www.elsevier.com/locate/pr

Dynamic selection of generative-discriminative ensembles for off-line signature verification

Luana Batista, Eric Granger, Robert Sabourin

PII: S0031-3203(11)00435-3
DOI: doi:10.1016/j.patcog.2011.10.011
Reference: PR 4299

To appear in: *Pattern Recognition*

Received date: 22 December 2010
Revised date: 26 September 2011
Accepted date: 21 October 2011

Cite this article as: Luana Batista, Eric Granger and Robert Sabourin, Dynamic selection of generative-discriminative ensembles for off-line signature verification, *Pattern Recognition*, doi:10.1016/j.patcog.2011.10.011

Dynamic Selection of Generative-Discriminative Ensembles for Off-Line Signature Verification

Luana Batista, Eric Granger, Robert Sabourin

Laboratoire d'imagerie, de vision et d'intelligence artificielle,
École de technologie supérieure
1100, rue Notre-Dame Ouest, Montreál, QC, H3C 1K3, Canada

Abstract

In practice, each writer provides only a limited number of signature samples to design a signature verification (SV) system. Hybrid generative-discriminative ensembles of classifiers (EoCs) are proposed in this paper to design an off-line SV system from few samples, where the classifier selection process is performed dynamically. To design the generative stage, multiple discrete *left-to-right* Hidden Markov Models (HMMs) are trained using a different number of states and codebook sizes, allowing the system to learn signatures at different levels of perception. To design the discriminative stage, HMM likelihoods are measured for each training signature, and assembled into feature vectors that are used to train a diversified pool of two-class classifiers through a specialized Random Subspace Method. During verification, a new dynamic selection strategy based on the K-nearest-oracles (KNORA) algorithm and on Output Profiles selects the most accurate EoCs to classify a given input signature. This SV system is suitable for incremental learning of new signature samples. Experiments performed with real-world signature data (comprised of genuine samples, and random, simple and skilled forgeries) indicate that the proposed dynamic selection strategy can significantly reduce the overall error rates, with respect to other EoCs formed using well-known dynamic and static selection strategies. Moreover, the performance of the SV system proposed in this paper is significantly greater than or comparable to that of related systems found in the literature.

Keywords: Off-line Signature Verification, Ensemble of Classifiers, Dynamic Selection, Hybrid Generative-Discriminative Systems, Hidden Markov Models, Incremental Learning.

Email addresses: lbatista@livia.etsmtl.ca (Luana Batista), eric.granger@etsmtl.ca (Eric Granger), robert.sabourin@etsmtl.ca (Robert Sabourin)

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. For complete and recent surveys of off-line SV, the reader is referred to [1, 2]. The Hidden Markov Models (HMMs) [3] have been successfully employed for classification due the sequencial nature and variable size of the signature data [4, 5, 6, 7]. In particular, the *left-to-right* topology of HMMs is well adapted to the dynamic characteristics of European and American handwriting, in which the hand movements are always from left to right.

Handwriting signatures are behavioural biometric traits that are known to incorporate a considerable amount of intra-class variability. Figure 1 presents the superimposition of several signature skeletons samples of the same writer. Note that the intrapersonal variability occurs mostly in the horizontal direction, since there is normally more space to sign in this direction. By using a grid segmentation scheme adapted to the signature size, Rigoll and Kosmala [7], and later Justino [6], have shown that HMMs are suitable for modeling the variabilities observed among signature samples of a same writer.

Figure 1: Example of several superimposed signature samples of the same writer.

Since the HMM is a generative classifier [8], it requires a considerable amount of training data to achieve a high level of performance. Unfortunately, acquiring signature samples for the design of off-line SV systems is a costly and time consuming process (for instance, in banking transactions, a client is asked to supply between 3 to 5 signatures samples at the time of his/her subscription). A related problem regards the generation of codebooks[1] needed to design discret HMMs. Typically, the data used to generate codebooks

[1] A codebook contains a set of symbols, each one associated with a cluster of feature vectors, used to generate sequences of discrete observations in discrete HMM-based systems.

are the same data that are employed to train the HMMs [5, 7]. The main drawback of this strategy is the need to reconstruct the codebook whenever a new writer is added to the system. Moreover, this strategy has been shown to yield poor system performance when few signature samples are available [9].

In this paper, the problem of having a limited amount of genuine signature samples is addressed by designing a hybrid off-line SV system based on the dynamic selection of generative-discriminative ensembles. To design the generative stage, multiple discrete *left-to-right* HMMs are trained using a different number of states and codebook sizes, allowing the system to learn signatures at different levels of perception. The codebooks are generated using signature samples of an independent database (also called development database), supplied by writers not enrolled to the SV system. This *prior* knowledge ensures that the SV system can be deployed even when a single user is enrolled. To design the discriminative stage, HMM likelihoods are measured for each training signature, and assembled into feature vectors that are used to train a diversified pool of two-class classifiers through a specialized Random Subspace Method.

Given a test signature during verification, the most accurate subset of classifiers is selected to form an EoC using a dynamic selection strategy based on K-nearest-oracles (KNORA) [10] and on Output Profiles [11]. As opposed to static selection, where a single ensemble of classifiers (EoC) is selected before operations, and applied to all input samples, dynamic selection allows for a different selection of EoCs according to each input sample. Moreover, when new reference samples become available, they can be incorporated to the system, incrementally, to improve the selection of the most adequate EoC.

To validate the proposed SV system, proof-of-concept experiments are carried out on real-world signature data from two datasets, namely, the Brazilian SV database [4] (comprised of genuine samples, and random, simple and skilled forgeries) and the GPDS database [12] (comprised of genuine samples, and random and skilled forgeries). The performance of the generative-discriminative ensembles formed with the proposed dynamic selection strategy is compared to that of other well-know dynamic and static selection strategies, with a traditional system based on HMMs, and with other relevant SV systems found in the literature. Moreover, the adaptive properties of the proposed SV system for incremental learning of new signature samples are investigated.

The rest of this paper is organized as follows. The next section briefly presents the state-of-the-art on hybrid generative-discriminative classifiers and on ensemble of classifiers. Section 3 presents the hybrid generative-discriminative off-line SV system, as well as the proposal of a new dynamic selection strategy. Section 4 describes the experimental methodology, including datasets, training protocol and measures used to evaluate system performance. Finally, the experiments are presented and discussed in Section 5.

2. Hybrid Generative-Discriminative Ensembles

Generative classifers differs from discriminative ones in that they can reproduce an input pattern in addition to recognizing it. A generative classifier learns the full joint distribution of a class, i.e., a model of the joint probability $P(X|Y)$, of the inputs X and the label Y, and may generate labeled instances according to this distribution. Prediction is performed via the Bayes rule to compute $P(Y = y_j|X = x_i)$, and then by assigning x_i to the most likely y_j. In contrast, a discriminative classifier models the decision boundary between class distributions by learning the posterior probability $P(Y|X)$ directly, or by learning a direct map from inputs X to the class labels [13, 14].

Despite the success of HMMs in SV, several important systems have been developed with discriminative classifiers [1, 2]. In fact, both generative and discriminative paradigms hold advantages and drawbacks. In classification problems, literature states that learning the full distribution $P(X|Y)$ is unnecessary. According to Vapnik [15], "one should solve the classification problem directly and never solve a more general problem as an intermediate step such as modeling $P(X|Y)$". Indeed, discriminative classifiers have been favored over generative ones in many pattern recognition problems due their low asymptotic error [16], although comparisons between both paradigms have shown that a discriminative classifier does not necessarily yield better performance [14, 17]. Moreover, generative classifiers may handle missing data [18], novelty detection, and supervised, unsupervised and incremental training more easily, since class densities are considered separately one from another [17]. It is therefore easy to add and remove classes as the operational environment unfolds.

Some hybrid approaches found in literature appear promising to exploit both generative and discriminative paradigms. In the hybrid handwritten 10-digit recognition system proposed by Abou-Moustafa et al. [16], a set of 20 discrete HMMs (two per class) is used to map the variable-length input patterns into single fixed-size likelihood vectors. In the classification stage, these vectors are presented to 10 SVMs (one per class) that provide the final decision through the one-against-all strategy. With a similar hybrid architecture, Bicego et al. [19] proposed a system for 2D-shape/face recognition where each sample of a class is modeled by a continuous HMM. This type of architecture can be viewed as a dissimilarity representation approach, in which input patterns are described by their distance with respect to a predetermined set of prototypes [20, 21]. Therefore, while the HMMs model a set of prototypes, the likelihoods provide similarity measures that define a new input feature space. This new space of similarities can, in principle, be used to train any discriminative classifier. The fact that two patterns x_1 and x_2 present similar degrees of similarity with respect to several HMMs enforces the hypothesis that x_1 and x_2 belong to the same class [19]. In a pure generative approach, an input pattern x_1 would be assigned to the most similar class model, neglecting all the information provided by a space of (dis)similarities (i.e., the distances with respect to the other classes).

The hybrid system achitectures presented in [16, 19] are particularly relevant for SV since they allow to model not only the genuine class, but also the impostor class. Traditional SV approches based on HMMs generally use only genuine signatures to train the system. Then, a decision threshold is defined by using a validation set composed of genuine and random forgery samples (in practice, only random forgeries are available during the design of a SV system).

Ensembles of classifiers (EoCs) have been used to reduce error rates of many challenging pattern recognition problems, including SV [22, 23, 24, 25]. The motivation of using EoCs stems from the fact that different classifiers usually make different errors on different samples. Indeed, it has been shown that, 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 [26, 27].

Bagging [28], Boosting [29] and Random Subspaces [30] are well-known methods for creating diverse 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. The Random Subspace Method is, therefore, well-suited for generating a pool of classifiers in applications that must deal with a limited number of training samples. While many classification methods suffer from the curse of dimensionality, large amounts of features can be exploited by the Random Subspace Method to improve the system performance [30].

Given a pool of classifiers, an important issue is the selection of a diversified subset of classifiers to form an EoC, such that the recognition rates are maximized during operations [10]. This task may be performed either statically or dynamically. Given a set of reference samples (generally not used to train the classifiers), a static selection approach selects the EoC that provides the best classification rates on that set. Then, this EoC is used during operations to classify any input sample. dynamic selection also needs a reference set to select the best EoC; however, this task is performed on-line, by taking into account the specific characteristics of a given sample to be classified. The KNORA strategy [10], for instance, finds for each input sample 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.

In a biometric 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 dynamic selection, this new data can be incorporated to the reference set (after being classified by the pool of classifiers) without any additional step.

3. A System for dynamic selection of Generative-Discriminative Ensembles

In this section, a hybrid generative-discriminative multi-classifier system is proposed for off-line SV. It consists of two stages – a generative stage that provides feature vectors for input patterns using a bank of HMMs; and a discriminative stage that classifies these feature vectors using an ensemble of two-class classifiers.

3.1. System Overview

Let $\mathcal{T}^i = I^i_{trn(l)}$, for $1 \leq l \leq N$, 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 2). 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 $\boldsymbol{\mathcal{F}^i_{trn(l)}} = \left\{\mathbf{F}^{\boldsymbol{i}}_{\boldsymbol{j}}\right\}$, for $1 \leq j \leq col$, where col is the number of columns in the grid.

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 $\boldsymbol{\mathcal{M}^i}= \left\{\lambda^i_b\right\}$, for $1 \leq b \leq B$, 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 $\boldsymbol{O^i_{trn(l)}}$, for $1 \leq l \leq N$, 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, a specialized Random Subspace Method is used to select the input space in which multiple two-class classifiers are trained. For each random subspace r, $1 \leq r \leq \mathcal{R}$, a smaller subset of likelihoods is randomly selected, with replacement, from $\mathbf{D}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right)$, for $1 \leq l \leq N$, and used to train a different classifier. During verification, 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 likelihood vector $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$(see Figure 3). Then, based on previously-classified signature samples – stored in the dynamic selection database –, the most accurate ensemble of classifiers is dynamically selected and used to classify the input likelihood vector. Such as the training set, the dynamic selection database contains genuine signature samples supplied by writer i, as well as random forgery samples taken from the developement database. Section 4 explains the partitioning of each dataset used in this work.

