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, Montréal, QC, Canada H3C 1K3

ARTICLE INFO

Article history:
Received 22 December 2010
Received in revised form
26 September 2011
Accepted 21 October 2011
Available online 29 October 2011

Keywords:
Off-line signature verification
Ensemble of classifiers
Dynamic selection
Hybrid generative–discriminative systems
Hidden Markov Models
Incremental learning

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-orphels (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 (composed 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.

© 2011 Elsevier Ltd. All rights reserved.

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 sequential nature and variable size of the signature data [4–7]. 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 behavioral biometric traits that are known to incorporate a considerable amount of intra-class variability. Fig. 1 presents the superimposition of several signature skeleton 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 modelling the variabilities observed among signature samples of a 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 and 5 signature samples at the time of his/her subscription). A related problem regards the generation of codebooks needed to design discrete HMMs. Typically, the data used to generate codebooks 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

---

* Corresponding author. Tel.: +1 514 396 8800; fax: +1 514 396 8595. E-mail addresses: batista@livia.etsmtl.ca, luana.bezerra@gmail.com (L. Batista), eric.granger@etsmtl.ca (E. Granger); robert.sabourin@etsmtl.ca (R. Sabourin).
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-orcharies (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] (composed of genuine samples, and random, simple and skilled forgeries) and the GPDS database [12] (composed 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-known 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 classifiers differ 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 \( P(Y = y_j|X = x_k) \), and then by assigning \( x_k \) 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 to 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 architectures 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 approaches 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 [22–25]. 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 \( C \) classifiers is averaged, the variance contribution in the bias–variance decomposition decreases by \( 1/C \), 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 sample subsets to train different classifiers, the Random Subspace Method uses 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 \( T^i = \mathcal{F}_{\text{trn},i} \), for \( 1 \leq i \leq N \), be the training set used to design a SV system for writer \( i \). The set \( T^i \) 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 \( \mathcal{F}_{\text{trn},i} \) in the training set \( T^i \), a set of features is generated (see Fig. 2). First, \( \mathcal{F}_{\text{trn},i} \) is described by means of pixel densities, which are extracted through a grid composed of rectangular cells. Each column of cells \( j \) is converted into a low-level feature vector \( \mathbf{f}_j = [f_{ij1}, f_{ij2}, \ldots] \), where each vector component \( f_{ij} \in [0,1] \). These components correspond to the number of black pixels in a cell divided by the total number of pixels of this cell. The signature \( \mathcal{F}_{\text{trn},i} \) is therefore represented by a set of low-level feature vectors \( \mathcal{F}_{\text{trn},i} = \{ \mathbf{f}_j \} \), for \( 1 \leq j \leq \text{col} \), where \( \text{col} \) is the number of columns in the grid.

Then, \( \mathcal{F}_{\text{trn},i} \) is quantized into a sequence of discrete observations \( \mathbf{O}_j = [o_{ij}] \), for \( 1 \leq j \leq \text{col} \). Each observation \( o_{ij} \) is a symbol provided by the codebook \( q \) (generated using the K-means algorithm). Since \( Q \) different codebooks are employed per writer, each training signature \( \mathcal{F}_{\text{trn},i} \) yields a set of observation sequences \( \mathbf{O}_{\text{trn},i} = \{ \mathbf{O}_j \} \), for \( 1 \leq q \leq Q \). The set of observation sequences, \( \mathbf{O}_{\text{trn},i} \), is then input to the bank of left-to-right HMMs \( \mathbf{M} = \{ \lambda_q \} \), for \( 1 \leq b \leq B \), from which a high-level feature vector \( \mathbf{D}(\mathbf{O}_{\text{trn},i}^{\mathbf{M}}) = \{ \mathbf{P}_1, \ldots, \mathbf{P}_B \} \) is extracted. Each component \( \mathbf{P}_b \) is a likelihood computed between an observation sequence \( \mathbf{O}_j^{\mathbf{M}} \) and a HMM \( \lambda_q \), where \( \lambda_q \) 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 \( \mathbf{O}_{\text{trn},i} \), for \( 1 \leq q \leq N \), used to obtain the HMM likelihood vectors are also used to train the HMMs in \( \mathbf{M} \). Apart 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 \( r \), \( 1 \leq r \leq R \), a smaller subset of likelihoods is randomly selected, with replacement, from \( \mathbf{D}(\mathbf{O}_{\text{trn},i}^{\mathbf{M}}) \), for \( 1 \leq l \leq N \), and used to train a different classifier. During verification, a given input signature \( \mathcal{F}_{\text{trn},i}^{\mathbf{M}} \) follows the same steps of feature extraction, vector quantization and likelihood extraction as performed with a training signature, resulting in the likelihood vector \( \mathbf{D}(\mathbf{O}_{\text{trn},i}^{\mathbf{M}}) \) (see Fig. 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 development database. Section 4 explains the partitioning of each dataset used in this work.
3.2 Bank of HMMs

