2] L. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–901, 1992.

[3] F. Casalino, F. Masulli, and A. Sperduti. Rule specialization in networks of fuzzy basis functions. *Intelligent Automation and Soft Computing*, 4:73–82, 1998.

[4] C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16:41–46, 1970.

[5] W. Bruce Croft and Leah S. Larkey. Combining classifiers in text categorization. In *SIGIR*, pages 289–297, 1996.

[6] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[7] Cesare Furlanello, Diego Giuliani, Edmondo Trentin, and Stefano Merler. Speaker normalization and model selection of combined neural networks. *Connection Science, Vol. 9, No. 1*, pages 31–50, 1997.

[8] M.D. Garris and R.A. Wilkinson. *NIST Special Database3 Handwritten Segmented Characters.* National Institute of Standard and Technology, Gaithesburg, MD , USA, 1992.

[9] S. Gutta and H. Wechsler. Gender classification of human faces using hybrid classifier systems. In *International Conference on Neural Networks*, volume 3, pages 1353–1358. IEEE, New York, 1997.

[10] Sherif Hashem. Effects of collinearity on combining neural networks. *Connection Science, Vol 8, No. 3 & 4*, pages 315–336, 1996.

[11] Daniel Jimenez and Nicolas Walsh. Dynamically weighted ensemble neural networks for classification. In *IJCNN Alaska*, pages 753–758, 1998.

[12] H.M. Kim and J.M. Mendel. Fuzzy basis functions: Comparisons with other basis functions. *IEEE Trans. on Fuzzy Systems*, 3:158–168, 1995.

[13] S. Knerr and A. Sperduti. Rejection driven hierarchy of neural network classifiers. In *International Symposium on Nonlinear Theory and its Applications '93*, pages 957–961, 1993. Waikiki, USA.

[14] M. W. Kurzynski. On the multistage bayes classifier. *Pattern Recognition*, 22(4):355–365, 1988.

[15] C.C. Lee. Fuzzy logic in control systems: fuzzy logic controller. I. *IEEE Transactions on Systems, Man and Cybernetics*, 20:404–418, 1990.

[16] Yong Liu and Xin Yao. A cooperative ensemble learning system. In *IJCNN*, pages 2202–2207, 1998.

[17] Richard Maclin and David Opitz. An empirical evaluation of bagging and boosting. In *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI-97/IAAI-97)*, pages 546–551, Menlo Park, July 27–31 1997. AAAI Press.

[18] F. Masulli, F. Casalino, and F. Vannucci. Bayesian properties and performances of adaptive fuzzy systems in pattern recognition problems. In M. Marinaro and P.G. Morasso, editors, *Proceedings of the European Conference on Artificial Neural Networks, ICANN-94*, pages 189–192, Sorrento, Italy, 1994. Springer.

[19] J.M. Mendel. Fuzzy logic systems for engineering: A tutorial. *Proceedings of the IEEE*, 83:345–377, 1995.

[20] David W. Opitz and Jude W. Shavlik. Actively searching for an effective neural network ensemble. *Connection Science, Vol 8, No. 3 & 4*, 1996.

[21] Bambang Parmanto, Paul W. Munro, and Howard R. Doyle. Reducing variance of committee prediction with resampling techniques. *Connection Science, Vol 8, No. 3 & 4*, 1996.

[22] P. Poddar and P.V.S. Rao. Hierarchical ensemble of neural networks. In *International Conference on Neural Networks*, volume 1, pages 287–292. IEEE, New York, 1993.

[23] P. Pudil, J. Novovicova, S. Blaha, and J. Kittler. Multistage pattern recognition with reject option. In *11th IAPR International Conference on Pattern Recognition*, volume 2, pages 92–95. IEEE, New York, 1992.

[24] C. Rodriguez, J. Muguerza, M. Navarro, A. Zarate, J.I. Martin, and J.M. Perez. A two-stage classifier for broken and blurred digits in forms. In *Fourteenth International Conference on Pattern Recognition*, volume 2, pages 1101 –1105. IEEE, New York, 1998.