The dynamic selection strategy proposed in this paper is based on the K-nearest-oracles (KNORA)

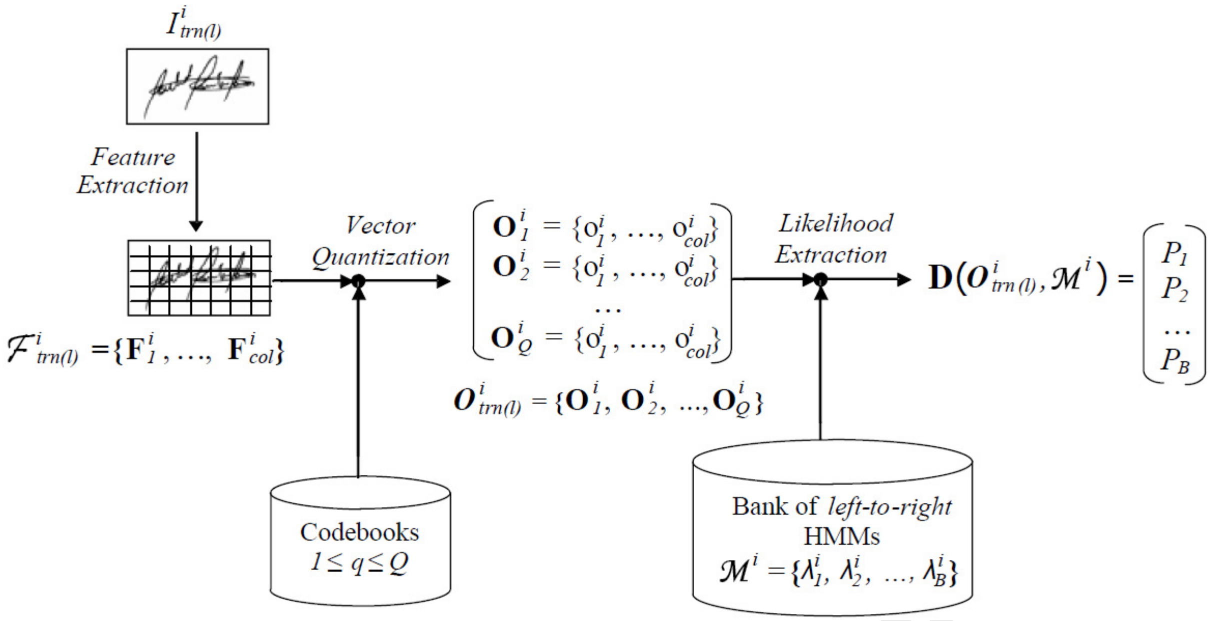

Figure 2: Design of the generative stage for a specific writer i.

(briefly described in Section 2), which has been successfully applied to handwritten-numeral recognition [10]. 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 dynamic selection database. As an alternative, the strategy proposed in this paper inputs the likelihood vector to all classifiers in the pool, and the resulting output labels are used to find the K-nearest neighbors in the dynamic selection database. Then, the classifiers that have correctly classified those neighbors are selected to classify the input likelihood vector.

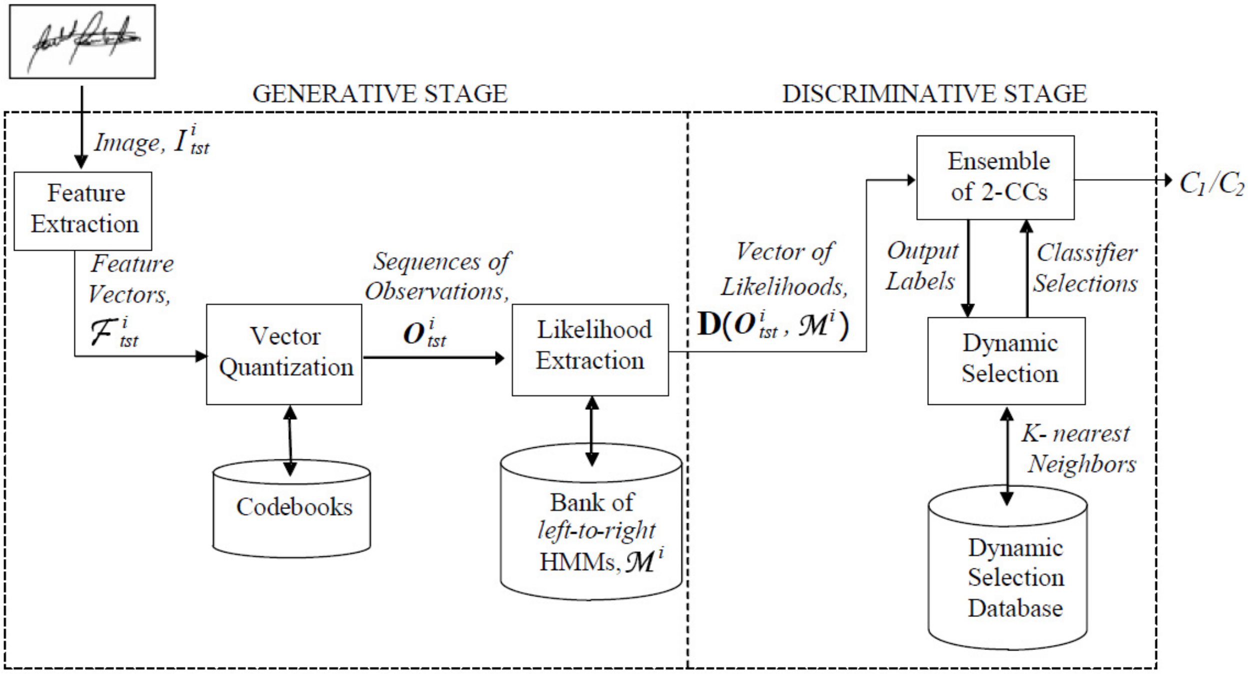

Figure 3: Entire hybrid generative-discriminative system employed during verification (for a specific writer i).

The rest of this section presents additional details on the bank of HMMs, the specialized algorithms to generate random subspaces and to perform dynamic selection of classifiers, and a complexity analysis of different components of the system.

3.2. Bank of HMMs

Let $\boldsymbol{\mathcal{M}^i} = \{\boldsymbol{w_1} \cup \boldsymbol{w_2}\}$ be the bank of *left-to-right* 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. Each HMM in $\boldsymbol{w_1}$ is trained on genuine signature sequences of a specific writer i by using a different number of states. In a similar manner, the HMMs in $\boldsymbol{w_2}$ are trained on random forgery sequences, that is, genuine signature sequences from writers not enrolled to the system. Besides the different number of states, different codebooks are used, allowing the system to learn a signature at different levels of perception. Section 4.3 presents the training strategy adopted for the HMMs.

Once the bank of HMMs is obtained, it is used to extract likelihood vectors (see Figure 4). Given the set of observation sequences $\boldsymbol{O^i_{trn(l)}}$ $=\{\mathbf{O}_1^i, \mathbf{O}_2^i, ..., \mathbf{O}_Q^i\}$ 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,

$$\mathbf{D}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right) = \begin{bmatrix} P(\mathbf{O}^i_q/\lambda^{(C_1)}_1) \\ P(\mathbf{O}^i_q/\lambda^{(C_1)}_2) \\ \cdots \\ P(\mathbf{O}^i_q/\lambda^{(C_1)}_R) \\ P(\mathbf{O}^i_q/\lambda^{(C_2)}_1) \\ P(\mathbf{O}^i_q/\lambda^{(C_2)}_2) \\ \cdots \\ P(\mathbf{O}^i_q/\lambda^{(C_2)}_S) \end{bmatrix} \tag{1}$$

If, for instance, $\lambda^{(C_1)}_1$ and $\lambda^{(C_2)}_S$ were 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 likelihood vector 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, the likelihood vector should contain higher values in the first R positions and lower values in the remaining S positions. If $\boldsymbol{O^i_{trn(l)}}$ belongs to class C_2, both R and S positions should contain low values. This allows a two-class classifier to discriminate samples of class C_1 from class C_2.

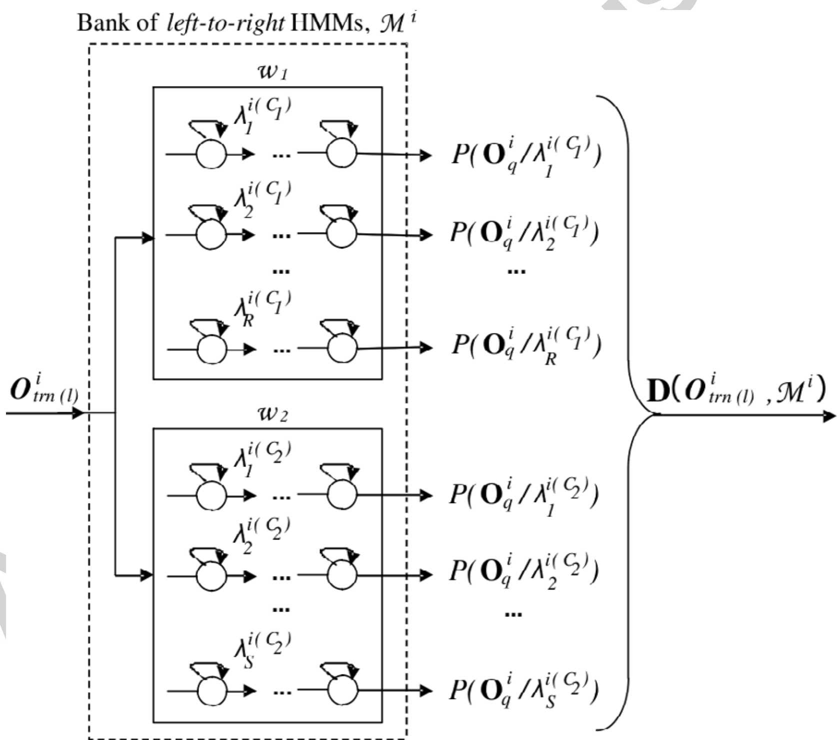

Figure 4: Bank of *left-to-right* HMMs used to extract a vector of likelihoods.

This procedure is performed on all $\boldsymbol{O^i_{trn(l)}}$, $1 \leq l \leq N$, and the resulting likelihood vectors $\mathbf{D}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right)$,

$1 \leq l \leq N$, are used to train a pool of two-class classifiers in the discriminative stage.

3.3. A Random Subspace Method for Two-Class Classifiers

Let $\boldsymbol{O^i_{trn(l)}}$, for $1 \leq l \leq N$, be the sequences of observations extracted from the training signatures of writer i and $\mathbf{D}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right)$, for $1 \leq l \leq N$, be their corresponding likelihood vectors – refered in this section as training vectors. From the first training vector, that is, $\mathbf{D}\left(\boldsymbol{O^i_{trn(1)}}, \boldsymbol{\mathcal{M}^i}\right)$, the specialized Random Subspace Method selects, with replacement, R' likelihoods from its R first positions (corresponding to $\boldsymbol{w_1}$), and S' likelihoods from its S last positions (corresponding to $\boldsymbol{w_2}$). Then, for each training vector $\mathbf{D}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right)$, for $1 \leq l \leq N$, the selected positions, R' and S', are used to form a new vector $\mathbf{D'}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right)$, which is stored in the training set $\mathcal{T}'$. Finally, the vectors $\mathbf{D'}\left(\boldsymbol{O^i_{trn(l)}}, \boldsymbol{\mathcal{M}^i}\right)$, for $1 \leq l \leq N$, in $\mathcal{T}'$, are used to train a two-class classifier c_r, where r, $1 \leq r \leq \mathcal{R}$, is the actual random subspace. This procedure is repeated for $\mathcal{R}$ random subspaces, resulting in a pool $\mathcal{C}$ of $\mathcal{R}$ different classifiers.

3.4. A New Strategy for Dynamic Ensemble Selection

Let $\boldsymbol{O^i_{ds(j)}}$, for $1 \leq j \leq M$, be the sequences of observations extracted from the Dynamic Selection database of writer i, and $\mathbf{D}\left(\boldsymbol{O^i_{ds(j)}}, \boldsymbol{\mathcal{M}^i}\right)$ be their corresponding likelihood vectors [2], 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, the DS vector is input to all 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 stored in the dynamic selection database.

During verification, 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 (see Figure 5). 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 classifiers that are able to classify the K corresponding DS vectors correctly are selected and used to classify the test vector. Finally, the classifier decisions are fused through majority voting.