Let $\mathcal{M}^i = \{w_1 \cup w_2\}$ be the bank of left-to-right HMMs, where $w_i = \{\lambda_1^{(i)}, \lambda_2^{(i)}, \ldots, \lambda_{10}^{(i)}\}$ is the set of $R$ HMMs of the genuine class $C_i$, and $w_2 = \{\lambda_1^{(g)}, \lambda_2^{(g)}, \ldots, \lambda_{10}^{(g)}\}$ is the set of $S$ HMMs of the impostor’s class $C_2$. Each HMM in $w_1$ 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 $w_2$ 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 Fig. 4). Given the set of observation sequences $O_{trn}^i = \{O_1^i, O_2^i, \ldots, O_r^i\}$ extracted from a training signature $I_{trn}^i$, the vector $D(O_{trn}^i; \mathcal{M}^i)$ is obtained by computing the likelihoods of $O_{trn}^i$ for each HMM in $\mathcal{M}^i$, that is,

$$D(O_{trn}^i; \mathcal{M}^i) = \begin{pmatrix}
    p(O_1^i|\lambda_1^{(g)}) \\
    p(O_1^i|\lambda_2^{(g)}) \\
    \vdots \\
    p(O_r^i|\lambda_1^{(g)}) \\
    p(O_r^i|\lambda_2^{(g)}) \\
    \vdots \\
    p(O_r^i|\lambda_{10}^{(g)})
\end{pmatrix}$$

If, for instance, $\lambda_1^{(g)}$ and $\lambda_2^{(g)}$ were trained with observation sequences extracted from the codebook $q = 10$, a compatible sequence from $O_{trn}^i$, that is, $O_3^{(10)}$, must be sent to both. Finally, the likelihood vector is labeled according to the class of $O_{trn}^i$. It is worth noting that, if $O_{trn}^i$ 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 $O_{trn}^i$ 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 $C_2$.

This procedure is performed on all $O_{trn}^i$, $1 \leq i \leq N$, and the resulting likelihood vectors $D(O_{trn}^i; \mathcal{M}^i)$, $1 \leq i \leq N$, 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 $O_{trn}^i$, for $1 \leq i \leq N$, be the sequences of observations extracted from the training signatures of writer $i$ and $D(O_{trn}^i; \mathcal{M}^i)$, for $1 \leq i \leq N$, be their corresponding likelihood vectors—referred in this section as training vectors. From the first training vector, that is,
D(O_{tr}^{r}, M^{t}), the specialized Random Subspace Method selects, with replacement, R’ likelihoods from its R first positions (corresponding to w_{i}), and S’ likelihoods from its S last positions (corresponding to w_{k}). Then, for each training vector D(O_{tr}^{r}, M^{t}), for 1 ≤ i ≤ N, the selected positions, R’ and S’, are used to form a new vector D(O_{tr}^{r}, M^{t}), which is stored in the training set T’. Finally, the vectors D(O_{tr}^{r}, M^{t}), for 1 ≤ i ≤ N, in T’, are used to train two-class classifiers c_i, where r, 1 ≤ r ≤ R, is the actual random subspace. This procedure is repeated for R different random subspaces, resulting in a pool C of R different classifiers.