[25] J. Rokui and H. Shimodaira. Multistage building learning based on misclassification measure. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 221–226. IEE, 1999.

[26] Bruce E. Rosen. Ensemble learning using decorrelated neural networks. *Connection Science, Vol 8, No. 3 & 4*, pages 373–384, 1996.

[27] Robert F. Schapire and Yoram Singer. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT-98)*, pages 80–91, New York, July 24–26 1998. ACM Press.

[28] Amanda J. C. Sharkey. On combining artificial neural nets. *Connection Science, Vol 8, No. 3 & 4*, pages 299–314, 1996.

[29] Amanda J. C. Sharkey. Modularity, combining and artificial neural nets. *Connection Science, Vol. 9, No. 1*, pages 3–10, 1997.

[30] S. Simon, H.A. Kestler, A. Baune, F. Schwenker, and G. Palm. Object classification with simple visual attention and a hierarchical neural network for subsymbolic-symbolic coupling. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 244–249. IEEE, New York, 1999.

[31] S.K. Tso, X.P. Gu, Q.Y. Zeng, and K.L. Lo. Input space decomposition and multi-level classification approach for ANN-based transient security assessment. In *Fourth*

International Conference on Advances in Power System Control, Operation and Management, volume 2, pages 499–504, 1997.

[32] Kagan Tumer and Joydeep Ghosh. Error correlation and error reduction in ensemble classifiers. *Connection Science, Vol 8, No. 3 & 4*, 1996.

[33] V.N. Vapnik. *Statistical Learning Theory.* Wiley & Sons, 1998.

[34] L. Wang and J.M. Mendel. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Trans. on Neural Networks*, 5:807–14, 1992.

[35] L. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Trans. on Systems, Man, and Cybernetics*, 22:1414–1427, 1992.

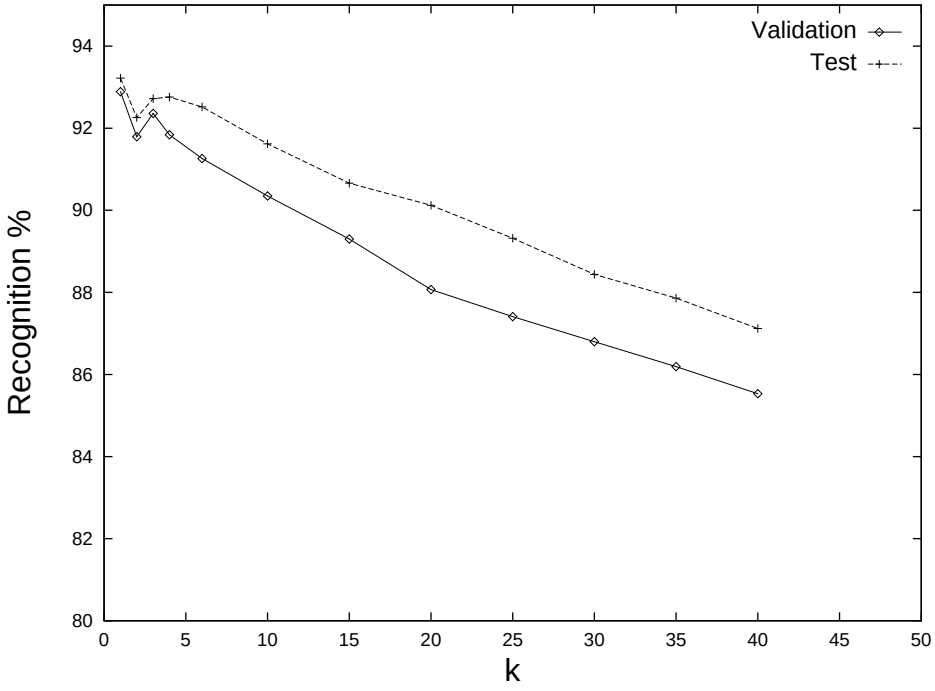


Figure 2: Performance of the k-Nearest-Neighbor on the test and validation sets using as reference database the full training set.