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

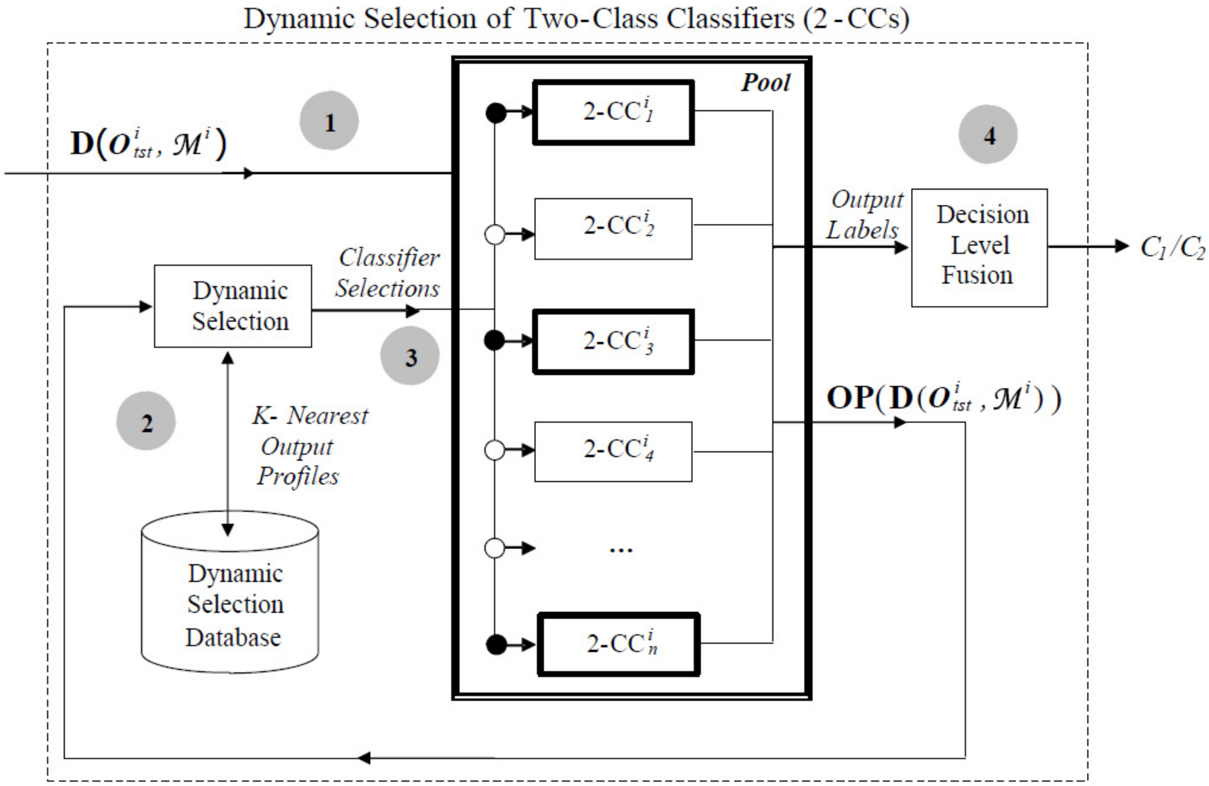

Figure 5: Illustration of the dynamic selection process for two-class classifiers (2-CCs) based on Output Profiles. In this example, 2-CC$_1^i$, 2-CC$_3^i$ and 2-CC$_n^i$ are selected.

In order to select the most accurate EoCs to classify a given test vector, the strategy above is implemented as two variants of KNORA, namely, OP-ELIMINATE and OP-UNION. They are defined as follows:

OP-ELIMINATE (see Algorithm 1). Given the test vector $\mathbf{D}\left(\boldsymbol{O}^{\boldsymbol{i}}_{\boldsymbol{tst}}, \boldsymbol{\mathcal{M}}^{\boldsymbol{i}}\right)$, the objective of this first variant is to find an ensemble of up to K classifiers that simultaneously classify its K-nearest neighbors in the dynamic selection database correctly. First, the test Output Profile, $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O}^{\boldsymbol{i}}_{\boldsymbol{tst}}, \boldsymbol{\mathcal{M}}^{\boldsymbol{i}}\right)\right)$, is calculated. Its K-nearest DS-OPs, $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O}^{\boldsymbol{i}}_{\boldsymbol{ds(k)}}, \boldsymbol{\mathcal{M}}^{\boldsymbol{i}}\right)\right)$, $1 \leq k \leq K$, are then obtained by using the Euclidean distance. For each classifier c_r, $r = 1, 2, ..., \mathcal{R}$, in the pool $\mathcal{C}$, the OP-ELIMINATE algorithm verifies if c_r is able to classify the K corresponding DS vectors $\mathbf{D}\left(\boldsymbol{O}^{\boldsymbol{i}}_{\boldsymbol{ds(k)}}, \boldsymbol{\mathcal{M}}^{\boldsymbol{i}}\right)$, $1 \leq k \leq K$, correctly. If so, c_r is added to the ensemble E; otherwise, the next classifier 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 classifier can correctly classify one DS vector. Finally, each classifier in the ensemble E submites a vote on the test vector, where final classification label $\mathcal{L}$ is obtained by using the majority vote rule.

Algorithm 1 : OP-ELIMINATE

Inputs:
– the number of nearest neighbors, K
– the input vector, $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$
– the number of random subspaces, $\mathcal{R}$
– the pool of classifiers, $\mathcal{C}$
– the Output Profiles, $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{ds(j)}}, \boldsymbol{\mathcal{M}^i}\right)\right)$, $1 \leq j \leq M$

Outputs:
– the final classification label, $\mathcal{L}$

STEP 1:
calculate the Output Profile $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)\right)$
STEP 2:
find its K nearest Output Profiles by calculating the Euclidean distance between $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)\right)$ and each $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{ds(j)}}, \boldsymbol{\mathcal{M}^i}\right)\right)$, where $1 \leq j \leq M$
STEP 3:
set $count = 1$;   // number of classifiers added to the ensemble
for each classifier c_r, $r = 1, 2, ..., \mathcal{R}$, in $\mathcal{C}$ **do**
  if classifier c_r classifies all $\mathbf{D}\left(\boldsymbol{O^i_{ds(k)}}, \boldsymbol{\mathcal{M}^i}\right)$, $1 \leq k \leq K$, correctly **and** $count \leq K$ **then**
    store c_r in the ensemble E, that is, $E(count) = c_r$;
    increment the variable *count*;
  end if
end for
STEP 4:
if size$(E) == 0$ **then**
  decrement K
  repeat the process from STEP 3
end if
STEP 5:
use the classifiers in E to classify $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$
by majority voting, return the final classification label $\mathcal{L}$

OP-UNION (see Algorithm 2). Given the test vector $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$ and its K-nearest neighbors in the dynamic selection database, the objective of this second variant is to find for each neighbor k, $1 \leq k \leq K$, an ensemble of up to K classifiers 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 classifier c_r, $r = 1, 2, ..., \mathcal{R}$, in the pool $\mathcal{C}$, the OP-UNION algorithm then verifies if c_r is able to classify 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 classifier in the pool is verified. After applying this procedure to all K-nearest neighbors, the classifiers in each ensemble E_k are combined in order to classify the test vector. Finally, the final classification label $\mathcal{L}$ is obtained by using the majority vote rule. Note that a same classifier can give more than one vote if it correctly classifies more than one DS vectors.

Algorithm 2 : OP-UNION

Inputs:
– the number of nearest neighbors, K
– the input vector, $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$
– the number of random subspaces, $\mathcal{R}$
– the pool of classifiers, $\mathcal{C}$
– the Output Profiles, $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{ds(j)}}, \boldsymbol{\mathcal{M}^i}\right)\right)$, $1 \leq j \leq M$

Outputs:
– the final classification label, $\mathcal{L}$

STEP 1:
calculate the Output Profile $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)\right)$
STEP 2:
find its K nearest Output Profiles by calculating the Euclidean distance between $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)\right)$ and each $\mathbf{OP}\left(\mathbf{D}\left(\boldsymbol{O^i_{ds(j)}}, \boldsymbol{\mathcal{M}^i}\right)\right)$, where $1 \leq j \leq M$
STEP 3:
for each neighbor k, $1 \leq k \leq K$, **do**
  set *count* $= 1$; // number of classifier added to the ensemble
  for each classifier c_r, $r = 1, 2, ..., \mathcal{R}$, in $\mathcal{C}$ **do**
    if classifier c_r classifies $\mathbf{D}\left(\boldsymbol{O^i_{ds(k)}}, \boldsymbol{\mathcal{M}^i}\right)$ correctly **and** *count* $\leq K$ **then**
      store c_r in the ensemble E_k, that is, $E_k(count) = c_r$
      increment the variable *count*;
    end if
  end for
end for
STEP 4:
use all E_k, $1 \leq k \leq K$, to classify $\mathbf{D}\left(\boldsymbol{O^i_{tst}}, \boldsymbol{\mathcal{M}^i}\right)$
return the final classification label $\mathcal{L}$ by majority voting

3.5. Complexity Analysis

The following complexity analysis considers the use of the Forward-Backward algorithm [3] for HMMs employed in the generative stage, and Support Vector Machines (SVM) with Radial Basis Function (RBF) kernel [31] in the discriminative stage. Assume a *left-to-right* HMM with $\mathcal{S}$ states, in which only transitions between two consecutive states are allowed, and a sequence of length L. The Forward-Backward algorithm has a complexity of $\mathcal{O}(L\mathcal{S})$ per iteration, in terms of both time and memory [32, 33]. For the SVM-RBF, the time and memory complexity during training is $\mathcal{O}(\mathcal{D}N^2)$, where N is the size of the training set and $\mathcal{D}$ is the number of input dimensions. During operations, the evaluation of each input sample has a time and memory complexity of $\mathcal{O}(\mathcal{D}V)$, where V is the number of support vectors [31, 34].

Table 1 presents the overall time complexities associated with the generative and discriminative stages, where the proposed SV system is composed of a bank of B HMMs and a pool of $\mathcal{R}$ SVMs, respectively. In the discriminative stage, α genuine samples *vs.* β forgery samples are used to train the SVMs (i.e., $N = \alpha + \beta$), while only α genuine samples are used to train each HMM in the generative stage. The dynamic selection strategies proposed in this paper are applied only during operations, where K is the number of nearest Output Profiles. Note that the Output Profiles are obtained from the pool of SVMs, whose complexity is shown in the second column.

Table 1: Time complexities of the generative and discriminative stages.

Phase	Bank of HMMs	Pool of SVMs	OP-ELIM./UNION
Training	$\mathcal{O}(\alpha BL\mathcal{S})$	$\mathcal{O}(\mathcal{R}\mathcal{D}N^2)$	-
Test	$\mathcal{O}(BL\mathcal{S})$	$\mathcal{O}(\mathcal{R}\mathcal{D}V)$	$\mathcal{O}(K\mathcal{R})$

4. Experimental Methodology

Given the generative-discriminative system proposed in Section 3, two scenarios are investigated:

- **Scenario 1 – abundant data.** A considerable number of genuine signatures per writer (i.e., 30) is assumed to be available to design an off-line SV system. The objective of this scenario is to analyze the impact of using the proposed dynamic selection strategies over other relevant ensemble selection strategies found in the literature.

- **Scenario 2 – sparse data.** A limited number of genuine signatures per writer is assumed to be available to design an off-line SV system. The objective of this scenario is to apply the proposed system to two realistic cases of SV. In the first case, three different systems are designed, each one with 4, 8 and 12 genuine signatures per writer. In the second case, a system is initially designed with 4 genuine signatures per writer, and new genuine samples available overtime are incrementaly added to system, without retraining the actual classifiers.

The rest of this section describes the signature databases, the grid segmentation scheme, the classifier training specifications and the performance evaluation method used in the experiments.

4.1. Off-line SV Databases

Two different off-line signature databases are used for proof-of-concept computer simulations: the Brazilian SV database, used by our research group [4, 6, 23, 35], and the GPDS database [12], used by other researchers [5, 36, 37, 38]. While the Brazilian SV database is composed of random, simple and skilled forgeries, the GPDS database is composed of random and skilled forgeries. A random forgery is usually a genuine signature sample belonging to a different writer. It is produced when the forger has no access to the genuine samples, not even the writer's name. In the case of simple forgeries, only the writer's name is known. Thus, the forger reproduces the signature in his/her own style. Finally, a skilled forgery represents a reasonable imitation of a genuine signature.

