Dynamic Selection of Ensembles of Classifiers Using Contextual Information

Paulo R. Cavalin[1], Robert Sabourin[1], and Ching Y. Suen[2]

[1] École de Technologie Supérieure, 1100 Notre-dame ouest, Montreal(QC), Canada, H3C-1K3
[2] CENPARMI, Concordia University, 1455 de Maisonneuve Blvd West, Montreal(QC), Canada, H3G-1M8

Abstract. In a multiple classifier system, dynamic selection (DS) has been used successfully to choose only the best subset of classifiers to recognize the test samples. Dos Santos et al's approach (DSA) looks very promising in performing DS, since it presents a general solution for a wide range of classifiers. Aiming to improve the performance of DSA, we propose a context-based framework that exploits the internal sources of knowledge embedded in this method. Named DSA^c, the proposed approach takes advantage of the evidences provided by the base classifiers to define the best set of ensembles of classifiers to recognize each test samples, by means of contextual information provided by the validation set. In addition, we propose a switch mechanism to deal with tie-breaking and low-margin decisions. Experiments on two handwriting recognition problems have demonstrated that the proposed approach generally presents better results than DSA, showing the effectiveness of the proposed enhancements. In addition, we demonstrate that the proposed method can be used, without changing the parameters of the base classifiers, in an incremental learning (IL) scenario, suggesting that it is also a promising general IL approach. And the use of a filtering method shows that we can significantly reduce the complexity of DSA^c in the same IL scenario and even resulting in an increase in the final performance.

1 Introduction

Dynamic selection (DS) of classifiers is a very interesting domain for multiple classifier systems (MCS). DS consists of selecting only the best members from the pool of classifiers, denoted as $C = \{c_1, c_2, \ldots, c_N\}$, to recognize the test sample $x_{i,test}$. As a result, the best classification scheme is defined for each sample, so that lower error rates are expected. Note that when more than one classifier is selected, we can refer to this method as dynamic selection of ensembles of classifiers (DSEoC).

A promising approach for DSEoC is Dos Santos et al's approach (DSA) [1], which is able to dynamically select ensembles of classifiers (EoCs) by using only crisp label outputs, i.e. class votes, provided by the base classifiers. DSA is a general DSEoC approach, since any type of classifier that outputs votes can be used as a base classifier. A general approach is very desirable since it can be easily adapted to different base classifiers, and it can be used with combinations of different types

of classifiers. This allows us, for example, to combine decisions of neural networks and hidden Markov models, or a classifier with the decision of a human expert.

Nonetheless, the structure of DSA is very rich, and many internal sources of knowledge have not been fully exploited yet. For example, we could improve the way we take advantage of the diversity presented by the members of the pool of EoCs. Also, we could improve the way these EoCs are selected. For example, we could select more than a single EoC to recognize $x_{i,test}$. By improving the way DSA uses these sources of knowledge, we believe that we can reach higher recognition rates, resulting in an approach that is both general and robust. Consequently, we propose a method that tries to use this knowledge in a better way.

The proposed method, to which we refer as DSA^c, includes the evidences produced by the base classifiers to take advantage of a labeled dataset (e.g. a source of contextual information) to indicate which is the best set of EoCs for each test sample. In this case, it is not a single EoC that is selected to recognize the test sample, but the best set that can comprise one or more EoCs. Such a selection uses output profiles, which are represented by the outputs of the base classifiers, to find the samples in the validation set that are those most similar to the test sample. Afterwards, we compute the best set of EoCs to recognize $x_{i,test}$ based on the evidences provided by those most similar validation samples. Furthermore, we also add a switch mechanism to DSA^c, aiming at choosing the best source of knowledge to compute the final decision whenever the answer provided by the dynamically selected EoC is not considered convincing enough. Consequently, the switch works as a tie-breaking approach, which reduces the chances of random decisions by using existing knowledge.

The remainder of this paper is organized as follows. In Section 2, we provide a brief description of DSA. In Section 3, we describe the proposed approach named DSA^c. Next, in Section 4, we report and discuss the results of an experimental evaluation performed on two handwriting recognition problems. Finally, in Section 5, we present conclusions and point out future work.