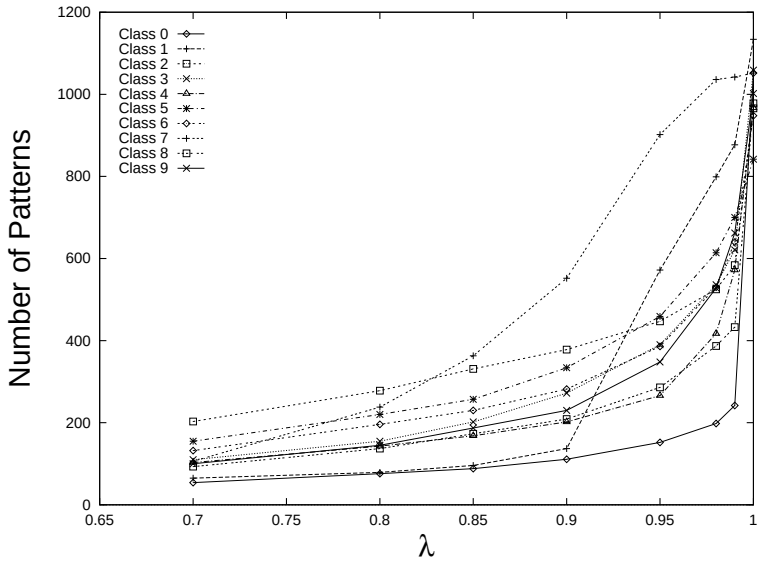


Figure 3: Composition of the reference database for the NR ($k = 1$) for different classes and λ values by using ESFBFN$_{20}$. Note that the database for $\lambda = 1.0$ is equal to the training set.

