2012 International Conference on Frontiers in Handwriting Recognition

Benchmarking of Update Learning Strategies on Digit Classifier Systems

D. Barbuzzi, D. Impedovo, G. Pirlo
Dipartimento di Informatica
Università degli Studi di Bari "Aldo Moro"
Bari, Italy
{d.impedovo, pirlo}@di.uniba.it

***Abstract*—Three different strategies in order to re-train classifiers, when new labeled data become available, are presented in a multi-expert scenario. The first method is the use of the entire new dataset. The second one is related to the consideration that each single classifier is able to select new samples starting from those on which it performs a miss-classification. Finally, by inspecting the multi expert system behavior, a sample misclassified by an expert, is used to update that classifier only if it produces a miss-classification by the ensemble of classifiers. This paper provides a comparison of three approaches under different conditions on two state of the art classifiers (SVM and Naive Bayes) by taking into account four different combination techniques. Experiments have been performed by considering the CEDAR (handwritten digit) database. It is shown how results depend by the amount of the new training samples, as well as by the specific combination decision schema and by classifiers in the ensemble.**

Keywords- Feedback learning, Multi Expert, Training Sample Selection

I. Introduction

A pattern recognition system consists of two main processes: enrollment (or training) and matching (or recognition). In the first phase, samples of specific classes are acquired, processed and features extracted. These features are labeled with the ground truth and used to generate the model representing the class. Matching mode performs the recognition of the (unknown) input pattern by comparing it to the enrolled templates. Depending by the specific scenario, a single classifier is not always able to gain acceptable or high performance, so that in many applications [1, 2, 4, 6] classifiers combination is a suitable solution.

On the other hand, as the specific scenario evolves, new labeled data can became available. In these situations, the typical issue is the way in which the new data should be used. Recently, it has been showed, that in cases where a Multi Expert system (ME) is adopted, the collective behavior of classifiers can be used to select the most profitable samples in order to update the knowledge base of classifiers [15, 18]. More specifically samples, to be used for re-training, are selected by considering those, misclassified by a specific expert of the set, which produces a misclassification at the ME level. This approach moves from the consideration that the collective behavior of a set of classifiers can convey more information than those of each classifier of the set, and this information can be exploited for classification aims [4, 5, 17].

This paper reports of a comparison of this approach to situation in which the entire new dataset is used for learning as well as the case in which specific samples are selected by the individual classifier. Tests have been performed on the task of handwritten digit recognition, on the CEDAR database, by considering different type of features, two state of the art classifiers (Support Vector Machine and a Naive Bayes classifier), and four different combination techniques (Majority Vote, Weighted Majority vote, Sum Rule and Product Rule). It is shown how results depend by the specific classifier, by the combination decision schema, as well as by training/test data distribution.

The paper is organized as follows: Section II presents the background of re-training and the different strategies. Experimental setup and results are, respectively, in Section III and IV. Section V reports conclusions of the work.

II. Learning Updating Strategies

A. Background

Template update is an interesting and open issue both in the case of supervised and semi-supervised learning.

Let us consider the scenario in which new labeled data become available. The question is: how to use new data? The simplest way, to update the knowledge base of the classifier, is probably to use the entire new set to retrain the system given the initial training condition or, depending by the classifier, the set of new+old data. On the other hand, many interesting algorithms can be adopted in order to select specific samples. Among the others, AdaBoost [7, 9] is able to improve performance of a classifier on a given data set by focusing the learner attention on difficult instances. Even if this approach is very powerful, it works well in the case of weak classifiers, moreover not all the learning algorithms accept weights for the incoming samples. Another interesting approach is the bagging one: a

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.186

number of weak classifiers trained on different subset (random instance) of the entire dataset are combined by means of the simple majority voting [8]. Unfortunately bagging and AdaBoost are designed and work well in the case of weak classifiers and, on the other hand, if applied to a ME system, they do not take into account the behavior of different classifiers in the ensemble, in fact they are applied considering a single classifier in a stand-alone modality. From this point of view, the first intent of this work is to deal with state of the art performing classifiers (not weak) and to determine strategies which can be applied whatever classifier is considered.