4.1.1. Brazilian SV database

The Brazilian SV database 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_{dev} contains signature samples from writers not enrolled to the system, and is used as *prior* knowledge to design the codebooks and the impostor's 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.

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 has 40 genuine samples, 10 simple forgeries and 10 skilled forgeries. In the first scenario, 20 genuine samples are available for training ($\mathcal{T}^i_{exp,20}$) and 10 for validation ($\mathcal{V}^i_{exp,10}$). In the second scenario, 4, 8 and 12 genuine samples are available for training, taken from $\mathcal{T}^i_{exp,20}$. The test set is the same for Scenarios 1 and 2, that is, each writer in DB_{exp} has 10 genuine samples ($TST^i_{true,10}$), 10 random ($TST^i_{rand,10}$) forgeries, 10 simple ($TST^i_{simp,10}$) forgeries and 10 skilled forgeries ($TST^i_{skil,10}$); where the random forgeries are genuine samples randomly selected from other writers in DB_{exp}.

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, as shown in Table 2. In the generative stage, each writer j in DB_{dev}, for $1 \leq j \leq 108$, has a set of HMMs trained using with his/her 20 genuine samples (or 4, 8 and 12 genuine samples, in Scenario 2). These HMMs compose the impostor's space, $\boldsymbol{w_2}$, from which different subspaces are selected by the Random Subspace Method. In the discriminative stage, 20 genuine samples (or 4, 8 and 12 genuine samples, in Scenario 2) are randomly chosen among all signature samples from DB_{dev} and used as random forgeries to train a pool of SVMs. In this case, the indice j is not specified in the training set $\mathcal{T}$ (as shown in second row, fourth column). A similar procedure is applied to the validation set $\mathcal{V}$, and to the GPDS database, which is described in the following section.

4.1.2. GPDS database

The GPDS database is composed of 16200 signature images digitized as 8-bit greyscale at resolution of 300 dpi. It contains 300 writers, where the first 160 are set as DB_{exp} and the remaining 140, as DB_{dev}. For each writer in both DB_{exp} and DB_{dev}, there are 24 genuine signatures and 30 skilled forgeries. In the literature, only 80 to 160 writers (out of 300) are used to develop the SV systems, which allow us to work with two datasets.

As this database has a limited number of genuine signatures per writer, it is employed only in Scenario 2 (see Table 3). For each writer j in DB_{dev}, 14 genuine signatures are available for training ($\mathcal{T}^j_{dev,14}$) and 10 for validation ($\mathcal{V}^j_{dev,10}$); while in DB_{exp}, each writer i has 14 genuine signatures available for training

Table 2: Datasets for a specific writer i, using the Brazilian SV database.

(a) Scenario 1 – design

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

(b) Scenario 2 – design

Dataset Name	Task	Genuine Samples	Random Forgery Samples
DB^i_{hmm}	HMM Training	4, 8, 12 from $\mathcal{T}^i_{exp,20}$	4, 8, 12 from $\mathcal{T}^j_{dev,20}$, for $1 \leq j \leq 108$
DB^i_{svm}	SVM Training		4, 8, 12 from $\mathcal{T}_{dev,20\times 108}$
DB^i_{grid}	SVM Grid Search		4, 8, 12 from $\mathcal{V}_{dev,10\times 108}$
DB^i_{roc}	ROC Curve		$\mathcal{V}^j_{dev,10}$, for $1 \leq j \leq 108$
DB^i_{ds}	Dynamic Selection		

(c) Scenarios 1 and 2 – verification

Dataset Name	Genuine Samples	Forgery Samples
DB^i_{tst}	$TST^i_{true,10}$	$TST^i_{rand,10} + TST^i_{simp,10} + TST^i_{skil,10}$

($\mathcal{T}^i_{dev,14}$) and 10 for test ($TST^i_{true,10}$). Moreover, 10 random forgeries ($TST^i_{rand,10}$) and 10 skilled forgeries ($TST^i_{skil,10}$) are used for test, where the random forgeries are genuine samples randomly selected from other writers in DB_{exp}. During the comparative analysis performed with systems in the literature (presented in Section 5.4), 30 skilled forgeries ($TST^i_{skil,30}$) are used for test, instead of 10.

In the following sections, the GPDS database is refered as GPDS-160, since a set of 160 writers (that is, DB_{exp}) is actually modeled by the proposed system.

4.2. Grid Segmentation

After conversion to black and white using the Otsu's binarization method [39], the signature images of the Brazilian SV database (composed of 400x1000 pixels) are divided in 62 horizontal cells, where each cell is a rectangle composed of 40x16 pixels. This grid resolution along with pixel density features have been successfully applied to this database in [6].

Without using any optimization process, a similar grid resolution is applied to the GPDS-300 database. Although this database contains images of different sizes (that vary from 51x82 pixels to 402x649 pixels), they are represented in a grid of 400x650 pixels, and segmented in 65 horizontal cells of 40x10 pixels. For real-world applications, the definition of writer-adapted grid resolutions may be unfeasible, specially when

Table 3: Datasets for a specific writer i, using the GPDS database.

(a) Scenario 2 – design

<table>
<tr><th>Dataset Name</th><th>Task</th><th>Genuine Samples</th><th>Random Forgery Samples</th></tr>
<tr><td>DB^i_{hmm}</td><td>HMM Training</td><td rowspan="5">4, 8, 12 from $\mathcal{T}^i_{exp(14)}$</td><td>4, 8, 12 from $\mathcal{T}^j_{dev,14}$, for $1 \leq j \leq 140$</td></tr>
<tr><td>DB^i_{svm}</td><td>SVM Training</td><td>4, 8, 12 from $\mathcal{T}_{dev,14\times140}$</td></tr>
<tr><td>DB^i_{grid}</td><td>SVM Grid Search</td><td>4, 8, 12 from $\mathcal{V}_{dev,14\times140}$</td></tr>
<tr><td>DB^i_{roc}</td><td>ROC Curve</td><td rowspan="2">$\mathcal{V}^j_{dev,14}$, for $1 \leq j \leq 140$</td></tr>
<tr><td>DB^i_{ds}</td><td>Dynamic Selection</td></tr>
</table>

(b) Scenario 2 – verification

<table>
<tr><th>Dataset Name</th><th>Genuine Samples</th><th>Forgery Samples</th></tr>
<tr><td rowspan="2">DB^i_{tst}</td><td rowspan="2">$TST^i_{true,10}$</td><td>$TST^i_{rand,10}$+$TST^i_{skil,10}$</td></tr>
<tr><td>$TST^i_{rand,10} + TST^i_{skil,30}$</td></tr>
</table>

dealing with a limited number of signatures samples per writer. Moreover, if we assume that the signature samples come from a same type of document, i.e., checks from a specific bank, the area used for signing is not supposed to vary.

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. Therefore, the images may have a variable number of horizontal cells, while the number of vertical cells is always 10, as shows the example of Figure 6.

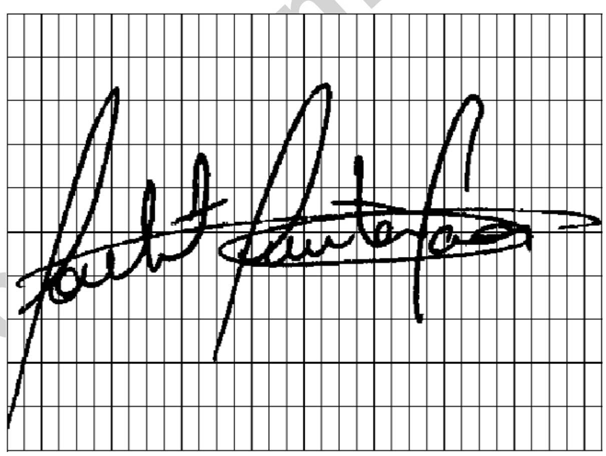

Figure 6: Example of grid segmentation scheme. By trying different resolutions, Justino [40] has shown that the grid with 10 vertical cells is the most suitable for the Brazilian SV database. This analysis was performed with DB_{dev}, that is, using signature samples from writers not enrolled to the system.

4.3. Training of the Generative Stage

Twenty-nine (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}. In Scenario 1, given a writer i and a codebook q, 20 genuine samples are taken from DB^i_{hmm} (i.e., $\mathcal{T}^i_{exp,20}$) and used to train a set of discrete *left-to-right* HMMs with different number of states. As the number of states varies from 2 to the length of the smallest sequence used for training (L_{min}), the genuine space, $\boldsymbol{w_1}$, 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's space, $\boldsymbol{w_2}$, there are thousands of HMMs taken from the writers in DB_{dev} (each writer j in DB_{dev} has a set of HMMs trained with his/her genuine samples).

Each HMM is trained by using the Baum-Welch Forward-Backward algoritm [3], and at each iteration t, a error measure $\mathcal{E}_t$ is calculated as:

$$\mathcal{E}_t = \frac{P(\mathbf{O}^i_q/\lambda^{(t)}) - P(\mathbf{O}^i_q/\lambda^{(t-1)})}{P(\mathbf{O}^i_q/\lambda^{(t)}) + P(\mathbf{O}^i_q/\lambda^{(t-1)})} \qquad (2)$$

where $P(\mathbf{O}^i_q/\lambda^{(t)})$ and $P(\mathbf{O}/\lambda^{(t-1)})$ represent the joint probabilities of the training sequences $\mathbf{O}^i_q$ have been generated by the HMM λ in the instants t and $t-1$, respectively. The goal is to reach an error of 10^{-5} or smaller [40]. Besides this stop criteria, a validation set (i.e., $\mathcal{V}^i_{exp,10}$, available only for Scenario 1) is used in order to select the optimal training point before overfitting.

In Scenario 2, the number of training sequences is dramatically reduced to 4, 8 and 12. For a given training sequence $\mathbf{O}^i_q$, a set of HMMs is trained by varying the number of states from 2 to $\frac{1}{3}$ of the sequence's size. The use of a single training sequence per HMM was previoulsy investigated in [19]. The main advantage of this strategy is that it allows to obtain a higher number of HMMs, adding more diversity to the next system stage.

4.4. Training of the Discriminative Stage

Although any discriminative two-class classifier can be used in the second stage, the SVM classifier with RBF kernel [41] was chosen because of its successful use in different pattern recognition problems.

By employing the LIBSVM toolbox [41], the parameters C and γ are found through a gridsearch technique. For each different pair {C, γ}, 10 different SVMs are trained using a variant of the cross-validation method, where the genuine samples performs the usual rotation, and the random forgery samples are changed each time. Since number of genuine samples in DB^i_{grid} can be smaller than the number of SVMs to be trained (i.e., in Scenario 2), the same genuine samples are used to train more than one SVM. On the other hand, new random forgery samples are selected, randomly and with remplacement, each time that a new SVM training is performed. Finally, the error rates provided by the 10 SVMs are averaged, and the pair {C, γ} providing the smallest error rates are used to train the final SVMs.

One hundred (100) SVMs are trained per writer, using the specialized Random Subspace Method with $R' = S' = 15$. Note that, for a same writer i, the training set, DB^i_{svm}, remains the same for all 100 SVMs.

4.5. Classifier Ensemble Selection

In this paper, the simulation results obtained with OP-UNION and OP-ELIMINATE are compared with KNORA-UNION/ELIMINATE [10], the standard combination of all classifiers, and Decision Templates (DT) [42].

With OP-UNION and OP-ELIMINATE, the search for the K-nearest neighbors is done by using the output labels provided by all 100 SVMs; while with KNORA-UNION and KNORA-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}. If the value of K is even, $K+1$ classifiers are used in order to avoid votes that result in a tie. In Scenario 1, K is set as 5; while in Scenario 2, K is set as 3, 5 and 7.

The standard combination of classifiers consists of sending the test vector $\mathbf{D}\left(\boldsymbol{O}^{\boldsymbol{i}}_{\boldsymbol{tst}}, \boldsymbol{\mathcal{M}}^{\boldsymbol{i}}\right)$ to all 100 SVMs and then fusing their corresponding output labels by majory vote. If a tie vote is obtained, the final output label is randomly chosen.