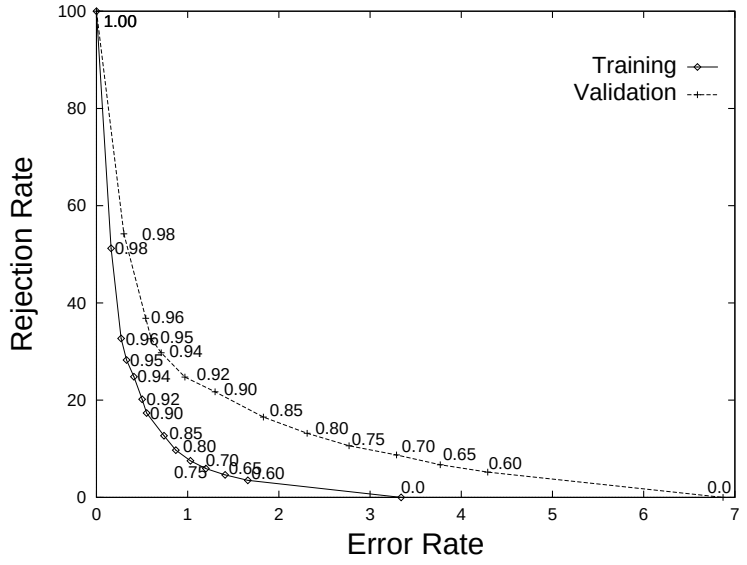
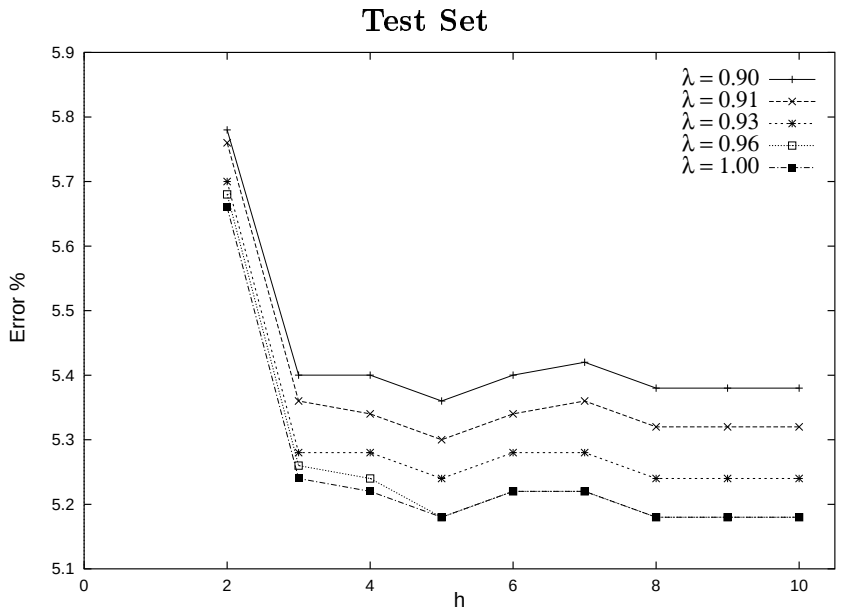
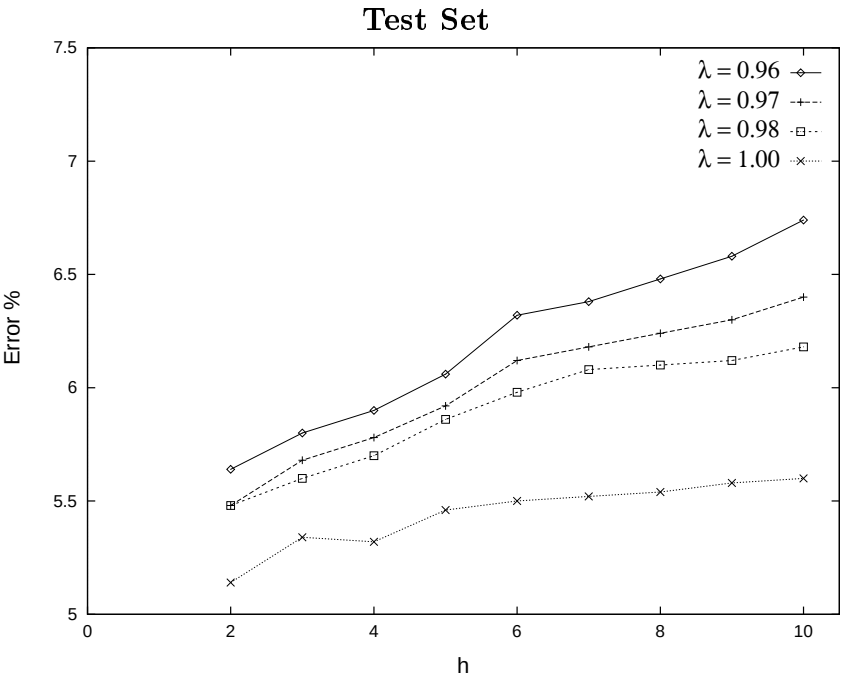


Figure 4: Effect of the rejection threshold (reported near each experimental point) on the training and validation set for the ESFBFN$_{20}$. The numbers shown on the curves are the values of the γ threshold for which that error/rejection value was obtained.

