Dynamic Ensemble Selection for Off-Line Signature Verification*

Luana Batista, Eric Granger and Robert Sabourin

Laboratoire d'imagerie, de vision et d'intelligence artificielle
École de technologie supérieure
1100, rue Notre-Dame Ouest, Montréal, QC, H3C 1K3, Canada
lbatista@livia.etsmtl.ca, {eric.granger, robert.sabourin}@etsmtl.ca

Abstract. Although not in widespread use in Signature Verification (SV), the performance of SV systems may be improved by using ensemble of classifiers (EoC). Given a diversified pool of classifiers, the selection of a subset to form an EoC may be performed either statically or dynamically. In this paper, two new dynamic selection (DS) strategies are proposed, namely OP-UNION and OP-ELIMINATE, both based on the K-nearest-oracles. To compare ensemble selection strategies, a hybrid generative-discriminative system for off-line SV system is considered. Experiments performed by using real-world SV data, comprised of genuine samples, and random, simple and skilled forgeries, indicate that the proposed DS strategies achieve a significantly higher level of performance in off-line SV than other well-known DS and static selection (SS) strategies. Improvements are most notable in problems where a significant level of uncertainty emerges due a considerable amount of intra-class variability.

1 Introduction

Signature Verification (SV) systems are relevant in many real-world applications, such as check cashing, credit card transactions and document authentication. In off-line SV, handwritten signatures are transcribed on sheets of paper, and at some later time scanned in order to obtain a digital representation. Given a digitized signature, an off-line SV system typically performs preprocessing, feature extraction and classification to authenticate the signature of an individual.

Handwritten signatures are behavioural biometric traits that are known to incorporate a considerable amount of intra-class variability. Although not in widespread use in off-line SV, a promising way to improving system performance is through ensemble of classifiers (EoC) [1,4]. The motivation of using EoCs stems from the fact that a diverse set of classifiers usually make different errors on input samples. Indeed, when the response of a set of $\mathcal{C}$ classifiers is averaged, the variance contribution in the bias-variance decomposition decreases by $1/\mathcal{C}$, resulting in a smaller expected classification error [9].

* This research has been supported by the Natural Sciences and Engineering Research Council of Canada.

Bagging, boosting and random subspaces are well-known methods for creating diversity among classifiers. While bagging and boosting use different samples subsets to train different classifiers, the random subspace method use different subspaces of the original input feature space. Given a diversified pool of classifiers, an important issue is the selection of a subset to form an EoC, such that the recognition rates are maximized during operations [6]. EoC selection may be performed either statically or dynamically. Based on a set of reference samples not used during training, static selection (SS) strategies select the EoC that provides the best classification rates on that set. Then, this EoC is employed during operations to classify any input sample. Dynamic selection (DS) strategies also need a reference set to select the best EoC, although this task is performed during operations, by taking into account the specific characteristics of a given sample to be classified.

In a pattern recognition system that starts with a limited number of reference samples, it is difficult to define *a priori* a single best EoC for the application. Ideally, the EoC should be continuously adapted whenever new reference samples become available. With DS, this new data can be incorporated to the reference set (after being classified by the pool of classifiers) without any additional step.

KNORA (K-nearest-oracles) is a DS strategy that has been successfully applied to handwritten numeral recognition [6]. For each input sample, the KNORA strategy finds its K-nearest neighbors in the reference set, and then selects the classifiers that have correctly classified those neighbors. Finally, the selected classifiers are combined in order to classify the input sample. The main drawback of KNORA is that a robust set of features must be defined in order to compute similarity between the input sample and the samples in the DS database.

As an alternative, this paper propose two new DS strategies, namely OP-UNION and OP-ELIMINATE, that use the classifier outputs (i.e., the output profile) to find the K-nearest neighbors. To validate the proposed and other reference dynamic and static selection strategies, a multi-classifier generative-discriminative system is considered. In this system, Hidden Markov Models (HMMs) are employed as feature extractors followed by Support Vector Machines (SVMs) as two-class classifiers. Proof-of-concept experiments are carried out on a real-world signature database [4,5,8], comprised of genuine samples, and random, simple and skilled forgeries. The rest of this paper is organized as follows. The next section presents the hybrid generative-discriminative system for off-line SV. Then, Section 3 proposes two new DS strategies. Finally, Section 4 describes the experimental methodology, and Section 5 presents and discusses the experiments.