Let us consider the case of new un-labeled data, two well known approaches, used in order to select specific samples, are self-training and co-training. These are semi-supervised learning paradigms (the updating process is performed using both the initial set of labeled templates and a set of unlabelled data acquired during the on-line operation). Self-training (or self-update) [13] is based on the concept that a classifier is retrained on its own most confident output produced from unlabeled data. The classifier is at first trained on the set of labeled data and, subsequently, several self-training iterations are performed to incorporate the unlabeled data until some stop rule is satisfied. Co-training (or co-update) [12] is the situation under which two classifiers improve each other. More specifically the first expert is up-dated with elements confidently recognized by the other one and vice-versa: the assumption is that classifiers involved in the co-training process have a conditionally independent view of the data. Co-train and self-train have a very strong role, in the current state of the art, on biometrics template up-dating process [14], moreover they have been applied recently even on the field of handwriting recognition [10, 11]. The main result observed for self-training on the task of handwritten word recognition [10] is that the challenge of successful self-training lies in finding the optimal tradeoff between data quality and data quantity for retraining. In particular, if the re-training is done with only those elements whose correctness can nearly be guaranteed, the retraining set does not change significantly and the classifier may remain nearly the same, or in other cases it could discard genuine samples whose distribution is far from the one already embedded in the knowledge base thus resulting in a performance degradation. Enlarging the retraining set, on the other hand, is only possible at the cost of increasing noise, i.e. adding mislabeled words to the training set. In this scenario, the co-train approach [11] appears to be much more interesting, in fact it does not suffer limitations of the self-update process, and performance improvement are more evident than those observed in the self-training case, even if the confidence threshold still plays a crucial role. Co-train can be easily extended from two to *n* classifiers but the basic observation is that, once more, even if an ensemble of classifiers is available, there is no analysis and use of their common behavior of classification given the input to be recognized and the specific combination schema.

From these observations, specific strategies are depicted in the next paragraph taking into account the task of supervised learning.

B. Learning Strategies

Let be:

- C_j, for j=1,2,…,M, the set of pattern classes,
- $P=\{x_k \mid k=1,2,...,K\}$, a set of pattern to be feed to the Multi Expert (ME) system. P is considered to be partitioned into S subsets $P_1, P_2, \ldots, P_s, \ldots, P_S$, being $P_s=\{x_k \in P \mid k \in [N_s\cdot(s-1)+1, N_s\cdot s]\}$ and $N_s=K/S$ (N_s integer), that are fed one after the other to the multi-expert system. In particular, P_1 is used for learning only, whereas $P_2, P_3, \ldots, P_s, \ldots, P_S$ are used both for classification and learning (when necessary);
- $y_s \in \Omega$, the label for the x_s pattern, $\Omega=\{C_1, C_2, ..., C_M\}$,
- A_i the *i-th* classifier for i=1,2,…,N,
- $F_i(k) = (F_{i,1}(k), F_{i,2}(k), \ldots, F_{i,r}(k), \ldots F_{i,R}(k))$ the numeral feature vector used by A_i for representing the pattern $x_k \in P$ (for the sake of simplicity it is here assumed that each classifier uses R numeral features)
- $KB_i(k)$, the knowledge base of A_i after the processing of P_k. In particular $KB_i(k)=\left(KB_i^1(k), KB_i^2(k), ..., KB_i^M(k)\right)$
- E the multi expert system which combines A_i hypothesis in order to obtain the final one.

Initially, first stage (*s=1*), the classifier A_i is trained using the patterns $x_k \in P^*_i = P_1$. Therefore, the knowledge base $KB_i(s)$ of A_i is initially defined as:

$$KB_i(s)=(KB^1_i(s), KB^2_i(s), \ldots, KB^j_i(s), \ldots, KB^M_i(s)) \quad (1a)$$

where, for j=1,2,…,M:

$$KB^j_i(s)=(F^j_{i,1}(s), F^j_{i,2}(s), \ldots, F^j_{i,r}(s), \ldots, F^j_{i,R}(s)) \quad (1b)$$

being $F^j_{i,r}(s)$ the set of the *r-th* feature of the *i-th* classifier for the patterns of the class C_j that belongs to P^*_i.