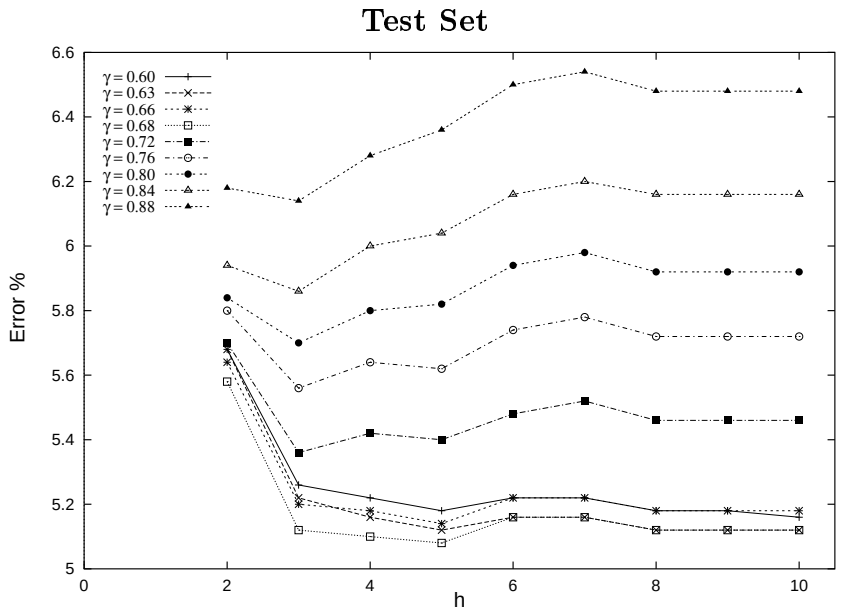


(a)

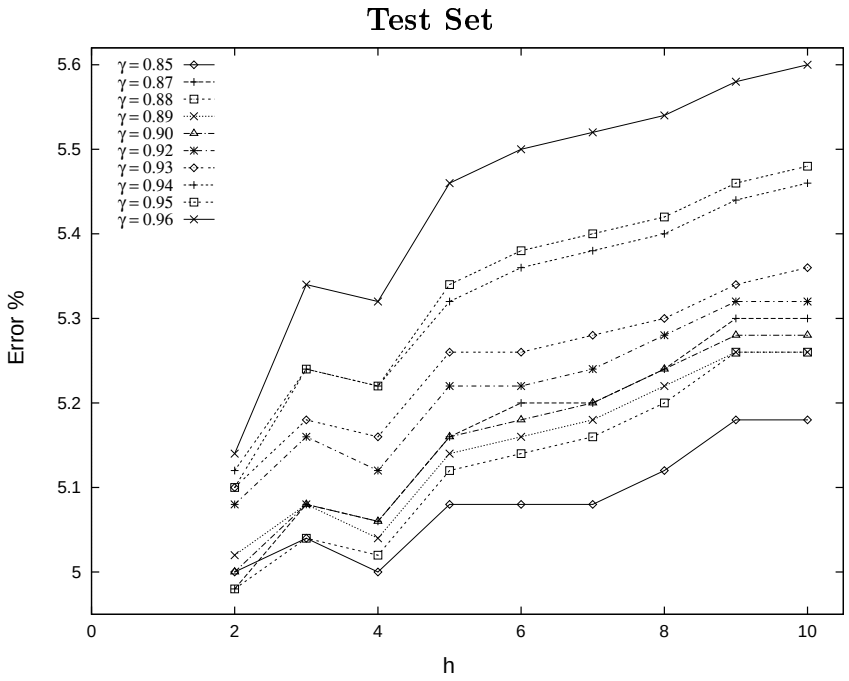


(b)

Figure 5: Error curves for two different instances of the Two-Stage System for a fixed value of γ and different values of λ. (a): the first classifier is a NN with 48 hidden units ($\gamma = 0.65$); (b): the first classifier is a ESFBFN with 20 (hidden) units ($\gamma = 0.96$).



(a)



(b)

Figure 6: Error curves for two different instances of the Two-Stage System for $\lambda = 1.0$ and different values of γ. (a): the first classifier is a NN with 48 hidden units; (b): the first classifier is a ESFBFN with 20 (hidden) units.