The decision templates (DTs) is a well-known dynamic selection method in the multi-classifier system (MCS) community [42]. First, each DS vector $\mathbf{D}\left(\boldsymbol{O}^{\boldsymbol{i}}_{\boldsymbol{ds(j)}}, \boldsymbol{\mathcal{M}}^{\boldsymbol{i}}\right)$, for $1 \leq j \leq M$, is sent to all SVMs, and its corresponding output labels are organized in a decision profile (DP) matrix, where each line corresponds to a different SVM and each column corresponds to a different class. Since we work with 100 two-class classifiers, each $\mathbf{DP}\left(\mathbf{D}\left(\boldsymbol{O}^{\boldsymbol{i}}_{\boldsymbol{ds(j)}}, \boldsymbol{\mathcal{M}}^{\boldsymbol{i}}\right)\right)$ is composed of 2 columns and 100 lines, where each cell contains the value 1 or 0. For instance, if the 100th SVM classifies $\mathbf{D}\left(\boldsymbol{O}^{\boldsymbol{i}}_{\boldsymbol{ds(j)}}, \boldsymbol{\mathcal{M}}^{\boldsymbol{i}}\right)$ as belonging to the genuine class C_1, the cell located in line 100 and column 1 will contain the value 1, while the cell located in line 100 and column 2 (which corresponds to the class C_2) will contain the value 0. Then, a decision template (DT) is calculated for each class C_j $(j = 1, 2)$ by averaging the DPs of the DS vectors belonging to this class. When a test vector $\mathbf{D}\left(\boldsymbol{O}^{\boldsymbol{i}}_{\boldsymbol{tst}}, \boldsymbol{\mathcal{M}}^{\boldsymbol{i}}\right)$ is presented, its decision profile matrix $\mathbf{DP}\left(\mathbf{D}\left(\boldsymbol{O}^{\boldsymbol{i}}_{\boldsymbol{tst}}, \boldsymbol{\mathcal{M}}^{\boldsymbol{i}}\right)\right)$ is calculated and compared to the decision templates $\mathbf{DT}(C_j)$. The comparison is done by using the Euclidean distance, and the higher the similarity between $\mathbf{DP}\left(\mathbf{D}\left(\boldsymbol{O}^{\boldsymbol{i}}_{\boldsymbol{ds(j)}}, \boldsymbol{\mathcal{M}}^{\boldsymbol{i}}\right)\right)$ and $\mathbf{DT}(C_j)$, the higher the support for class C_j. Finally, the most likely $\mathbf{DT}(C_j)$ is selected and the output label the most represented in this template is assigned to $\mathbf{D}\left(\boldsymbol{O}^{\boldsymbol{i}}_{\boldsymbol{ds(j)}}, \boldsymbol{\mathcal{M}}^{\boldsymbol{i}}\right)$.

The dynamic selection strategies proposed in this paper are compared as well with two reference systems proposed in our previous work, that is, (i) a traditional generative system based on HMMs [4] (refered in this paper as baseline system), and (ii) a hybrid system based on the static selection of generative-discriminative ensembles [35]. Both systems are briefly described in Section 5.1.

4.6. Performance Evaluation

In this paper, it is assumed that the overall system performance is measured by an averaged ROC curve described as follows [4, 43]. For each user i, the cumulative histogram of his/her random forgery scores (taken from DB^i_{roc}) is computed. Then, the scores providing a same value of cumulative frequency, γ, are used as thresholds to compute the operating points $\{TAR_i(\gamma), FAR_i(\gamma)\}$. Finally, the operating points associated with a same γ (and related to different users) are averaged. Note that γ can be viewed as the true negative rate (TRR = ratio of random forgeries correctly classified to the total of random forgeries) and that it may be associated with different thresholds. Fig. 7 shows an example where the thresholds associated with $\gamma = 0.3$ are different for users 1 and 2, that is $\tau_{user1}(0.3) \cong -5.6$ and $\tau_{user2}(0.3) \cong -6.4$. In other words, regarding user 1, 30% of the random forgery scores were below than -5.6, while for user 2, 30% of the random forgery scores were below than -6.4.

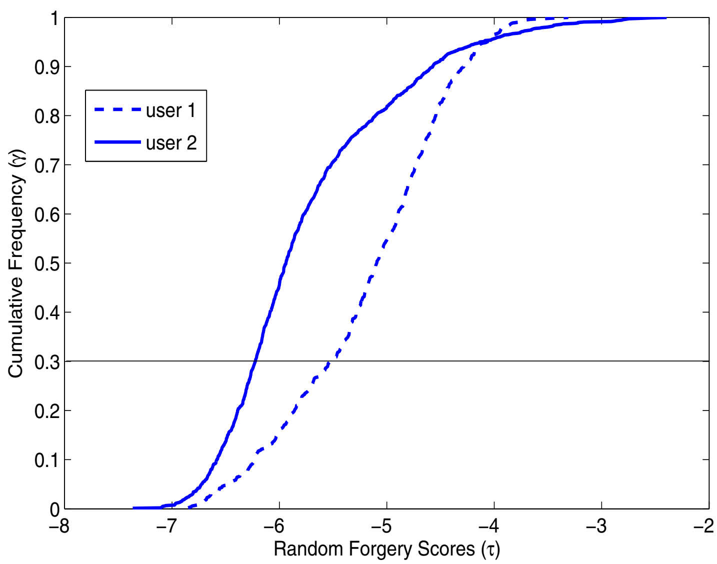

Figure 7: Cumulative histogram of random forgery scores regarding two different users.

Since different classifiers are trained through the Random Subspace Method, each classifier results in a different averaged ROC curve. To measure the system performance during verification, FRR and FAR are calculated by using the user-specific thresholds associated to a given γ of the averaged ROC curves. The average error rate (AER), also computed for a given γ, indicates the total error of the system, where FRR and FAR are averaged taking into account the *a priori* probabilities.

When the Brazilian SV database is used, the FAR is calculated with respect to three forgery types: random, simple and skilled (see DB^i_{tst} of Table 2), that is,

$$AER = (FRR + FAR_{rand} + FAR_{simp} + FAR_{skil})/4 \qquad (3)$$

While for GPDS database, FAR is calculated with respect to random and skilled forgeries (see DB^i_{tst} of Table 3), that is,

$$AER = (FRR + FAR_{rand} + FAR_{skil})/3 \qquad (4)$$

5. Simulation Results and Discussions

5.1. Reference Systems

The baseline off-line SV system [4] was designed under a traditional HMM-based approach, which consists of training a single HMM per writer. By using DB^i_{hmm} (see Table 2 (a)) and a single codebook, multiple HMMs were trained with a different number of states in order to isolate an HMM order that provides the smallest error $\mathcal{E}_t$ (see Equation (2)). This resulted in an AER ($\gamma = 1.0$) of 8.50% on test data. To reduce error rates and exploit the sub-optimal HMMs discarded by this baseline system, a multi-hypothesis system was also designed. By training a set of HMMs with a different number of states, and then selecting the most accurate HMM for each operating point of the ROC space, an AER ($\gamma = 1.0$) of 7.79% was obtained on test data.

In [35], a hybrid system based on the static selection of generative-discriminative ensembles was proposed. Given the same HMMs generated through the multi-hypothesis approach, only the most representative HMMs were selected to compose the generative stage, by using a greedy search algorithm. As performed in the bank of HMMs of Figure 4, the representative HMMs were employed as a feature extractors for the discriminative stage. To generate a pool of SVMs, a different SVM was trained using DB^i_{svm} (see Table 2 (a)), each time that the greedy algorithm added a new representative HMM to the system. Then, the ICON algorithm [44] was applied in order to incrementally construct the ensemble. Like the model selection algorithm, ICON consists in a greedy process that, during each iteration, chooses the SVM that most improves system performance on validation data when added to the ensemble. A margin-based measure called CI (from Chebishev's inequality) [45, 46] was employed to assess the performance of each ensemble. This resulted in an AER ($\gamma = 1.0$) of 5.50% on test data – which represents an improvement of 2.50% with respect to the baseline system in [4]. Figure 9 shows the corresponding $AERs$ curves.

5.2. Scenario 1 – abundant data

In this experiment, each DB^i_{ds} is composed of 10 genuine samples supplied by writer i (in DB_{exp}) *versus* 1080 random forgery samples taken from writers not enrolled to the system (see Table 2 (a)). Note that the random forgery samples are the same for all writers in DB_{exp}. Figure 8 shows the averaged ROC curves obtained with scores produced from 100 different SVMs using DB^i_{roc} , while Figure 9 presents the $AERs$ curves on test data (DB^i_{tst}), as function of operating points (γ).

Results indicate that OP-ELIMINATE and OP-UNION strategies provided the lowest $AERs$, demonstrating the advantage of using a dynamic selection approach based on Output Profiles – as opposed to KNORA, where the input feature space is used to find the K-nearest DS vectors. It is also beneficial to employ EoCs composed of a small set of base classifiers – in contrast to Decision Templates and to the standard combination of classifiers, where all base classifiers in the pool are part of the ensemble.

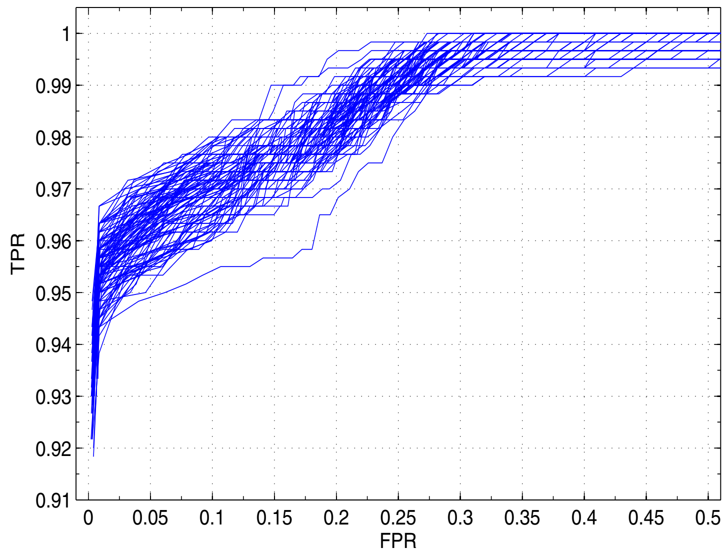

Figure 8: Averaged ROC curves obtained with scores produced from 100 different SVMs using DB^i_{roc} (from Brazilian data), under Scenario 1.s

OP-ELIMINATE and OP-UNION also achieved $AERs$ that are lower than those obtained with static selection, showing that the proposed dynamic selection strategies are more suitable for SV, where a significant level of uncertainty resides due the availability of partial knowledge during system design. Generaly, only genuine and random forgery samples are available to design a SV system. This system, in turn, must detect other forgery types during verification. Finally, the fewer performance of the baseline system is obtained because a pure generative approach was adopted for system design, where only the genuine class is modeled, and a single HMM is employed per writer. Tables 4 and 5 presents the overall results for $\gamma = 0.90$ and $\gamma = 1.0$, respectively.

Table 4: Overall error rates (%) obtained on Brazilian test data for $\gamma = 0.90$, under Scenario 1.

Method	FRR	FAR_{random}	FAR_{simple}	$FAR_{skilled}$	AER
OP-UNION	3.33	1.67	3.83	34.83	10.92
OP-ELIMINATE	**4.67**	**1.33**	**2.83**	**30.00**	**9.71**
KNORA-UNION	2.17	2.50	7.00	45.33	14.25
KNORA-ELIMINATE	2.33	2.67	6.33	44.33	13.92
Decision Templates	2.17	2.17	7.17	45.50	14.25
Combination of 100 SVMs	2.00	2.67	8.17	47.33	15.04
Static Selection [35]	2.17	8.00	5.83	35.17	12.79
Baseline [4]	0.33	12.17	20.67	78.83	28.00

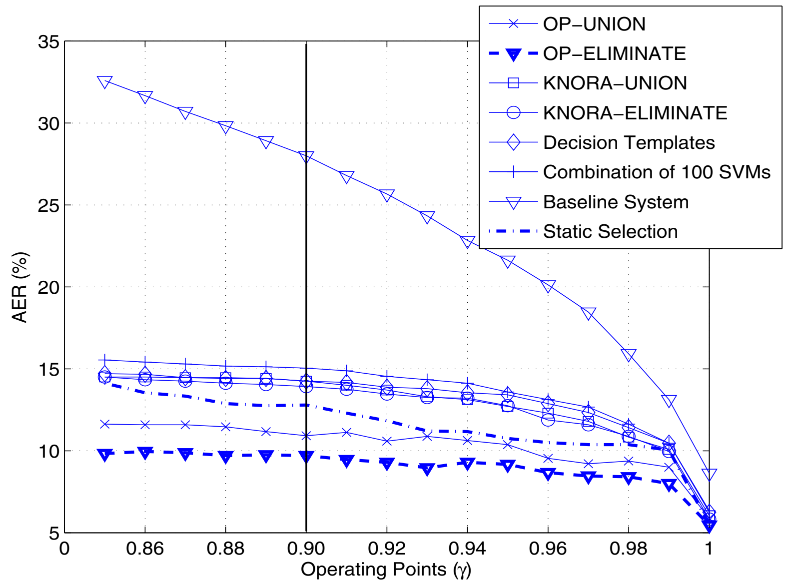

Figure 9: *AERs versus* operating points (γ) obtained on Brazilian test data with different SV systems, under Scenario 1.