Successively, the subsets $P_2, P_3, \ldots, P_s, \ldots, P_{S-1}$ are provided one after the other to the multi-classifier system both for classification and for learning. P_S is just considered to be the testing set in order to avoid biased or too optimistic results. When considering new labeled data (samples of P_2,

P_3, …, P_s, …, P_{S-1}), two different strategies can be followed in order to select patterns from P_s to train A_i :

1. $\forall x_t \in P_s : update\_KB_i$,

   i.e. all the available new patterns belonging to P_s are used to update the knowledge base of each individual classifier

2. $\forall x_t \in P_s \ni' A_i(x_t) \neq y_t : update\_KB_i$,

   i.e. the individual classifier A_i is updated by considering only samples belonging to P_s which have been misclassified by A_i itself.

The second approach is derived from AdaBoost and bagging and, at the same time, it can be considered as a supervised version of self-training.
In order to inspect and take advantage of the common behavior of the ensemble of classifiers, the third strategy proposed in this work (and compared to the previous two) is the following:

3. $\forall x_t \in P_s \ni' \left(A_i(x_t) \neq y_t \wedge E(x_t) \neq y_t\right) : update\_KB_i$

   i.e. the individual classifier A_i is updated by considering all its misclassified samples if and only if these produce (or contribute to) a misclassification of the ME.

For the sake of simplicity, let us consider a ME adopting three base classifiers combined by means of Majority Vote. In the case depicted in figure 1(a) in which two classifiers correctly recognize the sample x_i, both the first and the second approach would update the knowledge base of A_i with x_i thus increasing the similarity index [16] of A_1, A_2 and A_3 with the only advantage of increasing performance of A_1 on the training set and on pattern similar to x_i. On the other hand, the ME system would exhibit the previous performance without any improvements. The third approach would not update the knowledgebase of A_1. In the case depicted in figure 1(b), the updating of the knowledge base of A_1 and/or A_3 would produce the improvements of the ME performance.

The third strategy takes into account performance of the individual classifier as well as performance at ME level. It is able to select not only samples to be used for the updating process, but also classifiers to which those samples must be feed. Of course, it is evident that, many new samples will not be feed to a specific classifier and, in general, we could expect to observe performance degradation if compared to the other strategies. This can happen depending by performances of classifiers as well as by the ratio new/old data.

However we have to consider and remark that we are dealing with already trained and working classifiers: initial performances are expected to be high (not weak). This leads to two considerations:

- given a specific classifier, the difference between the confidence value in the case of miss-classification and in the case of correct one could be imputed to the fact that the specific classifier (features, matching technique, etc.) is unable to represent it, and no improvements would be obtained by introducing the new sample in the knowledge base. This is particularly true under the assumption that strong (not weak) classifiers are used.
- if each classifier in the ensemble were able to recognize exactly the same set of patterns, it would be un-useful their combination. From this point of view we are interested in not increasing the similarity index (SI) among classifiers.

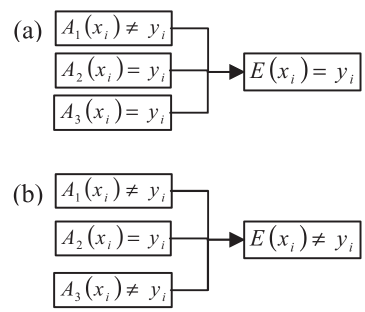

Figure 1. Examples of updating requests

III. EXPERIMENTAL SETUP

A. Data Set

A multi-expert system for handwritten digit recognition has been considered: the CEDAR database [9] $P=\{x_k \mid j=1,2,\ldots,20351\}$ (classes from "0" to "9") has been used.

The DB has been initially partitioned into 6 subsets:

- $P_1=\{x_1,x_2,x_3,\ldots,x_{12750}\}$,
- $P_2=\{x_{12751},\ldots,x_{14119}\}$,
- $P_3=\{x_{14120},\ldots,x_{15488}\}$,
- $P_4=\{x_{15489},\ldots,x_{16857}\}$,
- $P_5=\{x_{16858},\ldots,x_{18223}\}$,
- $P_6=\{x_{18224},\ldots,x_{20351}\}$.

