2011 International Conference on Document Analysis and Recognition

Updating Knowledge in Feedback-based Multi-Classifier Systems

D. Impedovo, G. Pirlo
Dipartimento di Informatica
Università degli Studi di Bari
Bari, Italy
pirlo@di.uniba.it

Abstract— **In pattern recognition tasks it is frequent that new (labeled) data became available as the specific application scenario evolves. When 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. More specifically a misclassified sample, for a particular classifier, is used to update that classifier only if that sample produces a misclassification by the ensemble of classifiers. This approach is compared 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. Successful results have been obtained by considering the CEDAR (handwritten digit) database, moreover it is also shown how they depend by the specific combination decision schema, as well as by data distribution.**

Keywords-Feedback learning, Multi Expert, Training Sample Selection

I. INTRODUCTION

Classifier combination is a widespread strategy to design high-performance classification systems. Many approaches have been proposed so far for classifier combination which can differ in terms of type of output they combine, system topology and degree of a-priori knowledge they use [1,2,3].

Of course, whatever classifier combination method is considered, the basic consideration is that the performance obtained by combining the outputs of several classifiers of a set can outperform or be more robust than each classifier of the set. The collective behavior of a set of classifiers can convey more information that those of each classifier of the set, and this information can be exploited for classification aims [4,5].

On the basis of this consideration this paper addresses the problem to verify the possibility, for each individual classifier, to learn from the collective behavior of the whole set when new data becomes available. In particular the problem of selecting specific samples in order to update the knowledge base of each single classifier is addressed and compared against standard approaches in which the entire set of new training samples is feed to each classifier, or situations in which each classifier is trained and boosted independently from the behavior of the others. For this purpose, a feed-back based parallel topology is considered and experimental tests are carried out in the field of classification of handwritten digits.

The experimental results demonstrate that, to some extent, it is possible to use the output of a multi-expert system to improve the performance of the individual classifiers or to improve their common behavior and, finally, to improve the performance of the whole system.

The paper is organized as follows: Section 2 presents feed-back based parallel topology; Section 3 shows the process of learning; the experimental results are presented in Section 4; Section 5 reports some discussions and the conclusion of the work.

II. A FEEDBACK BASED TOPOLOGY

In pattern recognition tasks, it is frequent that new (labeled) data became available as the specific application scenario evolves. Classifiers need to learn the new information without forgetting the knowledge previously acquired. Depending on the specific classifier, the knowledge base updating process can be a difficult task which poses many issues to be addressed. For instance classifiers such as support vector machines (SVMs) require, in their native form, that the new data is available with the old one in order to completely re-train the system. This, depending on the distribution of the old and of the new data, could causes the lost of some information. Other classifiers, such as nearest neighbor or Hidden/Gaussian Markov Model (HMM/GMM) could be more easily updated, since in the former case it is sufficient to introduce the new sample into the specific class set, while in the latter one, the final conditions generated by the old dataset could be considered as initial conditions for the new training session.

If the whole set of data (old+new) is available, many interesting algorithms can be adopted in order to train the system by selecting specific samples. Among the others, AdaBoost [10, 12] 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, not all the learning algorithms accept weights for the incoming samples. Another interesting approach is the bagging one: a number of weak classifiers trained on different subset (random instance) of the entire dataset are combined by means of the simple majority voting [11].

1520-5363/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDAR.2011.54

When the old data is no longer available, the concept of incremental learning must be taken into account: *"learning from new data without having access to old one, while retaining previously acquired knowledge"* [11]. Approaches based on AdaBoost, bagging or on some clustering technique cannot be adopted in their original formulation. Recently an algorithm called Learn++ [13] and some variations of it [14] have been introduced to face this problem. These approaches, inspired by AdaBoost, are able to generate an ensemble of classifiers for each data set that became available, and combine these ensembles to create a new ensemble of ensembles. Learn++ generates new classifiers for the new data and combine them with the previous ones by means of a weighted majority vote schema. Weights are based on classifier performances and updated during the training process. The approach is also able to learn new classes.

All the mentioned algorithms are generally adopted when considering a single classifier in a stand-alone modality. Its performance are boosted by considering its behavior on the specific dataset which becomes available.