2 A Hybrid System for Off-line SV

Let $\mathcal{T}^i = I^i_{trn(l)}$, $1 \leq l \leq N$, be the training set used to design a SV system for writer i. The set $\mathcal{T}^i$ contains genuine signature samples supplied by writer i, as well as random forgery samples supplied by other writers not enrolled to the system. For each signature $I^i_{trn(l)}$ in the training set $\mathcal{T}^i$, a set of features is

generated (see Figure 1). First, $I^i_{trn(l)}$ is described by means of pixel densities, which are extracted through a grid composed of retangular cells. Each column of cells j is converted into a low-level feature vector $\mathbf{F}^{\boldsymbol{i}}_{\boldsymbol{j}} = \left\{f^i_{j1}, f^i_{j2}, ...\right\}$, where each vector component $f^i_{jh} \in [0,1]$. These components correspond to the number of black pixels in a cell divided by the total number of pixels of this cell. The signature $I^i_{trn(l)}$ is therefore represented by a set of low-level feature vectors $\boldsymbol{\mathcal{F}^i_{trn(l)}} = \left\{\mathbf{F}^{\boldsymbol{i}}_{\boldsymbol{j}}\right\}$, $1 \leq j \leq col$, where col is the number of columns in the grid.

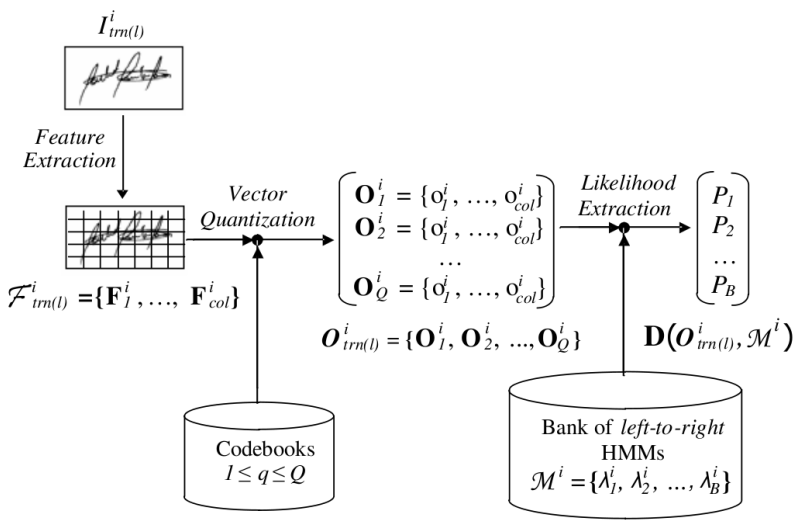

Fig. 1. Design of the generative stage for a specific writer i. After feature extraction, the signature is quantized into $\mathcal{Q}$ different sequences, from which $B > \mathcal{Q}$ likelihoods are obtained.

Then, $\boldsymbol{\mathcal{F}^i_{trn(l)}}$ is quantized into a sequence of discrete observations $\mathbf{O}^{\boldsymbol{i}}_{\boldsymbol{q}} = \left\{o^i_j\right\}$, for $1 \leq j \leq col$. Each observation o^i_j is a symbol provided by the codebook q (generated using the *K-means* algorithm). Since $\mathcal{Q}$ different codebooks are employed per writer i, each training signature $I^i_{trn(l)}$ yields a set of observation sequences $\boldsymbol{O^i_{trn(l)}} = \{\mathbf{O}^i_q\}$, for $1 \leq q \leq \mathcal{Q}$. The set of observation sequences, $\boldsymbol{O^i_{trn(l)}}$, is then input to the bank of *left-to-right* HMMs $\boldsymbol{\mathcal{M}^i} = \left\{\lambda^i_b\right\}$, $1 \leq b \leq B$, from which a high-level feature vector $\mathbf{D}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right) = \{P_1, ..., P_B\}$ is extracted. Each component P_b is a likelihood computed between an observation sequence $\mathbf{O}^i_q$ and a HMM λ^i_b, where λ^i_b can either correspond to the genuine class (i.e., trained with genuine samples from writer i), or to the impostor class (i.e., trained with random forgery samples). It is worth noting that the same

sequences $\boldsymbol{O^i_{trn(l)}}$, $1 \leq l \leq N$, used to obtain the HMM likelihood vectors are also used to train the HMMs in $\boldsymbol{\mathcal{M}^i}$. Appart from the different codebooks, a different number of states is employed to produce a bank of HMMs.