Table 5: Overall error rates (%) obtained on Brazilian test data for $\gamma = 1.0$, under Scenario 1.

Method	FRR	FAR_{random}	FAR_{simple}	$FAR_{skilled}$	AER
OP-UNION	8.17	0.67	0.67	14.00	5.88
OP-ELIMINATE	**7.50**	**0.33**	**0.50**	**13.50**	**5.46**
KNORA-UNION	8.17	0.67	0.67	14.67	6.04
KNORA-ELIMINATE	7.83	0.67	0.67	14.17	5.83
Decision Templates	8.67	0.50	0.67	15.17	6.25
Combination of 100 SVMs	8.83	0.50	0.67	15.33	6.33
Static Selection [35]	13.50	0.00	0.17	8.33	5.50
Baseline [4]	12.67	0.33	1.17	19.83	8.50

5.3. Scenario 2 – sparse data

In this experiment, each DB^i_{ds} is composed of 4, 8 and 12 genuine samples supplied by writer i (in DB_{exp}) *versus* several random forgery samples taken from writers not enrolled to the system (see Tables 2 (b) and 3 (a)), where the random forgery samples are the same for all writers in DB_{exp}. Figure 10 presents a comparison between the baseline system [35] and OP-ELIMINATE, when trained with 4, 8 and 12 genuine samples from the Brazilian SV database. The results obtained by these systems when trained with 20 genuine samples, previously presented in Scenario 1, are also shown in this graph. As expected, system performance improves as new genuine samples are used for training. Such an improvement is less pronounced with the baseline system. As with Scenario 1, the proposed system reached the smallest $AERs$.

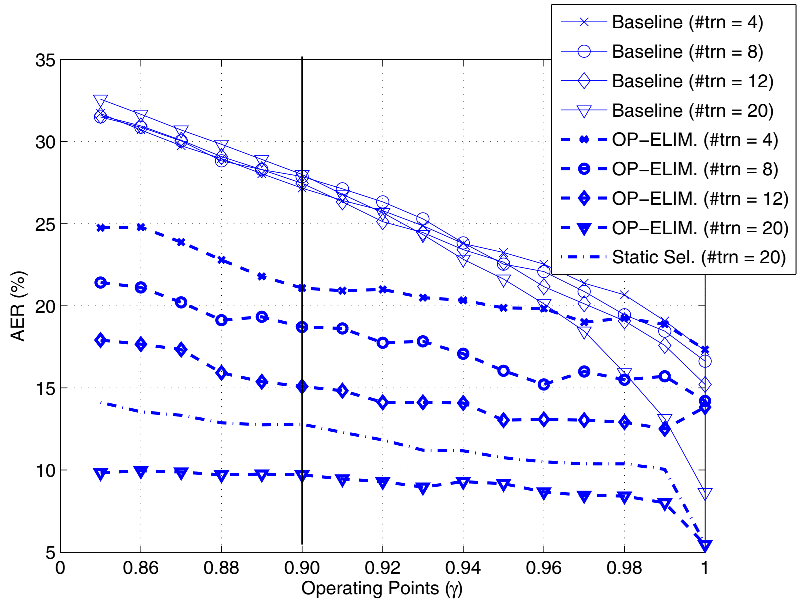

Figure 10: *AERs versus* operating points (γ) obtained on Brazilian test data with the baseline system and OP-ELIMINATE, under Scenario 2.

Figures 11 and 12 show the *AERs* obtained with the proposed systems for both Brazilian and GPDS-160 databases, and Table 6 presents the overall error rates obtained for $\gamma = 0.90$; where $\#trn$ indicates the number of genuine signatures in the training set (i.e., DB^i_{hmm} and DB^i_{svm}), and $\#ds$ indicates the number of genuine signatures in DB^i_{ds}. Note that Table 6 (d) is related to the Scenario 1, where the training and dynamic selection sets differ.

OP-UNION seems to be more suitable than OP-ELIMINATE when the base classifiers are trained with a very small number of signatures (for instance, 4 genuine signatures *vs.* 4 random forgeries). In fact, since classifiers trained with few signature samples are not very accurate, it is desirable to select a higher number of classifiers to form an EoC.

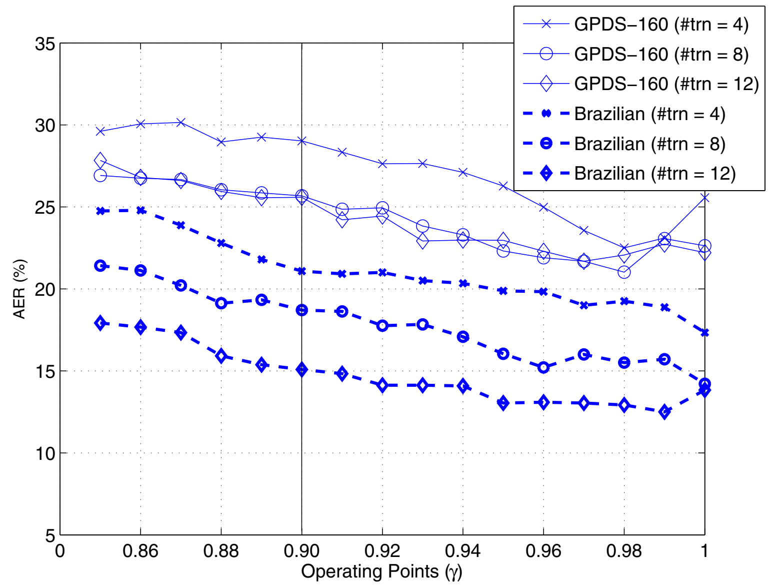

Figure 11: *AERs versus* operating points (γ) obtained on Brazilian and GPDS-160 test data with OP-ELIMINATE strategy, under Scenario 2.

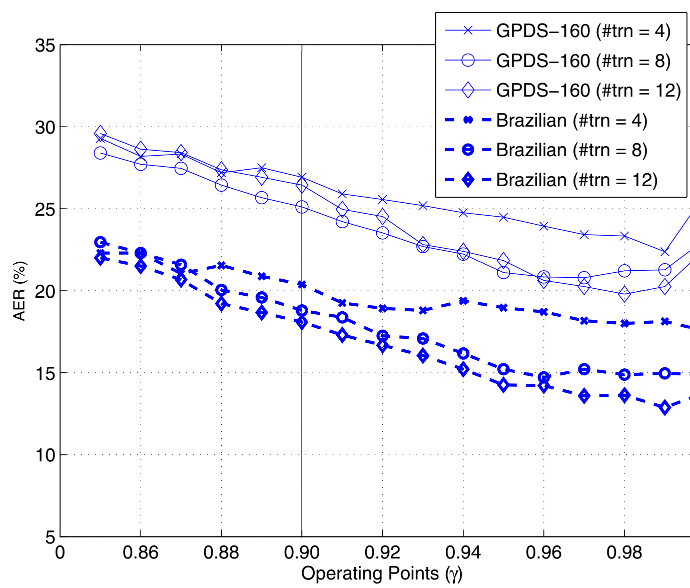

Figure 12: *AERs versus* operating points (γ) obtained on Brazilian and GPDS-160 test data with OP-UNION strategy, under Scenario 2.

Table 6: Overall error rates (%) obtained on Brazilian and GPDS-160 test data for $\gamma = 0.90$, under Scenario 2.

(a) $\#trn = 4, \#ds = 4, K = 3$

Database	Method	FRR	FAR_{random}	FAR_{simple}	$FAR_{skilled}$	AER
Brazilian	Baseline	16.17	9.67	15.00	67.83	27.17
	OP-UNION	**26.17**	**10.83**	**9.33**	**35.17**	**20.38**
	OP-ELIMINATE	27.33	11.67	10.50	34.83	21.08
GPDS-160	**OP-UNION**	**19.44**	**12.62**	N/A	**48.69**	**26.92**
	OP-ELIMINATE	20.69	16.56	N/A	50.50	29.25

(b) $\#trn = 8, \#ds = 8, K = 5$

Database	Method	FRR	FAR_{random}	FAR_{simple}	$FAR_{skilled}$	AER
Brazilian	Baseline	8.00	11.00	19.17	73.33	27.87
	OP-UNION	11.33	11.83	9.67	42.33	18.79
	OP-ELIMINATE	**16.00**	**11.33**	**8.67**	**38.83**	**18.71**
GPDS-160	**OP-UNION**	**14.88**	**11.44**	N/A	**49.00**	**25.10**
	OP-ELIMINATE	16.62	11.62	N/A	48.75	25.67

(c) $\#trn = 12, \#ds = 12, K = 7$

Database	Method	FRR	FAR_{random}	FAR_{simple}	$FAR_{skilled}$	AER
Brazilian	Baseline	5.17	11.50	19.33	73.83	27.46
	OP-UNION	7.83	10.67	10.50	43.33	18.08
	OP-ELIMINATE	**13.50**	**5.67**	**6.83**	**34.33**	**15.08**
GPDS-160	OP-UNION	13.75	11.62	N/A	53.94	26.44
	OP-ELIMINATE	**19.19**	**9.81**	N/A	**47.25**	**25.42**

(d) $\#trn = 20, \#ds = 10, K = 5$ (from Scenario 1)

Database	Method	FRR	FAR_{random}	FAR_{simple}	$FAR_{skilled}$	AER
Brazilian	Baseline	0.33	12.17	20.67	78.83	28.00
	Static Selection	2.17	8.00	5.83	35.17	12.79
	OP-UNION	3.33	1.67	3.83	34.83	10.92
	OP-ELIMINATE	**4.67**	**1.33**	**2.83**	**30.00**	**9.71**

The proposed system achieved higher error rates with the GPDS-160 database because it contains different image sizes, which vary (vertically and horizontally) even for a same writer. In this work, no normalization technique was employed. As explained in Section 4.2, a fixed-sized grid – suitable for the Brazilian SV database – was applied to all writers in the GPDS-160 database. With the Brazilian SV database, the region used for signing does not vary, since it simulates the case where the signature samples come from a same type of document, i.e., checks from a specific bank.

The final experiment investigates the adaptive capabilities of the proposed system when new genuine signatures are integrated incrementally. A limited number of genuine signatures are used to design both generative and discriminative stages. Then, the goal is gradually improve system performance by adding new genuine signatures to DB^i_{ds}.

First, DB^i_{ds} is composed of 4 genuine signatures *versus* 1080 random forgeries from DB_{dev}, as performed in the previous experiment. Then, DB^i_{ds} is updated twice, by adding 4 new genuine signatures each time. Figures 13 and 14 show the AER curves (in bold) obtained with OP-ELIMINATE using the Brazilian and GPDS-160 databases, respectively. The AER curves related to the systems trained with 8 genuine samples and 12 genuine samples, from the previous experiment, are also presented in these figures.