2 Dos Santos et al's approach (DSA)

The overall architecture of DSA is depicted in Figure 1. The main objective of this method is to dynamically find the best EoC, whose members are a subset of $C = \{c_1, c_2, \ldots, c_N\}$, to recognize the test sample $x_{i,test}$. This task is performed by considering only the recognition outputs $O_i = \{o_{i,1}, \ldots, o_{i,N}\}$ computed from C. Each output corresponds to a class from the set $\Omega = \{\omega_1, \ldots, \omega_M\}$.

DSA is divided into two phases: the design phase and the operational phase. During the design phase, the architecture that supports the dynamic selection of EoCs is created. In other words, the pool of EoCs $C^{*\prime} = \{C'_1, \ldots, C'_W\}$, where $C'_j \subset C, 1 \leq j \leq W$, is created during this phase and is a subset of all possible EoCs C^*. The pool $C^{*\prime}$ is generated by a search algorithm, which is a genetic algorithm (GA) in this work. Each individual is represented by a binary vector of N positions, where each bit represents whether or not a classifier is selected to be a member of an EoC. The fitness function, which has to be minimized, uses the error rate on the optimization set *Opt*, by applying the majority voting method on the EoCs assigned by each individual. In order to avoid overfitting, each individual

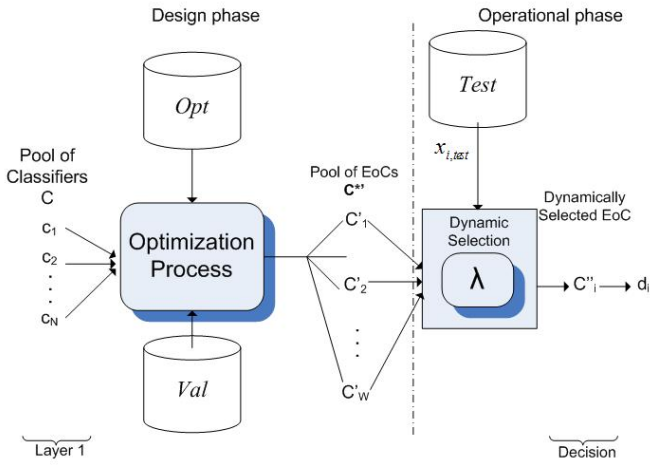


Fig. 1. Dos Santos et al's approach (DSA). The pool of classifiers is organized into a pool of EoCs during the design phase. During the operational phase, the EoC, which is dynamically selected by λ, produces the final decision.

is also evaluated on the validation set Val, and the best solutions are saved into an archive whose size is W. The archive is then used as $C^{*\prime}$.

The operational phase is composed of the modules that conduct the dynamic selection of the best EoC C''_i, which includes one of the EoCs in $C^{*\prime}$, to recognize $x_{i,test}$. This task is undertaken by using a dynamic selection function (DSF), to which we refer as λ. Afterwards, we compute the votes provided by all members in C''_i, and the class with the highest number of votes represents the final decision d_i.

The DSF λ is related to one DSF, as described in [1], such as Ambiguity-guided dynamic selection (ADS), Margin-based dynamic selection (MDS), and Class-strength dynamic selection (CSDS). In this work, we simply assign DSA to ADS for the sake of simplicity.

3 DSA^c: enhancing dynamic selection by using contextual information and a switch mechanism

In DSA, EoCs are dynamically selected by considering DSFs based on the extent of consensus. Despite that the extent of consensus is a well studied concept in literature [2], only the outputs of the most voted and the second most voted classes are used to select the ensemble. However, the information related to the other classes is wasted, even though such information could help this task. In addition, only one EoC is dynamically selected at a time. Finally, in many cases, the final recognition might not be confident enough, and random guesses are associated to the final result. In order to overcome these drawbacks, we propose DSA^c, depicted in Figure 2.