As long HMM likelihood vectors are produced during the design of the generative stage, the random subspace method (RSM) is used to select the input space in which multiple SVMs are trained. For each random subspace r, $1 \leq r \leq \mathcal{R}$, a smaller subset of likelihoods is randomly selected, with replacement, from $\mathbf{D}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right)$, $1 \leq l \leq N$, and used to train a different SVM. During operations, a given input signature I^i_{tst} follows the same steps of feature extraction, vector quantization and likelihood extraction as performed with a training signature, resulting in the HMM likelihood vector $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$. Then, based on previously-classified signature samples – stored in the dynamic selection (DS) database –, the most accurate ensemble of SVMs is dynamically selected from the pool and used to classify $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$(see Figure 2).

As described in Section 3, signature samples selected from the DS database are the K-nearest neighbors of the input sample to be classified. The DS database contains genuine samples supplied by writer i, as well as random forgery samples taken from writers not enrolled to the system. explains the partitioning of each dataset used in this work.

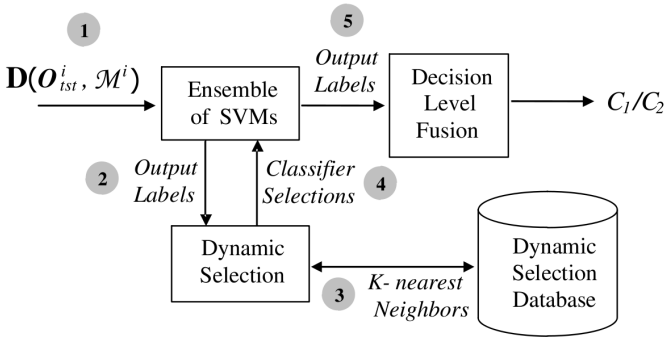

Fig. 2. System architecture of the discriminative stage for a specific writer i.

Bank of HMMs. Let $\boldsymbol{\mathcal{M}^i} = \{\boldsymbol{w_1} \cup \boldsymbol{w_2}\}$ be the bank of HMMs, where $\boldsymbol{w_1}= \{\lambda_1^{(C_1)}, \lambda_2^{(C_1)}, ..., \lambda_R^{(C_1)}\}$ is the set of R HMMs of the genuine class C_1, and $\boldsymbol{w_2}= \{\lambda_1^{(C_2)}, \lambda_2^{(C_2)}, ..., \lambda_S^{(C_2)}\}$ is the set of S HMMs of the impostor's class C_2. Given the set of observation sequences $\boldsymbol{O^i_{trn(l)}} =\{\mathbf{O}^i_1, \mathbf{O}^i_2, ..., \mathbf{O}^i_Q\}$ extracted from a training signature $I^i_{trn(l)}$, the vector $\mathbf{D}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right)$ is obtained by computing the likelihoods of $\boldsymbol{O^i_{trn(l)}}$ for each HMM in $\boldsymbol{\mathcal{M}^i}$, that is,

$$\mathbf{D}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right) = \begin{bmatrix} P(\mathbf{O}^i_q/\lambda_1^{(C_1)}) \\ P(\mathbf{O}^i_q/\lambda_2^{(C_1)}) \\ ... \\ P(\mathbf{O}^i_q/\lambda_S^{(C_2)}) \end{bmatrix} \quad (1)$$

If, for instance, $\lambda_1^{(C_1)}$ and $\lambda_S^{(C_2)}$ are trained with observation sequences extracted from the codebook $q = 10$, a compatible sequence from $\boldsymbol{O^i_{trn(l)}}$, that is, $\mathbf{O}^i_{q=10}$, must be sent to both. Finally, the vector $\mathbf{D}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right)$ is labeled according to the class of $\boldsymbol{O^i_{trn(l)}}$. It is worth noting that, if $\boldsymbol{O^i_{trn(l)}}$ belongs to class C_1, $\mathbf{D}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right)$ should contain higher values in the first R positions and smaller values in the remaining S positions, allowing a two-class classifier to discriminate samples of class C_1 from class C_2.