In particular, $P_1 \cup P_2 \cup P_3 \cup P_4 \cup P_5$ represent the set usually adopted for training when considering the CEDAR DB [6]. P_6 is the testing dataset. Each digit is zoned into 16 uniform (regular) regions [5], successively, for each region, the following set of features have been considered [6]:

F_1: features set 1: hole, up cavity, down cavity, left cavity, right cavity, up end point, down end point, left end point, right end point, crossing points, up extrema points, down extrema points, left extrema points, right extrema points;

F_2: features set 2 (contour profiles): max/min peaks, max/min profiles, max/min width, max/min height;

F_3: feature set 3 (intersection with lines): 5 horizontal lines, 5 vertical lines, 5 slant -45° lines and 5 slant +45° lines.

B. Classifiers

Tests have been performed taking into account Support Vector Machines (SVMs) and a Naive Bayes classifier (NB).

SVM is a binary (two-class) classifier, multi-class recognition is here performed by combining multiple binary SVMs. The kernel function adopted is the rbf gamma, performance are influenced by the standard deviation value (σ) and by the tolerance of classification errors in learning. In this work $\sigma^2=0.3var$ (*var* is the variance of the training data set) [6].

NB classifier fits, in the training phase, a multivariate normal density to each class C_j by considering a diagonal covariance matrix. Given an input to be classified, the Maximum a Posteriori (MAP) decision rule is adopted to select the most probable hypothesis among the different classes.

C. Combination techniques

Many approaches have been considered so far for classifiers combination. These approaches differ in terms of type of output they combine, system topology and degree of a-priori knowledge they use [1, 2, 3]. The combination technique plays a crucial role in the selection of new patterns to be feed to the classifier in the proposed approach.

In this work the following decision combination strategies have been considered and compared: Majority Vote (MV), Weighted Majority Vote (WMV), Sum Rule (SR) and Product Rule (PR). MV just considers labels provided by the individual classifiers, it is generally adopted if no knowledge is available about performance of classifiers so that they are equal-considered. The second approach can be adopted by considering weights related to the performance of individual classifiers on a specific dataset. Given the case depicted in this work, it seems to be more realistic, in fact the behavior of classifiers can be evaluated, for instance, on the new available dataset. In particular, let ε_i be the error rate of the *i-th* classifier evaluated on the last available training set, the weight assigned to A_i is $w_i = \log\left(1/\beta_i \right)$, being $\beta_i = \frac{\varepsilon_i}{1-\varepsilon_i}$.

Sum Rule (SR) and Product Rule (PR) take into account the confidence of each individual classifier given the input pattern and the different classes [1]. Before the combination, confidence values provided by different classifiers were normalized by means of Z-score.

IV. RESULTS

Results are reported in terms of error rate percentage (ER). Values of the similarity index (SI) are reported in the last row of each table, moreover for the different learning strategies and for each classifier, the following ratios are evaluated and reported:

$$R1 = {N_S}/{N_F} \cdot 100, \qquad R2 = {N_S}/{N_I} \cdot 100,$$

being N_F the total number of available new samples, N_S the number of samples (selected among the previous one) used for learning and N_I the number of samples used for the initial training. *R1* represents the percentage of new patterns selected from those available while *R2* is a measure of their influence on the initial training set. The label "*X*-feed" refers to the use of the *X* modality for the feedback training process: "*All*" is the feedback of the entire set (first strategy in par. *II.B*), "*A*" is feedback at classifier level (second strategy in par. *II.B*). "*MV*", "*WMV*", "*SR*", "*PR*" are feedback at ME level adopting, respectively, the majority vote, the weighted majority vote, the sum rule and the product rule schema.