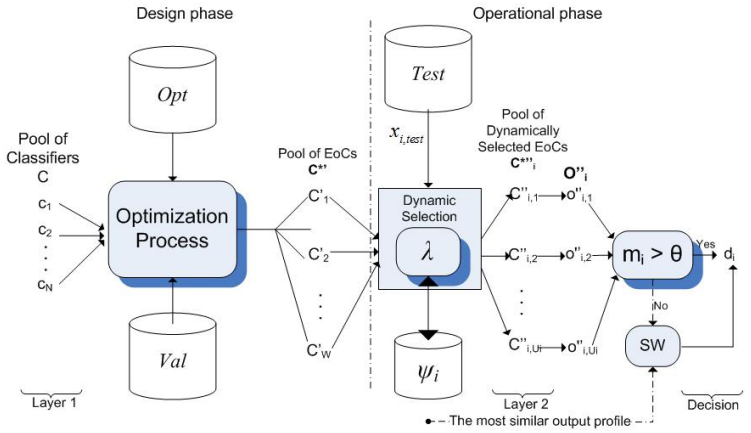


Fig. 2. An overview of the DSA^c approach. This method uses the knowledge provided by Val (converted into the set of output profiles Val').

The main objective of DSA^c is to use the validation database, transformed into output profiles, to point out which EoCs are the most competent to recognize the test sample $x_{i,test}$. An output profile is computed by the transformation T in Equation 1, where $x_i \in \Re^D$, $\tilde{x}_i \in Z^{N+}$, and N is the size of the pool of base classifiers C. Given that we know which EoC correctly recognizes each validation sample, the dynamically selected set of EoCs, denoted by $C^{*\prime\prime}{}_i = \{C''_{i,1}, \ldots, C''_{i_U}\}$, is composed by the EoCs that correctly classify the validation samples that are the ones most similar to the test samples in considering the output profiles.

$$T : x_i \Rightarrow \tilde{x}_i, \tag{1}$$

In greater detail, this approach works as follows. Consider the pool of EoCs $C^{*\prime}$, generated during the design phase. For each test sample $x_{i,test}$, we compute the best set of EoCs $C^{*\prime\prime}{}_i$, composed of members from $C^{*\prime}$. Each EoCs from $C^{*\prime}$ may appear several times in $C^{*\prime\prime}{}_i$, resulting in an automatic weighting approach. This task is achieved by considering the DSF λ, which is depicted in Figure 3.

The DSF λ works as follows. First, we apply T on $x_{i,test}$, resulting in $\tilde{x}_{i,test}$. Next, we compare $\tilde{x}_{i,test}$ to each output profile in Val', which is a database containing the output profiles of all validation samples in Val, e.g. $\tilde{x}_{j,val}$ $\forall x_{j,val} \in Val$, computed during the design phase. We compare these samples in terms of similarity, and store the degree of similarity between $\tilde{x}_{i,test}$ and $\tilde{x}_{j,val}$ in $\delta_{i,j}$. Note that we use one of the similarity measures described in Section 3.1 to compute $\delta_{i,j}$. The K most similar output profiles $\tilde{x}_{j,val}$, e.g. the validation samples related to the highest values of $\delta_{i,j}$, are stored in Ψ_i. Next, the EoCs from $C^{*\prime}$ which correctly recognize each sample in Ψ_i are computed. These EoCs are then included in $C^{*\prime\prime}{}_i$. As mentioned, an EoC appears in $C^{*\prime\prime}{}_i$ as many times as the number of samples that it correctly recognizes. Finally, $C^{*\prime\prime}{}_i$ is submitted to the remaining modules of DSA^c.