The addition newly-obtained genuine samples in DB^i_{ds} improved system performance in almost all operating points. With the Brazilian SV database (see Figure 13), the performance of the system using 4

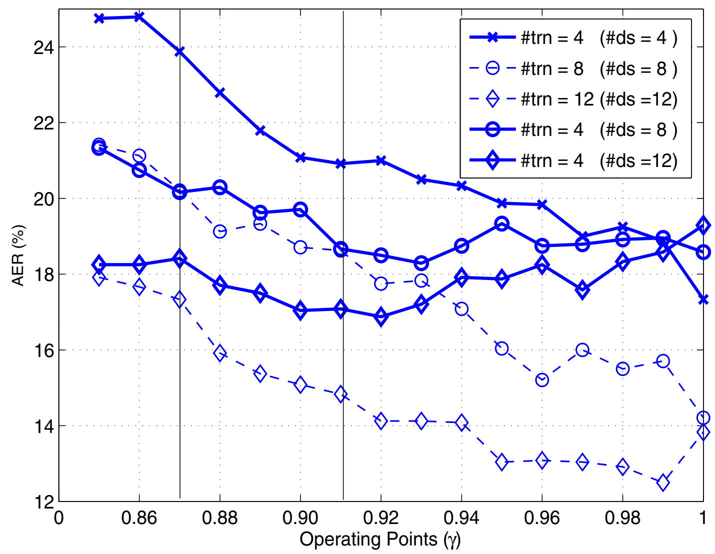

Figure 13: *AERs versus* operating points (γ) obtained on Brazilian test data with incremental updating and OP-ELIMINATE strategy, under Scenario 2.

genuine samples for training and 8 for dynamic selection is comparable to that of using 8 genuine samples for both training and dynamic selection in some operating points, such as $\gamma = 0.91$ and $\gamma = 0.87$. With the GPDS-160 database (see Figure 14), the performance of the system using 4 signatures for training and 12 for DS is comparable to or better than that of using 12 genuine samples for both training and dynamic selection, when $\gamma \leq 0.92$.

The main advantage of adapting DB^i_{ds} incrementally is that the actual classifiers need not be retrained. Moreover, more genuine signatures are exploited by OP-UNION and OP-ELIMINATE during the dynamic selection of classifiers. Although more complexes, the systems trained with 8 and 12 genuine samples provide, in general, lower error rates compared to the system trained with 4 genuine samples and using 8 and 12 samples for dynamic selection, respectively. Therefore, incremental updating of DB^i_{ds} represents a viable measure to improve system performance, and may be used in conjuction with incremental learning (IL) of classifiers.

5.4. Comparisons with systems in the literature

Table 7 presents the error rates provided by systems designed with the Brazilian SV database. While [23, 47] propose discriminative systems based on dissimilarity representation, [48] proposes a traditional generative system based on HMMs [48]. Finally, as described in Section 5.1, a multi-hypothesis system based on HMMs is proposed in [4]. Since these systems require a considerable number of signatures for training, they are compared with the best system obtained in Scenario 1.

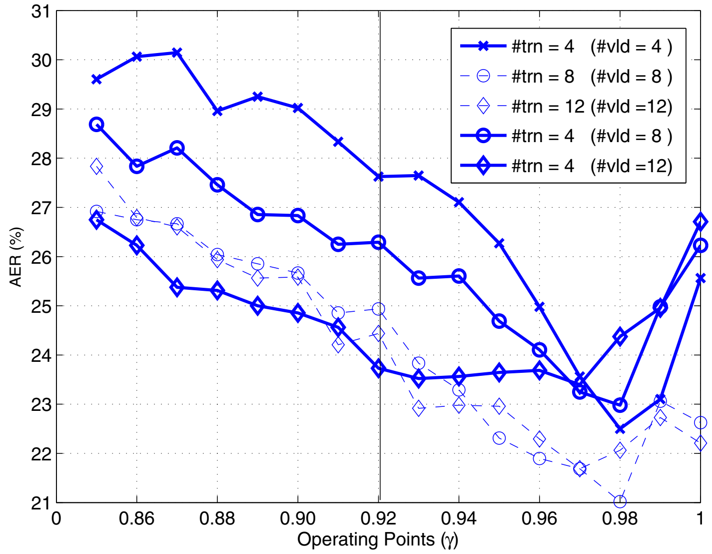

Figure 14: $AERs$ *versus* operating points (γ) obtained on GPDS test data with incremental updating of DB^i_{ds} and OP-ELIMINATE strategy, under Scenario 2.

Table 7: Overall error rates (%) provided by systems designed with the Brazilian SV database.

Reference	FRR	FAR_{random}	FAR_{simple}	$FAR_{skilled}$	AER
Batista et al. [4]	9.83	0.00	1.00	20.33	7.79
Bertolini et al. [23]	11.32	4.32	3.00	6.48	6.28
Justino et al. [48]	2.17	1.23	3.17	36.57	7.87
Santos et al. [47]	10.33	4.41	1.67	15.67	8.02
OP-ELIMINATE ($\gamma = 1.0$)	**7.50**	**0.33**	**0.50**	**13.50**	**5.46**

Comparisons with other systems is difficult because of the use of different features, databases and experimentation protocols. In our research, only genuine signatures and random forgeries are considered during training, validation and thresholding, since other forgery types are not available during the design of a real-world SV system. However, some authors have used skilled forgeries to select optimal decision thresholds. In order to compare with systems that use the GPDS database, the equal error rate (EER) – obtained when the threshold is set to have the FRR approximately equal to the FAR – is employed. Two operating points are chosen from the test scores: one regarding genuine signatures *vs.* random forgeries, and a second regarding genuine signatures *vs.* skilled forgeries (where 30 skilled forgeries are employed, instead of 10). Table 8 presents the $EERs$ provided by the proposed system and other systems designed with different subsets of the GPDS database. Results presented on multiple rows correspond to the use of different feature extraction/selection techniques or classifiers. In the work of Ferrer et al. [5], for instance, three different classifiers – HMMs, SVMs and Euclidean Distance-based classifiers – were trained using 12 signatures.

It is worth noting that both feature extraction and classification techniques presented in this paper were proposed taking into account the Brazilian SV database; which, posteriorly, were applied to the GPDS database without any optimization process. The systems presented in Table 8, however, have been optimized to the GPDS database, which explains the slightly lower error rates. Moreover, these systems have been designed and tested using a same set of writers, DB_{exp}. Our systems are based on two independent datasets: DB_{dev} is employed to generate codebooks and to train the impostor's class, and DB_{exp} is employed to train the genuine class and to test the system. It is therefore considered that the results obtained in this paper are comparable to those reported in the literature.

Table 8: *EERs* (%) provided by the proposed system and by other systems in the literature, using the GPDS database. LBP stands for local binary pattern, GLCM, for grey level co-occurrence matrix, and MDF, for modified direction feature.

(a) genuine signatures *vs.* random forgeries

Reference	Database	Technique	*#trn*	*FRR* (%)	*FAR* (%)
Ferrer et al. [5]	GPDS-160	HMM classifiers	4	4.30	3.80
			8	2.50	2.40
Ferrer et al.[5]	GPDS-160	HMM classifiers	12	2.20	3.30
		SVM classifiers		3.23	2.65
		Euclidean Distance		5.56	5.13
Vargas et al.[38]	GPDS-100	LBP + GLCM features	5	3.75	3.75
		LBP features		4.59	4.59
		GLCM features		6.40	6.40
Vargas et al.[38]	GPDS-100	LBP + GLCM features	10	1.76	1.76
		LBP features		2.41	2.41
		GLCM features		4.31	4.31
OP-UNION	GPDS-160	HMM + SVM classifiers	4	7.75	6.56
			8	5.38	5.44
			12	4.50	5.19

(b) genuine signatures *vs.* skilled forgeries

Reference	Database	Technique	*#trn*	*FRR* (%)	*FAR* (%)
Ferrer et al.[5]	GPDS-160	HMM classifiers	4	17.30	14.90
			8	13.40	14.90
Ferrer et al.[5]	GPDS-160	HMM classifiers	12	14.10	12.60
		SVM classifiers		15.41	13.12
		Euclidean Distance		16.21	15.66
Nguyen et al. [49]	GPDS-160	MDF features	12	17.25	17.25
		Gradient features		16.54	13.51
Vargas et al.[38]	GPDS-100	LBP + GLCM features	5	12.06	12.06
		LBP features		13.38	13.38
		GLCM features		17.12	17.12
Vargas et al. [38]	GPDS-100	LBP + GLCM features	10	9.02	9.02
		LBP features		10.53	10.53
		GLCM features		12.18	12.18
OP-UNION	GPDS-160	HMM + SVM classifiers	4	20.75	20.31
			8	16.69	17.38
			12	16.81	16.88

5.5. System Complexity

In Scenario 1, $Q(L_{min} - 1)$ HMMs are trained per writer, where Q is the number of codebooks (i.e., 29) and L_{min} is the size of the smallest training sequence. On the other hand, $\alpha Q(\frac{1}{3}L - 1)$ HMMs are trained per writer in Scenario 2, where α is the number of genuine signatures used for training and L is the size of the training sequence being modeled. This indicates that this scenario produces about $\alpha/3$ times more

HMMs than the previous one. Nevertheless, the time complexity to train an individual HMM is lower in Scenario 2, since HMMs are trained with a single observation sequence, and with a smaller number of states. Recall that the number of HMM states varies from 2 to L_{min} in Scenario 1 and from 2 to $\frac{1}{3}L$ in Scenario 2. Regarding the discriminative stage, each SVM has a fixed input feature dimension of 15+15 (i.e., $R' + S'$).

During the experiments, the number of HMM states varied from 2 to 33 in Scenario 1 and from 2 to 12 in Scenario 2, on average. By considering only the genuine space, $\boldsymbol{w_1}$, 29x(33-1) HMMs were trained per writer in Scenario 1, and 4x29x(12-1) HMMs were trained per writer in Scenario 2, when $\alpha = 4$. Table 9 presents the average number of HMMs, states, SVM inputs and support vectors employed in each scenario. Despite the overproduction of base classifiers in both generative and discriminative stages, each individual base classifier holds a very low complexity.

Table 9: Average number of HMMs, states, SVM inputs, and support vectors (SVs) in each scenario.

Scenario	α	HMMs per writer ($\boldsymbol{w_1}$)	HMM states	SVM inputs ($R' + S'$)	SVs per SVM
1	20	928	2 to 33	15+15	25
2	4	1276	2 to 12		7
	8	2552			11
	12	3828			16

6. Conclusions

In this paper, the challenge of designing off-line SV systems from a limited amount of genuine signature samples is addressed through dynamic selection of hybrid generative-discriminative ensembles. In the generative stage, multiple discrete *left-to-right* HMMs are trained using a different number of states and codebook sizes, and employed as feature extractors for the discriminative stage. In the discriminative stage, HMM likelihoods are measured for each training signature, and assembled into feature vectors that are used to train a diversified pool of two-class classifiers through a specialized Random Subspace Method. During verification, a dynamic selection strategy selects the most accurate EoCs to classify a given input signature. Experiments performed with two real-world signature databases (comprised of genuine samples, and random, simple and skilled forgeries) indicate that the proposed dynamic selection strategy can significantly reduce the overall error rates, with respect to other EoCs formed using well-known dynamic and static selection strategies. Moreover, the performance of the hybrid generative-discriminative system is greater than or comparable to that of relevant systems found in the literature.

The system proposed in this paper combines of the advantages of multiple generative and discriminative classifiers to achieve a very high classification rate in off-line SV. The use of different codebooks and HMM states allows the system to learn each signature at different levels of perception. The codebooks – as well as the impostor class – are obtained from signatures of an independent database, ensuring that the SV system can be designed with a single user.

Another important contribution is the proposal of two new dynamic selection strategie (OP-ELIMINATE and OP-UNION), based on KNORA [10] and on Output Profiles [11], which were shown to be more suitable for off-line SV than other well-known dynamic and static selection strategies. The decision of using OP-UNION or OP-ELIMINATE may be based on the number of genuine samples employed to train the classifiers. During experiments, it was observed that OP-UNION provides better results than OP-ELIMINATE when the SVM classifiers are trained with a small number of signatures (for instance, 4 genuine signatures *vs.* 4 random forgeries). Since classifiers trained with a limited number of signature samples are less accurate, more classifiers are needed to form a robust EoC.

Finally, by choosing among different γ values from the averaged ROC curve, the system can be adjusted according to the risk linked to an input sample. In banking applications, for instance, the decision to use a specific operating point may be associated with the amount of the check. As an example, if a user rarely signs high value checks, signing for large amounts would require operating points related to low $FARs$, as would be provided by a γ value close to 1. Lower amounts would translate to operating points related to low $FRRs$, since the bank and user would not feel comfortable with frequent false rejections.

A challenging issue in biometrics is to take into account the aging of reference data in long-lived systems [50]. In this respect, the proposed SV system can be adapted such that new genuine signature samples may be integrated incrementally. As new genuine signature samples become available, the system performance may be improved overtime, without the need of retraining the actual classifiers.

7. Acknowledgments

This research has been supported by the Fonds Québécois de la Recherche sur la Nature et les Technologies and by the Natural Sciences and Engineering Research Council of Canada.