In this work the problem of learning new instances, independently from the problem of the availability of the old data, is faced by considering the behavior of a multi expert system. The idea is to have an ensemble of classifiers which can differ one from the other on the feature type and/or on the matching technique. From this perspective, updating the knowledge base of a multi classifier system must take into accounts not only the performance of each base classifier, but also the common behavior of the ensemble given the unseen data.

In a traditional multi-classifier system using a parallel topology (Fig. 1), the input pattern x_t is fed to N individual classifiers in parallel. Each classifier A_i provides a response $A_i(x_t)$, on the basis of the information stored in its knowledge base KB_i. The responses obtained by all the classifiers are then combined to obtain the final results $E(x_t)$ according to a suitable combination strategy $E(A_1(x_t), A_2(x_t),..., A_N(x_t)) \rightarrow E(x_t)$ [1].

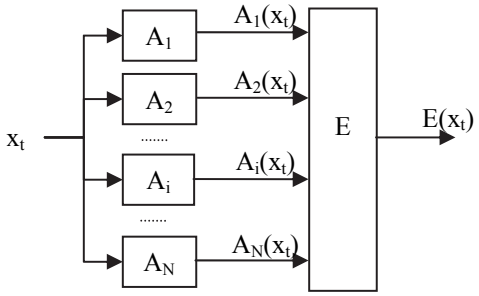

Figure 1. Multi-Classifier Parallel System

In the *feed-back* based *parallel* topology here investigated (Fig. 2) and compared against standard approaches, the final results $E(x_t)$ provided by the multi-expert system, according to $A_i(x_t)$ provided by the single classifier, determines whether or not the pattern is feed to the system for updating the knowledge base of individual classifiers [8].

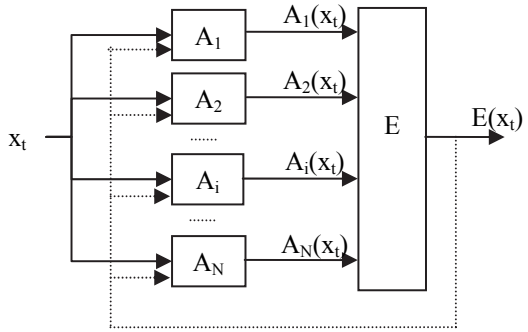

Figure 2. Feedback in the Multi-Classifier Parallel System

III. LEARNING FROM COLLECTIVE BEHAVIOUR

The learning process and the classification process are here considered as a unique, dynamical process, as it happens for human beings. In this process, after an initial training stage, the multi-expert system performs at the same time classification and learning, when necessary. More precisely, 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, …, P_s, …, 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,…,P_s,…,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 $KB_i(k) = (KB_i^1(k), KB_i^2(k),..., KB_i^M(k))$
- 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.

Three 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 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. A_i is updated by considering all misclassified samples independently from the final hypothesis provided by the ME.
3. $\forall x_t \in P_s \ni' (A_i(x_t) \neq y_t \wedge E(x_t) \neq y_t) : update\_KB_i$, i.e. 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 a simple Majority Vote approach. In the case depicted in figure 3(a), both the first and the second approach would update the knowledge base of A_1 with x_i while the third one would not update the knowledgebase of A_1. In this case performance of A_1 would be increased on the training set and on pattern similar to x_i, while the ME system would exhibit the previous performance without any improvements. In the case depicted in figure 3(b), the updating of the knowledge base of A_1 and/or A_3 would produce the improvements of the ME performance.

The first approach is the standard one: all new data are used to update the knowledge base. The second approach is based on the concept that each single classifier is able to select the most profitable samples in order to increase its own performance. Both approaches do not get benefits from the ME behavior. The third approach select patterns able to increase performances of the Multi Expert (ME) System taking into account performance of other classifiers in the pool. The first two approaches are expected to increase the similarity index [8] much than the third one. Of course from a performance point of view, the advantages offered by each approach strictly depends upon old-new-test data distribution. Moreover, it has to be underlined that, the ME strategy adopted strongly influence performance of the three different approaches.

In this work two decision combination strategies *E(·)* have been considered:

- Majority Vote (MV),
- Weighted Majority Vote (WMV).