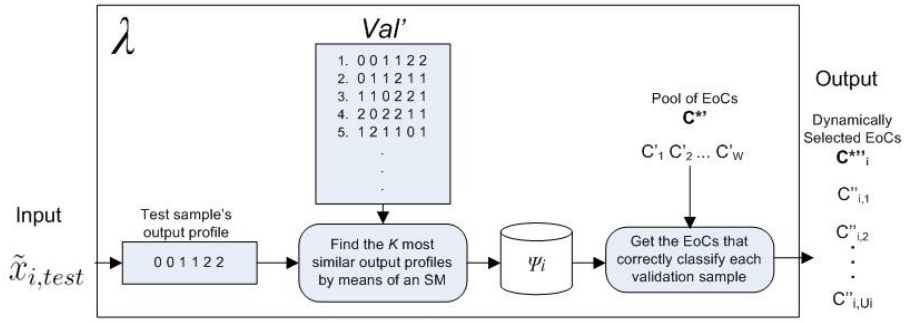


Fig. 3. The DSF λ. For each test sample, we find K validation samples with the most similar output profiles, to form the set ψ_i. The EoCs that correctly classify the validation samples in Ψ_i are used to compose the set $C^{*\prime\prime}$, which is then used to compute the final decision of DSA^c.

The last step consists of submitting the outputs of $C^{*\prime\prime}{}_i$ to the switch mechanism, represented by the SW module in Figure 2. Here we employ the concept of margin [2] to identify whether or not the answers provided by $C^{*\prime\prime}_i$ are confident enough, using a threshold θ. In considering the margin m_i, for the test sample $x_{i,test}$, if $m_i > \theta$, then we use the most voted class indicated by $C^{*\prime\prime}{}_i$. Otherwise, we use the label of the most similar validation sample from Ψ_i. The main goal of this scheme is to use contextual information also in the switch mechanism. Note that, hereafter, the margin is represented by the difference between the number of votes of the most voted class and the second most voted one.

In the next section we present the similarity measures used to compute $\delta_{i,j}$.

3.1 Similarity measures (SMs)

Hereafter, we use the following additional notations: $\tilde{x}_{i,test,k}$ represents the output of classifier k for $x_{i,test}$, and $\tilde{x}_{j,val,k}$ represents the same for $x_{j,val}$. In addition, for each $x_{j,val}$, the set of flags $CC_j = \{cc_{j,1}, cc_{j,2}, \ldots, cc_{j,W}\}$, where each $cc_{j,k}$ is a binary value, represents whether C'_k has correctly classified $x_{j,val}$ or not. In other words, $cc_{j,k} = 1$ if C'_k correctly classifies $x_{j,val}$, otherwise, $cc_{j,k} = 0$.

In this work we consider three different SMs. Note that they are individually used by λ. The three SMs are described below.

Euclidean distance (ED) The Euclidean distance between the output profile of $\tilde{x}_{i,test}$ and each $\tilde{x}_{j,val}$ $\forall j$, represented by the following equation:

$$ED_{i,j} = \sum_{k=1}^{N} |\tilde{x}_{i,test,k} - \tilde{x}_{j,val,k}| \tag{2}$$

Template matching (TM) The computation of the number of classifiers that provide the same output. This SM is computed by maximizing Equation 3.

$$TM_{i,j} = \frac{\sum_{k=1}^{N} \alpha_{i,j,k}}{N} \tag{3}$$

$$\alpha_{i,j,k} = \begin{cases} 1, & \text{if } \tilde{x}_{i,test,k} = \tilde{x}_{j,val,k} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Oracle-based template matching (OTM) In considering that each $\tilde{x}_{j,val}$ is related to the correct class label $correct_{j,val}$, we compute the number of classifiers that produce the correct class label for $\tilde{x}_{j,val}$ and provide the same output as $\tilde{x}_{i,test}$. Equation 5, which has to be maximized, computes this SM mathematically.

$$OTM_{i,j} = \frac{\sum_{k=1}^{N} \beta_{j,i,k}}{\sum_{k=1}^{N} \gamma_{j,k}} \tag{5}$$

$$\beta_{i,j,k} = \begin{cases} 1, & \text{if } \tilde{x}_{i,test,k} = \tilde{x}_{j,val,k} \text{ and } \tilde{x}_{j,val,k} = correct_{j,val} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$\gamma_{j,k} = \begin{cases} 1, & \text{if } \tilde{x}_{j,val,k} = correct_{j,val} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

These SMs result in three different versions of DSA^c:

1. DSA$^c_{ED}$, where $\delta_{i,j} = 1 - \text{ED}_{i,j}$;
2. DSA$^c_{TM}$, where $\delta_{i,j} = \text{TM}_{i,j}$;
3. DSA$^c_{OTM}$, where $\delta_{i,j} = \text{OTM}_{i,j}$.