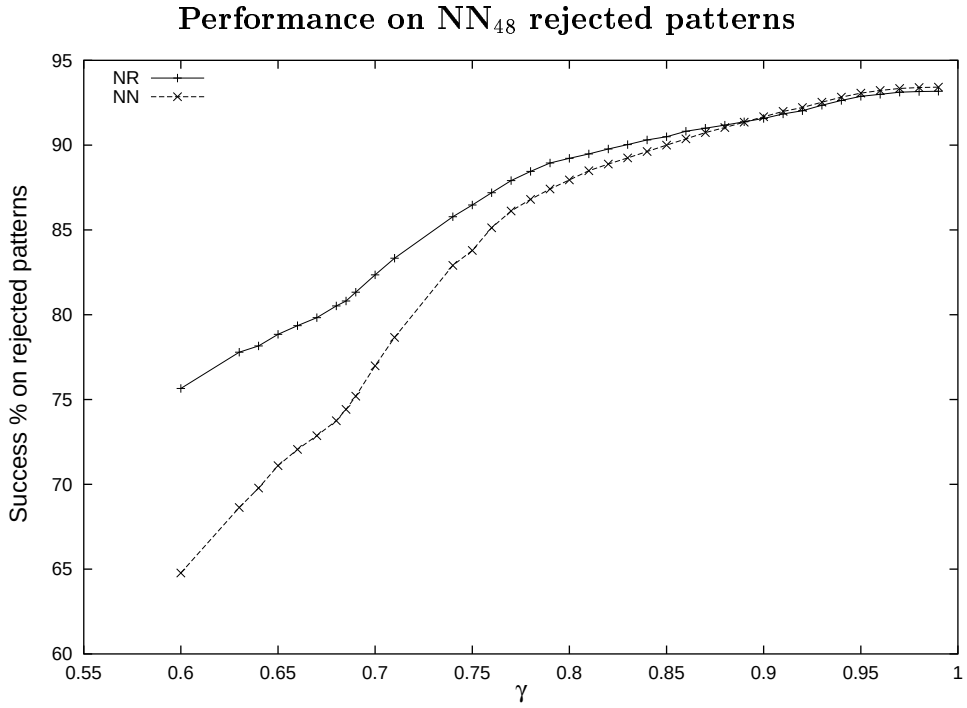


Figure 7: Performance of the NR (with full trainig set) and NN_{48} on the patterns rejected by the NN_{48} for different values of γ.