3 New Strategies for Dynamic Ensemble Selection

Let $\boldsymbol{O^i_{ds(j)}}$, $1 \leq j \leq M$, be the sequences of observations extracted from the DS database of writer i, and $\mathbf{D}\left(\boldsymbol{O^i_{ds(j)}}, \boldsymbol{\mathcal{M}^i}\right)$be their corresponding likelihood vectors [1], for $1 \leq j \leq M$. For each DS vector $\mathbf{D}\left(\boldsymbol{O^i_{ds(j)}}, \boldsymbol{\mathcal{M}^i}\right)$, an output profile (OP) is calculated as follows. First, $\mathbf{D}\left(\boldsymbol{O^i_{ds(j)}}, \boldsymbol{\mathcal{M}^i}\right)$ is input to all SVM classifiers c_r, $r = 1, 2, ..., \mathcal{R}$, in the pool of classifiers $\mathcal{C}$. Each c_r receives as input only the vector positions related to its respective subspace. Then, the resulting output labels are stored as a vector to form a DS output profile, $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{ds(j)}}, \boldsymbol{\mathcal{M}^i}\right)\right)$. This procedure is repeated for all DS vectors, resulting in a set of DS-OPs. For simplicity, it is assumed that the DS-OPs are also supposed to be stored in the DS database.

During operations, when a test vector $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$ is presented to the off-line SV system, four main steps are performed. First, the output profile $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)\right)$ is calculated, as performed for the DS vectors. Second, the Euclidean distance is computed between $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)\right)$ and each DS-OP, in order to find its K-nearest neighbors. Third, the SVMs that correctly classify the K corresponding DS vectors are selected and used to classify $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$. Finally, the SVMs decisions are fused through majority voting. The two following variants of KNORA are proposed to manage output profiles:

1) OP-ELIMINATE. Given the test vector $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$, the objective of this first variant is to find an ensemble of up to K SVMs that simultaneously classify its K-nearest neighbors in the DS database correctly. After obtaining $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)\right)$, its K-nearest DS-OPs, $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{ds(k)}}, \boldsymbol{\mathcal{M}^i}\right)\right)$, $1 \leq k \leq K$, are found via Euclidean distance. For each SVM c_r, $r = 1, 2, ..., \mathcal{R}$, in the

[1] During dynamic selection, $\mathbf{D}\left(\boldsymbol{O^i_{ds(j)}}, \boldsymbol{\mathcal{M}^i}\right)$ is refered as a DS vector.

pool $\mathcal{C}$, the OP-ELIMINATE algorithm verifies if c_r has previously classified all corresponding DS vectors $\mathbf{D}\left(\boldsymbol{O^i_{ds(k)}}, \boldsymbol{\mathcal{M}^i}\right)$, $1 \leq k \leq K$, correctly. If so, c_r is added to the ensemble E; otherwise, the next SVM in the pool is verified. In the case where no classifier ensemble can correctly classify all K DS vectors, the value of K is decreased until at least one SVM can correctly classify a DS vector. Finally, each SVM in the ensemble E submites a vote on the test vector $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$, where final classification label $\mathcal{L}$ is obtained by using the majority vote rule.

2) OP-UNION. Given the test vector $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$ and its K-nearest neighbors in the DS database, the objective of this variant is to find for each neighbor k, $1 \leq k \leq K$, an ensemble of up to K SVMs that correctly classify it. First, the test output profile, $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)\right)$, and its K-nearest DS-OPs, $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{ds(k)}}, \boldsymbol{\mathcal{M}^i}\right)\right)$, $1 \leq k \leq K$, are obtained, such as performed for OP-ELIMINATE. For each neighbor k, and for each SVM c_r, $r = 1, 2, ..., \mathcal{R}$, in the pool $\mathcal{C}$, the OP-UNION algorithm then verifies if c_r has previously classified the DS vector $\mathbf{D}\left(\boldsymbol{O^i_{ds(k)}}, \boldsymbol{\mathcal{M}^i}\right)$ correctly. If so, c_r is added to the ensemble E_k; otherwise, the next SVM in the pool is verified. After applying this procedure to all K-NNs, the SVMs in each ensemble E_k are combined in order to classify the test vector $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$. Finally, the final classification label $\mathcal{L}$ is obtained by using the majority vote rule. Note that a same SVM can give more than one vote if it correctly classifies more than one DS vectors.