In the next section, we present an experimental evaluation of these methods.

4 Experiments

In this section we present a series of experiments whose main goals are: 1) to evaluate whether dynamic methods are better than static ones; 2) to evaluate if DSA^c results in lower error rates than DSA;

The aforementioned evaluation is supported by considering these methods: the original classifier with full representation space (all original features); the best base classifier from C; the fusion of all base classifiers in C by using MV; fusion of all base classifiers in C by using decision templates (DT) [3], by considering the three proposed SMs; the best EoC from $C^{*\prime}$.

All methods are evaluated on two handwriting recognition problems: digits and uppercase letters, extracted from the NIST-SD19 database. For both problems, the original feature set is composed of 132 features, extracted from concavities and contours [4]. In addition, two different test sets for digits are used for evaluating digit recognition: *NIST-digits-test1* and *NIST-digits-test2*. Table 1 presents a detailed description of each database.

Table 1. Experimental setup. (NC: number of classes; NF: number of features; NFE: number of features in the subspace, after applying the RSS method; VM: validation method;).

Problem	NC	Set $Train$	Set Opt	Set Val	Set $Test$	NF	NFE
Digits	10	5,000	10,000	10,000	*t1* 60,089 *t2* 58,646	132	32
Letters	26	43,160	3,980	7,960	12,092	132	32

For each dataset, 100 base classifiers, with 32 features, are generated from the original 132 features based on the random subspaces (RSS) ensemble generation method [5] for the number of features used for each problem). Two different base classifiers are considered: k-nearest neighbors classifiers with $k = 1$ (1NN), and C4.5 decision tree (DTree) classifiers.

To generate the pool of EoCs, GA is used to find an archive with the 25 best solutions on Val, representing $C^{*\prime}$, guided by the optimization set Opt. The following parameters were used in this work: population size: 128; maximum number of generations: 1,000; probability of crossover: 0.8; probability of mutation: 0.01; one-point crossover and bit-flip mutation [1]. The experiments are replicated 30 times, where in each replication the archive provided by GA is generally different. The results represent the mean error rates over the 30 replications.

To validate the results statistically, we use the Kruskal-Wallis nonparametric statistical test. We test the equality among the mean values, using a confidence level of 95%. Dunn-Sidak correction is applied to critical values.

4.1 Results and discussion

The results in error rates are presented in Table 2. For both digits and letters, DSA^c with $K = 30$. Also, the parameter θ was set to zero after preliminary evaluations.

These experiments show that the proposed approach DSA^c has successfully presented lower error rates than DSA in all problems. This proves that both the use of contextual information to select multiple EoCs, as well as the switch mechanism, were able to better use the sources of knowledge embedded in DSA. Consequently, these results demonstrate that dynamic selection outperforms static selection.

One interesting result from letters with DTree, shows the effectiveness of the switch method to decrease the dependency on the pool of EoCs. In that problem, the final error rates are lower than the oracle of the pool of EoCs. It would be impossible to reach this result without this mechanism.

Evaluation of DSA^c in an incremental learning scenario One by-product of the proposed DSA^c approach, is the ability to adapt the system to knowledge acquired incrementally by simply adding more data to Val. As a result, we can incrementally learn new data with no need to update the parameters of the base classifiers. For this reason, we evaluate the impact of an increase in the size of Val for DTree classifiers on *NIST-digits-test1*.

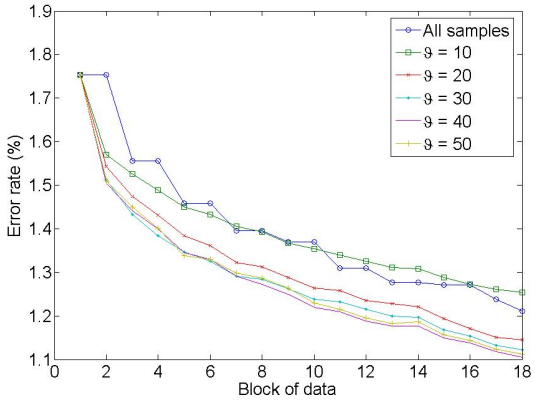
Table 2. Error rates using 1NN and DTree classifiers, at zero-level rejection. Results in bold present the best approach among static MO, DSA, DT, and the proposed DSA^c with K set to 30. Underlined results represent the statistically-significant best method. Highlighted by * are the proposed approaches. Between parentheses is the variance of each approach ($\times 10^{-2}$).