Table I reports results related to the use of SVM. The three set of features F_1, F_2 and F_3 (see par. *III.A*) lead, respectively, to SVM_1, SVM_2 and SVM_3. P_1 is used for training and P_6 for testing. $P_2 \cup P_3 \cup P_4 \cup P_5$ is used for feedback learning. In this case the total amount of new samples is the 42.86% of the number of samples of the initial training set (P_1). The first column (*No-feed*) reports results related to the use of P_1 for training and of P_6 for testing, without applying any feedback (*R1=R2=0%*), while the approach *All-feed* uses all samples belonging to the new set in order to update the knowledge base of each single classifier (*R1=100%, R2=42.86%*). Depending by the combination technique, a specific strategy can outperform the others. In two cases out four (MV-feed and SR-feed), the multi-expert strategy outperforms the use of the entire new dataset, while on three cases out 4 (MV-feed, WMV-feed and SR-feed) the multi-expert strategy outperforms the feedback at single expert level. In the case of Majority Vote and of Sum Rule, feedback at ME level definitively outperforms other two approaches. In these cases it is of interest the fact that a very restricted subset of samples is selected for re-training.

Table II reports results related to the use of NB classifier under the same conditions of the previous experiment. In this case performance improvements provided by feedback at single expert level are always better than those obtained by any ME technique. At the same time, it must be underlined that the worst re-training strategy is the one which considers the entire set of new available samples.

A typical implementation of co- and self-train takes into account multiple training iterations. In this work experiments have been performed considering up to 3 iterations.

TABLE I. SVM, FEEDBACK - $P_2 \cup P_3 \cup P_4 \cup P_5$

	No-feed ER	A- feed			MV-feed			WMV-feed			SR-feed			PR- feed			All-feed ER
		ER	R1	R2	ER	R1	R2	ER	R1	R2	ER	R1	R2	ER	R1	R2	
SVM_1	2.94	2.82	6.13	2.63	3.01	2.62	1.12	3.01	1.70	0.73	2.96	1.39	0.60	2.96	1.17	0.50	2.92
SVM_2	8.37	8.13	10.46	4.48	8.36	3.06	1.31	8.55	2.14	0.92	8.04	1.66	0.71	8.22	1.32	0.56	7.97
SVM_3	4.09	4.35	4.12	1.76	4.46	2.27	0.97	4.46	2.27	0.97	4.32	1.26	0.54	4.18	0.95	0.41	4.23
MV	2.54	2.49			**2.35**			X			X			X			2.58
WMV	**1.69**	1.83			X			1.79			X			X			1.74
SR	1.46	1.41			X			X			**1.36**			X			1.41
PR	1.22	**1.17**			X			X			X			1.22			**1.17**
SI	91.29	91.30			90.98			90.91			91.32			91.24			91.55

TABLE II. BF, FEEDBACK - $P_2 \cup P_3 \cup P_4 \cup P_5$

	No-feed ER	A- feed			MV-feed			WMV-feed			SR-feed			PR- feed			All-feed ER
		ER	R1	R2	ER	R1	R2	ER	R1	R2	ER	R1	R2	ER	R1	R2	
BF_1	6,81	5,97	11,54	4,95	6,44	5,63	2,41	6,67	3,84	1,65	6,72	3,99	1,71	6,63	3,38	1,45	6,53
BF_2	12,55	11,51	14,49	6,21	11,70	7,10	3,04	11,94	5,31	2,27	12,08	4,45	1,90	12,08	3,77	1,61	12,27
BF_3	10,62	9,59	10,92	4,68	10,24	5,74	2,46	10,24	5,74	2,46	10,29	3,71	1,59	10,34	2,87	1,23	11,04
MV	6,44	**5,64**			5,97			X			X			X			6.63
WMV	4,56	**4,14**			X			4,23			X			X			4.70
SR	3,67	**3,24**			X			X			3,52			X			3.81
PR	3,10	**2,82**			X			X			X			3,01			3.10
SI	84,05	85,34			84,70			84,52			84,37			84,32			84.35