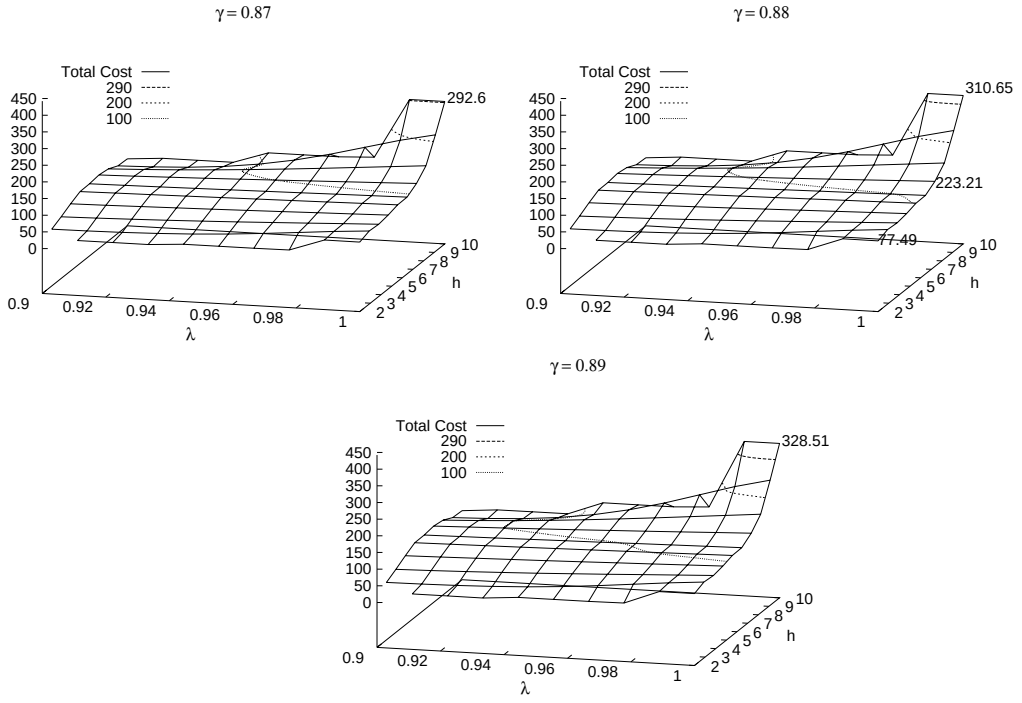
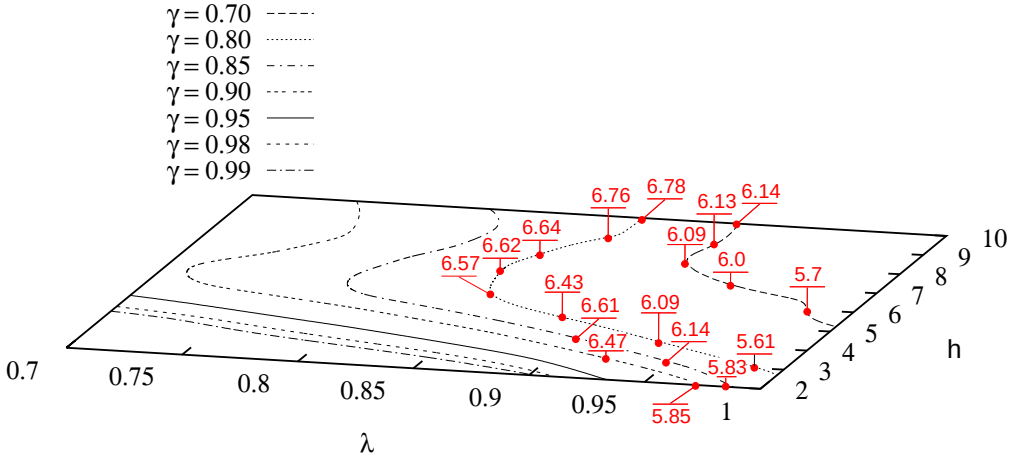
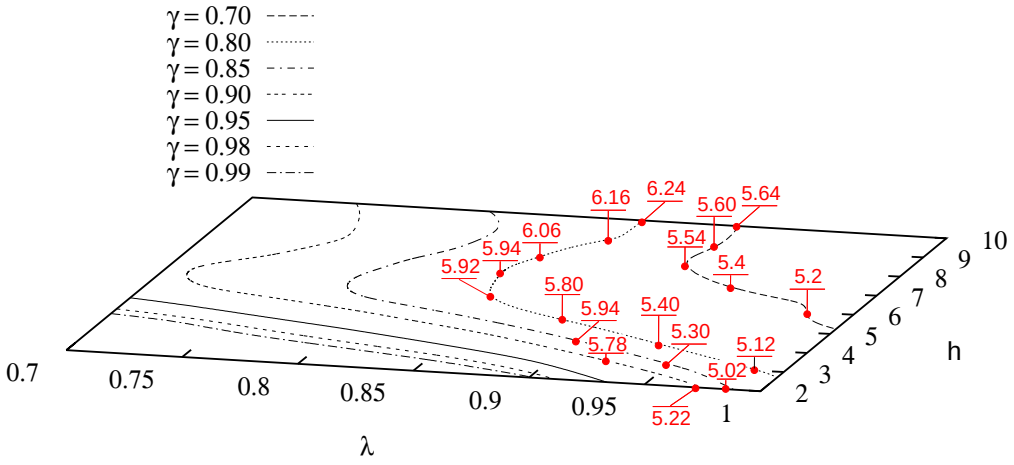


Figure 8: Plots representing an estimate of $C_H(\gamma)$ for different values of γ. $C_H(\gamma)$ is plotted for different values of h and λ. Costs must be multiplied for 10^3.



(*a*)



(*b*)

Figure 9: Level curves of the estimated $C_H(\gamma)$ for different values of γ given a fixed cost of 10^5. The performance obtained by a Two-Stage System with a ESFBFN with 20 (hidden) units on (*a*) the validation set, and (*b*) the test set, for admissible points on the curves (i.e. $\lambda \geq \gamma$ and integer values for h), are shown. Given a cost (e.g. 10^5), the corresponding plot on the validation set (e.g. (*a*)) can be used to decide the best setting for λ and h (for this example, $\gamma = 0.8$, $\lambda = 0.988$, $h = 3$, which is almost optimal with respect to the test set: see plot (*b*)). Notice that plot (*b*) is consistent with plot (*a*).