Classifier	**1NN**			**DTree**		
Method	**Digits**		**Letters**	**Digits**		**Letters**
	test1	*test2*		*test1*	*test2*	
Static selection						
Oracle C	0.05	0.17	0.18	0.01	0.04	0.04
All features	6.66	9.76	7.82	11.07	18.20	13.50
Best from C	7.52	13.99	14.47	10.30	19.18	17.13
MV all C	3.72	8.10	6.60	2.92	6.67	6.06
Best from $C^{*\prime}$	3.60 (3.83)	7.77 (7.74)	6.56 (6.73)	2.98 (4.98)	6.77 (1.12)	6.21 (7.82)
DT$_{TM}$ C	2.55	5.74	4.95	2.00	5.00	4.64
DT$_{OTM}$ C	4.74	9.74	7.56	2.70	6.03	7.15
DT$_{ED}$ C	2.97	6.57	6.55	2.56	6.26	7.44
Dynamic selection						
Oracle $C^{*\prime}$	1.97 (0.02)	4.59 (1.14)	3.87 (4.42)	1.87 (1.03)	4.39 (4.36)	4.53 (2.49)
DSA	3.61 (0.08)	7.87 (0.17)	6.43 (0.48)	2.87 (0.06)	6.61 (0.29)	6.06 (0.41)
$*$DSA$^c_{TM}$	**2.37** (0.02)	**5.34** (0.04)	4.62 (0.16)	**1.76** (0.02)	**4.36** (0.04)	4.20 (0.05)
$*$DSA$^c_{OTM}$	2.63 (0.03)	5.88 (0.17)	**4.10** (0.25)	2.16 (0.03)	4.96 (0.08)	**3.89** (0.06)
$*$DSA$^c_{ED}$	2.43 (0.03)	5.43 (0.16)	4.39 (0.28)	1.83 (0.05)	4.64 (0.10)	4.32 (0.09)

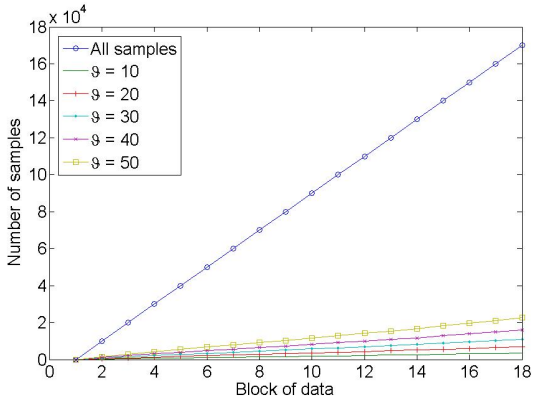
We simulated an incremental scenario by incrementally increasing the number of samples in Val. We take advantage of the large set of digits available in the NIST SD19 database, by increasing the size of Val from 10,000 to 180,000 samples. Those are the remaining samples in the hsf_{1-3} series of the database. In addition, we also evaluate a control mechanism to select only samples that present the margin below a threshold ϑ, in considering the margin of the base classifiers, e.g. $m_i < \vartheta$, where m_i represents the difference of votes between the two most voted classes. The main idea is to hold only samples that present uncommon output profiles to reduce the size of Val. As a consequence, the final complexity of DSA^c is also reduced.