The first one 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. In 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 defined:

$$w_i = \log\left(1/\beta_i\right),$$

being $\beta_i = \dfrac{\varepsilon_i}{1-\varepsilon_i}$.

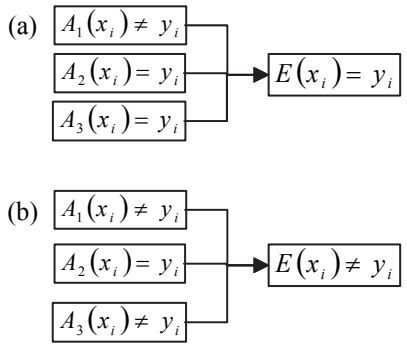

Figure 3. Examples of updating requests

IV. Experimental Results

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. P_1 contains the 70% of training samples.

In the first experiment P_1 is used for training while P_2, P_3, P_4 and P_5 are considered as new data and used both for feedback training and testing: performance of the system on P_2, P_3, P_4 and P_5 are evaluated before and after the feedback process. P_6 is always used for test.

Each digit is zoned into 16 uniform (regular) regions [5], successively for each region the following set of features have been considered [6]:

1. features 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;
2. contour profiles: max/min peaks, max/min profiles, max/min width, max/min height;
3. intersection with lines: 5 horizontal lines, 5 vertical lines, 5 slant -45° lines and 5 slant +45° lines.

The three different features types leads to three different classifiers, respectively named A_1 , A_2 and A_3 . Each classifier adopts a simple nearest neighbor matching algorithm. The classification accuracy is influenced by the number of nearest neighbors k. Results reported in this work have been obtained by k=3 and the tie of class scores was solved by the 1-NN rule [6].

Results are reported, in the following tables, in terms of error rate percentage. In particular, 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, "A" is feedback at classifier level, "MV" and "WMV" are feedback at ME level adopting, respectively, the majority vote and the weighted majority vote approaches.

Table 1 shows results related to the use of P_1 for training and P_6 for testing. P_2 , P_3, P_4, P_5 were independently used, one from the other, for feedback learning, performance were evaluated for each set and the average is finally reported. Values of the similarity index (SI) are reported in the last row.

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

	No-feed	**A-feed**	**MV-feed**	**WMV-Feed**	**All-feed**
A_1	16.49	16.42	16,57	16.56	16.20
A_2	13.91	13.96	13.96	13.96	13.85
A_3	6.25	6.15	6.16	6.16	6.51
MV	7.75	7.72	7.72	X	7.88
WMV	5.26	5.24	X	5.19	5.33
SI	80.67	80.75	80.67	80.67	80.70

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. If patterns to be feed are selected by the ME adopting the simple Majority Vote schema, performance of the ME are exactly the same would be obtained if each single classifier would have selected them. A different trend is observed if the WMV combination technique is adopted: performance of the ME are considerably improved with respect to all other cases.

Of course the advantages of the selection of samples by the ME system are more evident if the same dataset is used for feedback learning and test. Table 2 reports results related to the use of P_1 for training, P_2, P_3, P_4, P_5 were independently used, one from the other, for feedback learning and testing. Performance were evaluated for each set and the average is finally reported. Of course it is obvious that the use of all samples of the new dataset offers the best performance, but it also results in a very high Similarity Index. For higher values of the SI, the combination of experts could be un-useful.

TABLE II. FEEDBACK AND TESTING - P_2 , P_3, P_4, P_5

	No-feed	**A-feed**	**MV-feed**	**WMV-feed**	**All-feed**
A_1	21.09	10.10	17.13	18.84	9.99
A_2	15.15	6.17	11.07	12.99	6.09
A_3	7.12	2.96	4.45	4.45	2.85
MV	8.73	2.24	2.16	X	2.07
WMV	5.50	1.39	X	1.39	1.32
SI	76,25	88.33	79.37	77.81	88.45

Since results are influenced by the distribution of samples of the feedback set and of the testing set, tests have been also performed by considering different, random partitioning for P_2 , P_3, P_4, P_5. Results obtained are perfectly similar to those here reported.

Finally $P_1 \cup P_2 \cup P_3 \cup P_4 \cup P_5$ were used for training. P_6 was randomly partitioned into two subsets: the first was used for feedback training and the second for testing. Results are in table 3.