4 Experimental Methodology

Brazilian SV database. It contains 7920 samples of signatures that were digitized as 8-bit greyscale images over 400X1000 pixels, at resolution of 300 dpi. The signatures were provided by 168 writers and are organized in two sets: the development database (DB_{dev}) and the exploitation database (DB_{exp}). DB_{exp} contains signature samples from writers enrolled to the system, and is used to model the genuine class. It is composed of 3600 signatures supplied by 60 writers. Each writer i has 40 genuine samples, 10 simple forgeries and 10 skilled forgeries. 20 genuine samples are available for training ($\mathcal{T}^i_{exp(20)}$) and 10 for validation ($\mathcal{V}^i_{exp(10)}$). As test set, each writer i has 10 genuine samples, 10 random forgeries, 10 simple forgeries and 10 skilled forgeries; where the random forgeries are genuine samples randomly selected from other writers in DB_{exp}.

DB_{dev} contains signature samples from writers not enrolled to the system, and is used as *prior* knowledge to design the codebooks and the impostor class. It is composed of 4320 genuine samples supplied by 108 writers. Each writer j has 40 genuine samples, where 20 are available for training ($\mathcal{T}^j_{dev(20)}$) and 10 for validation ($\mathcal{V}^j_{dev(10)}$). The remaining 10 samples, available for test, are not employed in this work. Given a writer i enrolled to the system, DB_{dev} and DB_{exp} are used to compose different datasets employed in different phases of the system design (see Table 1).

The signature images are divided in 62 columns of cells, where each cell is a rectangle composed of 40X16 pixels. To absorb the horizontal variability of the signatures, the images are aligned to the left and the blank cells in the end of the images are discarded.

Table 1. Datasets for a specific writer i, using the Brazilian SV database.

Dataset Name	Task	Genuine Samples	Random Forgery Samples
DB^i_{hmm}	HMM Training	$\mathcal{T}^i_{exp(20)}$ + $\mathcal{V}^i_{exp(10)}$	$\mathcal{T}^{j=1:108}_{dev(20)}$ + $\mathcal{V}^{j=1:108}_{dev(10)}$
DB^i_{svm}	SVM Training	$\mathcal{T}^i_{exp(20)}$	20 from $\mathcal{T}^{j=1:180}_{dev(20)}$
DB^i_{grid}	SVM Grid Search	$\mathcal{V}^i_{exp(10)}$	10 from $\mathcal{V}^{j=1:180}_{dev(10)}$
DB^i_{roc}	ROC Curve		$\mathcal{V}^{j=1:180}_{dev(10)}$
DB^i_{ds}	Dynamic Selection		

HMM Training. 29 different codebooks q ($1 \leq q \leq 29$) are generated by varying the number of clusters from 10 to 150, in steps of 5; using all training and validation signatures of DB_{dev}. Given a writer i and a codebook q, DB^i_{hmm} is employed to train a set of discrete *left-to-right* HMMs with different number of states, using the Baum-Welch algorithm. As the number of states varies from 2 to the lenght of the smallest sequence used for training (L_{min}), the genuine class is composed of a variable number HMMs (i.e., 29x(L_{min}-1)) that depends on the writer's signature size. On the other hand, to compose the impostor class, there are thousands of HMMs taken from the writers in DB_{dev}.

SVM Training. By using the RSM, 100 subspaces composed of 30 dimensions each (i.e., 15 likelihoods randomly selected from each class) are used to train 100 different SVMs (RBF kernel) per writer. For a same writer i, the training set, DB^i_{svm}, remains the same for all 100 SVMs.