References

[1] L. Batista, D. Rivard, R. Sabourin, E. Granger, P. Maupin, State of the Art in Off-line Signature Verification, in: B. Verma, M. Blumenstein (Eds.), Pattern Recognition Technologies and Applications: Recent Advances, IGI Global, 1st edn., 2007.

[2] D. Impedovo, G. Pirlo, Automatic Signature Verification: The State of the Art 38 (5) (2008) 609–635.

[3] L. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, IEEE 77 (2) (1989) 257–286.

[4] L. Batista, E. Granger, 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 (2010) 33–47, ISSN 1433-2833.

[5] M. Ferrer, J. Alonso, C. Travieso, Offline Geometric Parameters for Automatic Signature Verification using Fixed-Point Arithmetic, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (6) (2005) 993–997.

[6] E. Justino, A. El-Yacoubi, F. Bortolozzi, R. Sabourin, An Off-Line Signature Verification System Using HMM and Graphometric Features, in: International Workshop on Document Analysis Systems, 211–222, 2000.

[7] G. Rigoll, A. Kosmala, A Systematic Comparison Between On-line and Off-line Methods for Signature Verification with Hidden Markov Models, in: International Conference on Pattern Recognition, vol. 2, 1755–1757, 1998.

[8] C. Drummond, Discriminative vs. Generative Classifiers for Cost Sensitive Learning, in: Canadian Conference on Artificial Intelligence. Lecture Notes in Artificial Intelligence, 479–490, 2006.

[9] A. El-Yacoubi, E. Justino, R. Sabourin, F. Bortolozzi, Off-line Signature Verification using HMMs and Cross-Validation, in: IEEE Workshop on Neural Networks for Signal Processing, 859–868, 2000.

[10] A. Ko, R. Sabourin, A. Britto, From dynamic classifier selection to dynamic ensemble selection, Pattern Recognition 41 (5) (2008) 1718–1731.

[11] P. Cavalin, R. Sabourin, C. Suen, Dynamic Selection of Ensembles of Classifiers Using Contextual Information, in: Ninth International Workshop on Multiple Clasifier System, 145–154, 2010.

[12] J. Vargas, M. Ferrer, C. Travieso, J. Alonso, Off-line Handwritten Signature GPDS-960 Corpus, in: International Conference on Document Analysis and Recognition, 764–768, 2007.

[13] C. Drummond, Discriminative vs. Generative Classifiers for Cost Sensitive Learning, in: Canadian Conference on Artificial Intelligence, 479–490, 2006.

[14] A. Ng, M. Jordan, On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes., in: Advances in Neural Information Processing Systems, 841–848, 2001.

[15] V. Vapnik (Ed.), The Nature of Statistical Learning Theory, Springer-Verlag, New York, 2nd edn., 1999.

[16] K. Abou-Moustafa, M. Cheriet, C. Suen, Classification of Time-Series Data Using a Generative/Discriminative Hybrid, in: Ninth International Workshop on Frontiers in Handwriting Recognition, IEEE Computer Society, 51–56, 2004.

[17] Y. Rubinstein, T. Hastie, Discriminative vs Informative Learning, in: Third International Conference on Knowledge Discovery and Data Mining, 49–53, 1997.

[18] R. Raina, Y. Shen, A. Y. Ng, A. McCallum, Classification with Hybrid Generative/Discriminative Models, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems 16, MIT Press, Cambridge, MA, 2004.

[19] M. Bicego, V. Murino, M. Figueiredo, Similarity-Based Clustering of Sequences Using Hidden Markov Models, Pattern Recognition 37 (12) (2004) 2281–2291.

[20] E. Pekalska, R. Duin, Classifiers for Dissimilarity-based Pattern Recognition, in: International Conference on Pattern Recognition, vol. 2, 12–16, 2000.

[21] E. Pekalska, R. Duin, The Dissimilarity Representation for Pattern Recognition: Foundations And Applications (Machine Perception and Artificial Intelligence), World Scientific Publishing Co., Inc., 2005.

[22] R. Bajaj, S. Chaudhury, Signature Verification using Multiple Neural Classifiers, Pattern Recognition 30 (1997) 1–7.

[23] D. Bertolini, L. Oliveira, E. Justino, R. Sabourin, Reducing Forgeries in Writer-Independent Off-Line Signature Verification through Ensemble of Classifiers, Pattern Recognition 43 (2010) 387–396.

[24] R. Sabourin, G. Genest, An extended-shadow-code based approach for off-line signature verification. II. Evaluation of several multi-classifier combination strategies, International Conference on Document Analysis and Recognition 1 (1995) 197.

[25] R. Sabourin, G. Genest, F. Prêteux, Off-Line Signature Verification by Local Granulometric Size Distributions, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (9) (1997) 976–988.

[26] D. Tax, One-class Classification, Ph.D. thesis, TU Delft, 2001.

[27] K. Tumer, J. Ghosh, Analysis of decision boundaries in linearly combined neural classifiers, Pattern Recognition 29 (2) (1996) 341 – 348.

[28] L. Breiman, Bagging Predictors, Machine Learning 2 (1996) 123–140.

[29] Y. Freund, Boosting a weak learning algorithm by majority, in: third annual workshop on Computational learning theory, Morgan Kaufmann Publishers Inc., 202–216, 1990.

[30] T. Ho, The Random Subspace Method for Constructing Decision Forests, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (8) (1998) 832–844.

[31] C. Burges, A tutorial on support vector machines for pattern recognition, Data mining and knowledge discovery 2 (2) (1998) 121–167.

[32] E. Alpaydin, Introduction to Machine Learning (Adaptive Computation and Machine Learning), The MIT Press, 2004.

[33] W. Khreich, E. Granger, A. Miri, R. Sabourin, On the memory complexity of the forward-backward algorithm, Pattern Recognition Letters 31 (2) (2010) 91–99.

[34] H. Cao, T. Naito, Y. Ninomiya, Approximate RBF Kernel SVM and Its Applications in Pedestrian Classification, in: The 1st International Workshop on Machine Learning for Vision-based Motion Analysis, 2008.

[35] L. Batista, E. Granger, R. Sabourin, A Multi-Classifier System for Off-Line Signature Verification Based on Dissimilarity Representation, in: Ninth International Workshop on Multiple Clasifier System, 264–273, 2010.

[36] L. Martinez, C. Travieso, J. Alonso, M. Ferrer, Parameterization of a forgery handwritten signature verification system using SVM, in: International Carnahan Conference on Security Technology, 193–196, 2004.

[37] G. Pirl, D. Impedovo, E. Stasolla, C. Trullo, Learning Local Correspondences for Static Signature Verification, in: Proceedings of the XIth International Conference of the Italian Association for Artificial Intelligence Reggio Emilia on Emergent Perspectives in Artificial Intelligence, Springer-Verlag, 385–394, 2009.

[38] J. Vargas, M. Ferrer, C. Travieso, J. Alonso, Off-line signature verification based on grey level information using texture features, Pattern Recognition (2011) 375–385.

[39] R. Gonzalez, R. Woods (Eds.), Digital Image Processing, Prentice Hall, 2nd edn., 2002.

[40] E. Justino, O Grafismo e os Modelos Escondidos de Markov na Verificacao Automatica de Assinaturas, Ph.D. thesis, PUC-PR, Brasil, 2001.

[41] C. Chang, C. Lin, LIBSVM: a library for support vector machines, in: http://www.csie.ntu.edu.tw/~ cjlin/libsvm, 2001.

[42] L. Kuncheva, J. Bezdek, R. Duin, Decision templates for multiple classifier fusion: an experimental comparison, Pattern Recognition 34 (2001) 299–314.

[43] A. Jain, A. Ross, Learning User-specific Parameters in a Multibiometric System, in: International Conference on Image Processing, 57–60, 2002.

[44] A. Ulas, M. Semerci, O. Yildiz, E. Alpaydin, Incremental construction of classifier and discriminant ensembles, Information Sciences 179 (9) (2009) 1298–1318.

[45] L. Breiman, Random Forests, Machine Learning 45 (2001) 5–32.

[46] M. Kapp, R. S. P., Maupin, An Empirical Study on Diversity Measures and Margin Theory for Ensembles of Classifiers,

in: 10th International Conference on Information Fusion, 1–8, 2007.

[47] C. Santos, E. Justino, F. Bortolozzi, 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, 498–502, 2004.

[48] E. Justino, F. Bortolozzi, R. Sabourin, Off-Line Signature Verification Using HMM for Random, Simple and Skilled Forgeries, in: International Conference on Document Analysis and Recognition, 105–110, 2001.

[49] V. Nguyen, Y. Kawazoe, T. Wakabayashi, U. Pal, M. Blumenstein, Performance Analysis of the Gradient Feature and the Modified Direction Feature for Off-line Signature Verification, in: International Conference on Frontiers in Handwriting Recognition, 303–307, 2010.

[50] J. Pato, L. Millett, Biometric Recognition: Challenges and Opportunities, National Academies Press, 2010.

Research Highlights

>We designed a generative-discriminative system for off-line SV with few samples. >Multiple HMMs are used as feature extractors. >HMM likelihoods are used to train a diversified pool of 2-class classifiers. >The classifier selection process is performed dynamically. >The proposed dynamic selection method is suitable for incremental learning.

***Author Biography**

Luana Bezerra Batista obtained a Ph.D. in Engineering from the École de Technologie Supérieure, Université du Québec in Montréal, in 2011. In 2005, she was research assistent in a collaborative project between HP-Brazil and Federal University of Campina Grande, in the area of facial expression recognition for digital cameras. In 2004, she worked for a company in Brazil developing intelligent solutions for fault detection and diagnosis in electrical systems, besides doing data mining for financial applications. She received a BSc in 2002 from the Federal University of Paraíba and, in 2004, and a MSc from the Federal University of Campina Grande, both degrees in Computer Science

Eric Granger obtained a Ph.D. in Electrical Engineering from the École Polytechnique de Montréal in 2001, and from 1999 to 2001, he was a Defence Scientist at Defence R&D Canada in Ottawa. Until then, his work was focused primarily on neural network signal processing for fast classification of radar signals in Electronic Surveillance (ES) systems. From 2001 to 2003, he worked in R&D with Mitel Networks Inc. on algorithms and dedicated electronic circuits (ASIC/SoC) to implement cryptographic functions in Internet Protocol (IP)-based communication platforms. In 2004, Dr. Eric Granger joined the ÉTS, where he has been developing applied research activities in the areas of machine learning, patterns recognition, signal processing and microelectronics. He presently holds the rank of Assistant Professor in the département de génie de la production automatisée (GPA). Since joining ÉTS, he has been a member of the Laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA), and his main research interests are adaptive classification systems, incremental learning, ambiguity and novelty detection, neural and statistical classifiers, and multi-classifier systems, with applications in military surveillance (recognition of radar signals), biometric authentification (from signatures and faces), and intrusion detection in computer and network security.

Dr. R. Sabourin joined in 1977 the physics department of the Montreal University where he was responsible for the design, experimentation and development of scientific instrumentation for the Mont Mégantic Astronomical Observatory. His main contribution was the design and the implementation of a microprocessor-based fine tracking system combined with a low-light level CCD detector. In

1983, he joined the staff of the École de Technologie Supérieure, Université du Québec, in Montréal where he co-founded the Dept. of Automated Manufacturing Engineering where he is currently Full Professor and teaches Pattern Recognition, Evolutionary Algorithms, Neural Networks and Fuzzy Systems. In 1992, he joined also the Computer Science Department of the Pontifícia Universidade Católica do Paraná (Curitiba, Brazil) where he was co-responsible for the implementation in 1995 of a master program and in 1998 a PhD program in applied computer science. Since 1996, he is a senior member of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI, Concordia University). Dr Sabourin is the author (and co-author) of more than 260 scientific publications including journals and conference proceeding. He was co-chair of the program committee of CIFED'98 (Conférence Internationale Francophone sur l'Écrit et le Document, Québec, Canada) and IWFHR'04 (9th International Workshop on Frontiers in Handwriting Recognition, Tokyo, Japan). He was nominated as Conference co-chair of ICDAR'07 (9th International Conference on Document Analysis and Recognition) that has been held in Curitiba, Brazil in 2007. His research interests are in the areas of handwriting recognition, signature verification, intelligent watermarking systems and bio-cryptography.