TABLE III. FEEDBACK - P_6

	No-feed	**A-feed**	**MV-feed**	**WMV-feed**	**All-feed**
A_1	15.75	14.62	15.00	15.65	13.03
A_2	14.15	13.68	13.96	14.34	13.59
A_3	6.37	6.28	6.28	6.28	6.00
MV	7.69	7.40	7.50	X	6.94
WMV	5.15	5.06	X	4.75	6.03
SI	80.66	81.66	81.26	80.72	82.72

In this case the two subsets (feedback-learning) exhibit a more uniform distribution of patterns than the cases in table 1. In fact results obtained with the WMV feedback schema sensibly outperform all other approaches.

V. CONCLUSIONS AND DISCUSSIONS

This paper shows the possibility to improve the effectiveness of a multi-classifier system by a suitable use of the information extracted from the collective behavior of the classifiers. More precisely, when a new dataset becomes available, the final decision obtained by combining the individual decisions provided by each classifier, has been used to upgrade the knowledge base of the individual

classifiers, when necessary, according to a feed-back based topology. The experimental results reported in this paper demonstrate that the collective behavior of a set of classifiers *can* provide useful information to improve system performance to a certain extent. Performance of the approach have shown to depend 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. Probably, performances depends also by the feature type and by the matching strategy adopted by individual classifiers. Future work will inspect this issue.

Finally it should be considered that, although the upgrading of the knowledge base of the individual classifiers can lead to an improvement of the performance of each individual classifiers, it can also lead to a reduced complementary in the individual behaviors, reducing the overall performance gain of the multi-classifier system. From this point of view, the use of the Learn++ algorithm should be considered in order to allow incremental learning. In particular the new classifier generated could be included in the set of already available classifiers, so that each classifier would be designed for a specific data cluster avoiding performance degradation and keeping low the similarity index.

In conclusion, this paper shows the performance of a multi-classifier system can be improved by exploiting the collective behavior of the set of classifiers. Of course, in order to make this strategy feasible, additional research is necessary.

REFERENCES

[1] J. Kittler, M. Hatef, R.P.W. Duin, J. Matias, "On combining classifiers", *IEEE Trans. on Pattern Analysis Machine Intelligence*, 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. Pattern Analysis and Machine Intelligence*, 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, 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*, No. 3, Feb. 2005, pp. 369-379.

[5] Lucchese M.G., Impedovo, S. Pirlo, G., "Optimal Zoning Design by Genetic Algorithms", *IEEE Trans. Sys. Men and Cybern.* - Part A, Vol. 36 , Issue: 5, Sept. 2006, pp. 833-846.

[6] C.L. Liu, K. Nakashima, H. Sako, H. Fujisawa, "Handwritten digit recognition: benchmarking of state-of-the-art techniques", *Pattern Recognition* 36 (2003) 2271 – 2285.

[7] D. Impedovo, G. Pirlo, L. Sarcinella, E. Stasolla, "Artificial Classifier generation for Multi-Expert System Evaluation, in IEEE Proc. Of ICFHR 2010, pp. 421-426.

[8] G. Pirlo, C.A. Trullo, D. Impedovo, "A Feedback-Based Multi-Classifier System", IEEE proc. of ICDAR 2009, pp. 713-717.

[9] J. Hull, "A database for handwritten text recognition research", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 16, n. 5, pp. 550–554, 1994.

[10] R.E. Schapire, "The strength of weak learnability", Mach. Learn., vol. 5, no.2, pp. 197-227, 1990.

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

[12] Y. Freud and R.E. Schapire, "Decision-theoretic generalization of on-line learning and an application to boosting", J. Comput. Syst. Sci., vol. 55, no. 1, pp. 119-139, 1997.

[13] R. Polikar, "Learn++: An Incremental Learning Algorithm Basedon Psyco-Physiological Models of Learning", in IEEE proc. Of the 23rd Annual conference IEEE/EMBS, 2001.

[14] M. Muhlbaier, A. Topalis, R. Polikar, "Learn++.MT, A New Approach to Incremental Learning", in proceedings of Multiple Classifier Systems, LNCS 3077, pp. 52-61, 2004.