Comparison of Techniques. In this paper, the simulation results obtained with OP-UNION/ELIMINATE are compared with KNORA-UNION/ELIMINATE [6], with the standard combination of all classifiers, and with Decision Templates (DT) [7] – a well-known DS method in the multi-classifier system (MCS) community. With OP-UNION/ELIMINATE, the search for the K-nearest neighbors is performed using the output labels provided by all 100 SVMs; while with KNORA-UNION/ELIMINATE, only the SVM input subspace providing the lowest error rates on DB^i_{ds} is used during the search. The value of K is defined as being half of the number of genuine samples in DB^i_{ds}, that is, 5. Comparisons are performed as well with two reference systems proposed in our previous research, that is, (i) a traditional generative system based on HMMs [2] (refered in this paper as baseline system), and (ii) a hybrid system based on

the static selection of generative-discriminative ensembles [3]. In both systems, HMMs are trained by using $\mathcal{T}^i_{exp(20)} + \mathcal{V}^i_{exp(10)}$, from the Brazilian SV database.

Performance Evaluation. The overall system performance is measured through different operating points of an averaged ROC curve. To obtain this curve using DB^i_{roc}, the operating points $\{TPR_i, FPR_i\}$ related to different users, $i = 1, .., N$, are averaged if they have a same true negative rate (TNR, or $1 - FPR$) – refered in this paper as γ. The average error rate (AER) indicates the total error of the system for a specific γ, where the false negative rate (FNR) and the false positive rate (FPR) are averaged taking into account the *a priori* probabilities. In the experiments, the FPR is calculated with respect to three forgery types: random, simple and skilled.

5 Simulation Results

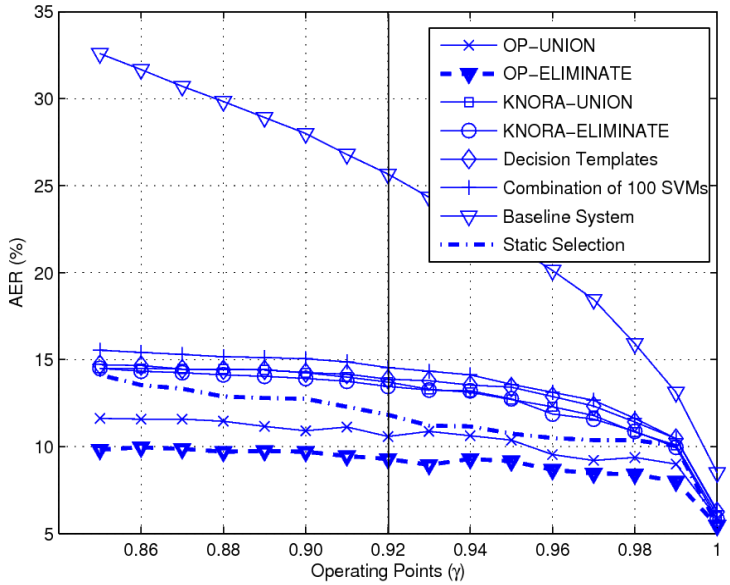

Fig. 3. *AERs versus* operating points (γ) obtained on Brazilian test data using off-line SV systems that employ different ensemble selection techniques.

Figure 3 presents the $AERs$ curves obtained on test data (DB^i_{tst}) as function of operating points (γ). These results indicate that the proposed DS strategies, i.e., OP-UNION and OP-ELIMINATE, provided the lowest $AERs$, demonstrating the advantage of using a DS approach based on output profiles – as opposed

to KNORA, where the input feature space is used to find the K-nearest DS samples. It is also beneficial to employ EoCs composed of a small set of base classifiers – in contrast to DTs and to the standard technique of combination of classifiers, where all base classifiers in the pool are part of the ensemble.

Table 2. Overall error rates (%) obtained on test data for $\gamma = 0.92$.

Method	FNR	FPR_{random}	FPR_{simple}	$FPR_{skilled}$	AER
OP-UNION	**4.17**	**1.50**	**3.50**	**33.17**	**10.58**
OP-ELIMINATE	**5.17**	**1.00**	**2.83**	**28.17**	**9.29**
KNORA-UNION	2.17	2.50	6.83	43.33	13.71
KNORA-ELIMINATE	2.33	2.67	6.33	42.50	13.46
Decision Templates	2.67	2.17	6.33	44.33	13.88
Combination of 100 SVMs	2.33	2.67	7.33	45.83	14.54
Static Selection [3]	2.17	5.67	5.00	34.50	11.83
Baseline [2]	0.33	9.83	18.17	74.33	25.67

Table 3. Overall error rates (%) compared to reference SV systems from literature.