Figure 4(a) presents the results of these experiments. As shown, the incremental increase of Val is effective in reducing the error rates. By using all samples, the final error rates are reduced from 1.75% to about 1.2%. Furthermore, the proposed control mechanism is also effective in reducing the error rates. With $\vartheta = 40$, the final error rates are reduced to about 1.1%. In addition, we demonstrate in Figure 4(b) the effects of using the margin-based control mechanism. The best approach, represented by $\vartheta = 40$, used only 25,948 samples for training. Comparing with the use of all 180,000 training samples, we can reach better results by using only around 15% of this set and drastically reduce the complexity of DSA^c.

In addition, we also compare the performances of DSA and DSA^c by considering a rejection mechanism on *NIST-digits-test1* with DTree classifiers. The reject uses the margin of the dynamically selected EoC for DSA, and the margin of the dynamically selected set of EoCs for DSA^c. As depicted in Figure 5, DSA^c rejected less samples than DSA to reach the same error rates (for example, with 2.0 % of error, DSA rejects about 2% of samples, and DSA^c rejects only 1.5%.). When more samples are used for training, the DSA^c rejects even less samples. And, the performance of the approach using $\vartheta = 40$ was better than the one that uses all samples, even though the former was trained with much fewer samples.



(a) Performance



(b) Complexity

Fig. 4. Incremental evaluation of DSA^c ($K = 30$) with DTree classifiers on *NIST-digits-test1*.

5 Conclusion and future work

In this paper we proposed a novel methodology to improve a state-of-the-art approach for DSEoC, e.g. Dos Santos et al's approach (DSA). The proposed approach, referred to as DSA^c, uses the knowledge provided by output profiles to dynamically select EoCs. Furthermore, a switch mechanism was included to reduce the dependency on the pool of EoCs.

Experiments conducted on two handwriting recognition problems have confirmed that dynamic selection is really promising in improving the use of multiple classifiers. In addition, DSA^c has been effective to improve DSA. Also, the simulation of an incremental learning scenario showed us that we can improve the performance of DSA^c by increasing the size of the validation set only, without changing the parameters of the base classifiers.

As future work, many directions to improve DSA^c can be followed. One of these directions consists of reducing the overall complexity. We can, for example,

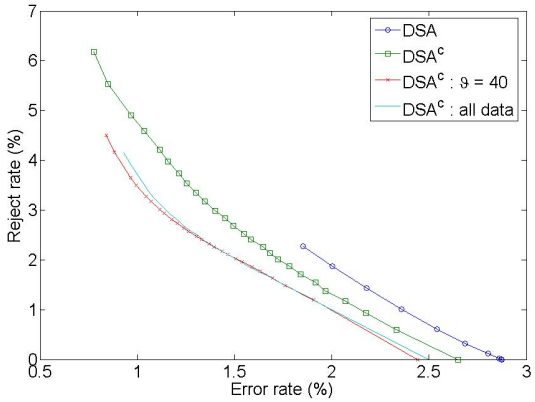


Fig. 5. Impact of a rejection mechanism on *NIST-digits-test1* with DTree classifiers.

use some proposed ideas to reduce the complexity of 1NN classifiers which work in a similar way. In addition, we must evaluate other strategies towards reducing the error rates of this method. This can be achieved by means of using other SMs, by filtering examples from Val, and so forth.

6 Acknowledgments

The authors would like to acknowledge the CAPES-Brazil and NSERC-Canada for the financial support.

References

1. Dos Santos, E.M., Sabourin, R., Maupin, P.: A dynamic overproduce-and-choose strategy for the selection of classifier ensembles. Pattern Recognition **41** (2008) 2993–3009
2. Hansen, L.K., Liisberg, C., Salamon, P.: The error-reject tradeoff. Open Systems & Information Dynamics **4**(2) (1997) 159–184
3. Kuncheva, L.I., Bezder, J.C., Duin, R.P.W.: Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognition **34** (2001) 299–314
4. Oliveira, L.E.S., Sabourin, R., Bortolozzi, F., Suen, C.Y.: Automatic recognition of handwritten numeral strings: A recognition and verification strategy. IEEE Trans. on Pattern Analysis and Machine Intelligence **24**(11) (2002) 1438–1454
5. Ho, T.: The random subspace method for construction decision forests. IEEE Trans. on Pattern Analysis and Machine Intelligence **20** (1998) 832–844