TABLE III. BF, 3 FEEDBACK LEARNING ITERATIONS - $P_2 \cup P_3 \cup P_4 \cup P_5$

	No-feed ER	A- feed			MV-feed			WMV-feed			SR-feed			PR- feed			All-feed ER
		ER	R1	R2	ER	R1	R2	ER	R1	R2	ER	R1	R2	ER	R1	R2	
BF_1	6,81	5,59	30.19	12.93	6.16	15,06	6,45	6,48	10,41	4,46	6,44	10,50	4,50	6.58	8,96	3,84	6,77
BF_2	12,55	10.71	38.27	16.40	11,28	18,00	7,71	11,70	13,48	5,78	11.61	11,64	4,99	11.56	9,68	4,15	12,45
BF_3	10,62	9.12	29.36	12.58	9.68	15,00	6,43	9.63	15,08	6,46	10,20	9,46	4,05	10,39	7,46	3,20	11,61
MV	6,44	5.60			**5,39**			X			X			X			6.91
WMV	4,56	**4.09**			X			4,13			X			X			4.79
SR	3,67	**3.01**			X			X			3,34			X			3.99
PR	3,10	**2.63**			X			X			X			2.76			3.24
SI	84,05	86.47			85.26			84,96			84.56			84,32			85.37

TABLE IV. FEEDBACK - P_2, P_3, P_4, P_5

	No-feed ER	C-feed			MP- feed			WMV- Feed			SR- Feed			PR- Feed			All-feed ER
		ER	R1	R2	ER	R1	R2	ER	R1	R2	ER	R1	R2	ER	R1	R2	
SVM	2.07	2.10	1,60	0,17	2.06	1,18	0,13	2.07	1,33	0,14	2.08	1,18	0,13	2.08	1,15	0,12	1.91
NB	4.18	4.02	4,18	0,45	4.10	2,25	0,24	4.11	2,23	0,24	4.13	2,38	0,25	4.11	2,28	0,24	3.93
MP	3.48	3.44			3.51			X			X			X			**3.32**
WMV	2.54	*2.54*			X			*2.54*			X			X			**2.35**
SR	2.35	2.35			X			X			**2.30**			X			2.41
PR	2.21	2.22			X			X			X			2.21			**2.17**
SI	94.24	94.42			94.33			94.31			94.31			94.30			94.62

In the case of SVM classifiers, slight improvements have been observed in terms of ER while conferming the general trend between different feedback strategies already reported in table I. A much more interesting trend has been observed in the case of NB classifiers (table III) where 3 re-training iterations have been considered. The spread between performance obtained with a single expert strategy and a ME one is sensily lower than the case of a single iteration (table II), moreover MV-feed is able to outperform feedback at single expert level. It is also of interest to observe that, due to over-fitting, results obtained giving the entire set of new samples for feedback learning provides a decreasing of performance as the number of iterations increase.

Finally, table IV reports results related to the use of a unique feature set F= $F_1 \cup F_2 \cup F_3$, the two different classifiers and a reduced set of samples provided for feedback learning. In particular P_1 is used for the initial training and P_6 for testing. P_2, P_3, P_4, P_5 are independently used, one from the other, for feedback learning, performance were evaluated for each set and the average is finally reported. The first consideration is that the ER performed by SVM is so low that it appears un-useful combining it with BF (no complementary info is added). Of course this represents an extreme working point. Feedback at ME level is able to outperform All-feed in the only case of Sum Rule (SR) by using a reduced subset of the available new patterns. In all other cases, results provided by the ME feedback approach are equal to those obtained by feedback at single expert level.

A statistical significance of the $\alpha<0.03$ level was achieved for all tests performed and here reported.

V. Conclusions

This paper shows the possibility to improve the effectiveness of a multi-classifier system, when new labeled data are available, by a suitable use of the information extracted from the collective behavior of the classifiers. Experiments have been performed considering state of the art classifiers, features and combinations techniques. It has been showed that performance of feedback training strictly depend by the classifier structure, by the combination strategy of the ME which is responsible for sample selection, but also by the data distribution, and the similarity between samples in the feedback set and samples of the testing set. It has also showed that multiple training iterations on the same set of data are able to improve performance both in the case of feedback at single and multi expert level. Finally, also in cases of which the cardinality of the new selected training set is negligible if compared to that of the initial training set, the feedback strategy is able to produce improvements.