Method	FNR	FPR_{random}	FPR_{simple}	$FPR_{skilled}$	AER
Batista et al. [2]	9.83	0.00	1.00	20.33	7.79
Bertolini et al. [4]	11.32	4.32	3.00	6.48	6.28
Justino et al. [5]	2.17	1.23	3.17	36.57	7.87
Santos et al. [8]	10.33	4.41	1.67	15.67	8.02
OP-UNION ($\gamma = 1.0$)	**8.17**	**0.67**	**0.67**	**14.00**	**5.88**
OP-ELIMINATE ($\gamma = 1.0$)	**7.50**	**0.33**	**0.50**	**13.50**	**5.46**

OP-UNION and OP-ELIMINATE strategies also achieved $AERs$ that are lower than those obtained with SS. This represents a situation where DS is superior to SS, i.e., in a problem where a significant level of uncertainty emerges due a considerable amount of intra-class variability. Finally, the lowest performance of the baseline system is obtained because a pure generative approach is adopted for system design, i.e., only the genuine class is modeled, and a single HMM is employed per writer.

Table 2 presents the overall results for $\gamma = 0.92$. Note that the proposed DS strategies provide lower $FPRs$ at the expense of higher $FNRs$. In practice, the trade-off between FPR and FNR can be adjusted according to the risk linked to an input sample. In banking applications, for example, the decision to use a specific operating point may be associated with the amount of the check. In the simplest case, for a user that rarely signs high value checks, big amounts would require operating points related to low $FPRs$, such as would be provided by a

γ close to 1; while lower amounts would require operating points related to low $FNRs$, since the user would not feel comfortable with frequent rejections.

Finally, Table 3 presents a comparison with other systems designed with the Brazilian SV database. By assuming that the objective of these systems is to minimize the AER, the comparison is performed with $\gamma = 1$.

6 Conclusions

In this paper, two new DS strategies based on KNORA, namely OP-UNION and OP-ELIMINATE, are proposed to improve performance of off-line SV systems. These strategies employ the classifier outputs (i.e., the output profile), instead of the input feature space, to find the most accurate EoC for a given input sample. To compare ensemble selection strategies, a hybrid off-line SV system is considered. In this system, HMMs are employed as feature extractors followed by SVMs as two-class classifiers.

Experiments performed using real world signature data indicate that OP-UNION/ELIMINATE can achieve a higher level of accuracy in off-line SV than other reference DS and SS strategies. This is especially true in problems where a significant level of uncertainty emerges due a considerable amount of intra-class variability. Future work consists of investigating the adaptive capabilities of the proposed strategies for incremental learning of new reference samples.

References

1. R. Bajaj and S. Chaudhury. Signature verification using multiple neural classifiers. *Pattern Recognition*, 30:1–7, 1997.
2. L. Batista, E. Granger, and R. Sabourin. Improving performance of hmm-based off-line signature verification systems through a multi-hypothesis approach. *International Journal on Document Analysis and Recognition*, 13:33–47, March 2010.
3. L. Batista, E. Granger, and R. Sabourin. A multi-classifier system for off-line signature verification based on dissimilarity representation. In *Ninth International Workshop on Multiple Clasifier System*, pages 264–273, April 2010.
4. D. Bertolini, L. Oliveira, E. Justino, and R. Sabourin. Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. *Pattern Recognition*, 43:387–396, 2010.
5. E. Justino, F. Bortolozzi, and R. Sabourin. Off-line signature verification using hmm for random, simple and skilled forgeries. In *International Conference on Document Analysis and Recognition*, pages 105–110, 2001.
6. A. Ko, R. Sabourin, and A. Britto. From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, 41(5):1718–1731, 2008.
7. L. Kuncheva, J. Bezdek, and R. Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34:299–314, 2001.
8. C. Santos, E. Justino, F. Bortolozzi, and R. Sabourin. An off-line signature verification method based on the questioned document expert's approach and a neural network classifier. In *International Workshop on Frontiers in Handwriting Recognition*, pages 498–502, 2004.
9. K. Tumer and J. Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, 29(2):341 – 348, 1996.