Future works will inspect deeply the possibility of iterative re-training given a set of new labeled samples as well as the possibility of evaluate the approaches on the task of semi-supervised learning.

References

[1] J. Kittler, M. Hatef, R.P.W. Duin, J. Matias, "On combining classifiers", IEEE Trans. on PAMI, Vol.20, no.3, pp.226-239, 1998.

[2] R. Plamondon, S.N. Srihari, “On-line and Off-line Handwriting Recognition: A comprehensive survey”, IEEE Trans. on PAMI, Vol. 22, n.1, pp. 63-84, 2000.

[3] C.Y. Suen, C. Nadal , R. Legault, T.A. Mai, L. Lam, “Computer Recognition of unconstrained handwritten numerals”, Proc. IEEE, Vol. 80, Issue 7, pp. 1162-1180, 1992.

[4] C.Y. Suen, J. Tan, “Analysis of errors of handwritten digits made by a multitude of classifiers”, Pattern Recognition Letters, Vol. 26, Issue 3, pp. 369-379, 2005.

[5] G. Pirlo, D. Impedovo “Fuzzy-Zoning-Based Classification for Handwritten Characters”, IEEE Trans. on Fuzzy Systems, Vol. 19 , Issue 4, pp. 780-785, 2011.

[6] C.L. Liu, K. Nakashima, H. Sako, H. Fujisawa, “Handwritten digit recognition: benchmarking of state-of-the-art techniques”, Pattern Recognition, Vol. 36, No. 10, pp. 2271-2285, 2003.

[7] R.E. Schapire, “The strength of weak learnability”, Machine Learning, vol. 5, Issue 2, pp. 197-227, 1990.

[8] R. Polikar, “Bootstrap-Inspired Techniques in Computational Intelligence”, IEEE Signal Processing Magazine, Vol. 24, No. 4, pp. 59-72, 2007.

[9] Y. Freud and R.E. Schapire, “Decision-theoretic generalization of on-line learning and an application to boosting”, J. of Computer and System Sciences, vol. 55, no. 1, pp. 119-139, 1997.

[10] V. Frinken, H. Bunke, “Evaluating Retraining Rules for Semi-Supervised Learning in Neural Network Based Cursive Word Recognition”, proc. of ICDAR, pp. 31-35, 2009.

[11] V. Frinken, A. Fischer, H. Bunke, A. Fornes, “Co-Training for Handwritten Word Recognition”, in proc. of ICDAR, pp. 314-318, 2011.

[12] A. Blum and T. Mitchell, “Combining Labeled and Unlabeled Data with Co-Training”. ACM Proc. of COLT, pp. 92–100, 1998.

[13] H. J. Scudder, “Probability of Error of Some Adaptive Pattern-Recognition Machines”. IEEE Trans. on Information Theory, Vol. 11, No. 3, pp. 363–371, 1965.

[14] U. Uludag, A. Ross, A. Jain, “Biometric template selection and update: a case study in fingerprints”, Pattern Recognition, Vol. 37, Issue 7, pp. 1533–1542, 2004.

[15] D. Impedovo, G. Pirlo, "Updating Knowledge in Feedback-based Multi-Classifier Systems", in IEEE proc. of International Conference on Document Analysis and Recognition (ICDAR2011), pp. 227-231, 2011.

[16] D. Impedovo, G. Pirlo, L. Sarcinella, E. Stasolla, “Artificial Classifier generation for Multi-Expert System Evaluation”, in IEEE Proc. Of International Conference on Frontiers in Handwriting Recognition (ICFHR 2010), pp. 421-426, 2010.

[17] D. Impedovo, G. Pirlo, M. Petrone, "A multi-resolution multi-classifier system for speaker verification", early view on Wiley Expert Systems, 2011.

[18] G. Pirlo, C.A. Trullo, D. Impedovo, “A Feedback-Based Multi-Classifier System”, IEEE proc. of International Conference on Document Analysis and Recognition (ICDAR 2009), pp. 713-717, 2009.