3.4. A new strategy for dynamic ensemble selection

Let O_{dej}^{r}, for 1 ≤ j ≤ M, be the sequences of observations extracted from the Dynamic Selection database of writer i, and D(O_{dej}^{r}, M^{t}) be their corresponding likelihood vectors, for 1 ≤ j ≤ M. For each DS vector D(O_{dej}^{r}, M^{t}), an Output Profile (OP) is calculated as follows. First, the DS vector is input to all classifiers c_r, r = 1, 2, ..., R, in the pool of classifiers C. Each classifier c_r receives only the vector positions related to its respective subspace. Therefore, the resulting output labels are stored as a vector to form a DS Output Profile, OP(D(O_{dej}^{r}, M^{t})). 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 D(O_{test}^{r}, M^{t}) is presented to the off-line SV system, four main steps are performed (see Fig. 5). First, the Output Profile OP(D(O_{test}^{r}, M^{t}))) is calculated, as performed for the DS vectors. Second, the Euclidean distance is computed between OP(D(O_{test}^{r}, M^{t}))) 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.

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 D(O_{test}^{r}, M^{t}), 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, OP(D(O_{test}^{r}, M^{t}))), is calculated. Its K-nearest DS-OPs, OP(D(O_{dej}^{r}, M^{t}))), 1 ≤ k ≤ K, are then obtained by using the Euclidean distance. For each classifier c_r, r = 1, 2, ..., R, in the pool C, the OP-ELIMINATE algorithm verifies if c_r is able to classify the K corresponding DS vectors D(O_{dej}^{r}, M^{t})), 1 ≤ k ≤ K. 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 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 submits a vote on the test vector, where final classification label L is obtained by using the majority vote rule.

**Algorithm 1.** OP-ELIMINATE.

**Inputs:**
- the number of nearest neighbors, K
- the input vector, D(O_{test}^{r}, M^{t})
- the number of random subspaces, R
- the pool of classifiers, C
- the Output Profiles, OP(D(O_{dej}^{r}, M^{t}))), 1 ≤ j ≤ M

**Outputs:**
- the final classification label, L

1: 1: STEP 1:
2: calculate the Output Profile OP(D(O_{test}^{r}, M^{t})))
3: 2: STEP 2:
4: find its K nearest Output Profiles by calculating the Euclidean distance between OP(D(O_{test}^{r}, M^{t}))) and each OP(D(O_{dej}^{r}, M^{t}))), where 1 ≤ j ≤ M
5: 3: STEP 3:
6: set count=1; // number of classifiers added to the ensemble
7: 4: for each classifier c_r, r = 1, 2, ..., R, in C do

---

2 During dynamic selection, D(O_{dej}^{r}, M^{t}) is referred as a DS vector.
added to the ensemble. OP-ELIMINATE. For each neighbor $r \leq K$ and its $K$-nearest neighbors in the dynamic selection database, the objective of this second variant is to find for each neighbor $r$, $1 \leq k \leq K$, an ensemble of up to $K$ classifiers that correctly classify it. First, the test Output Profile, $OP(D_{\text{test}}(O_i \hat{M}^j))$, and its $K$-nearest DS-OPs, $OP(D_{\text{ds}}(O_i \hat{M}^j))$, $1 \leq k \leq K$, are obtained, such as performed for OP-ELIMINATE. For each neighbor $k$, and for each classifier $c$, $r = 1, 2, \ldots, \mathcal{R}$, in the pool $c$, the OP-UNION algorithm then verifies if $c$ is able to classify the DS vector $D_{\text{ds}}(O_i \hat{M}^j)$ correctly. If so, $c_i$ 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, $D(O_{\text{test}}(O_i \hat{M}^j))$
- the number of random subspaces, $\mathcal{R}$
- the pool of classifiers, $\mathcal{C}$
- the Output Profiles, $OP(D_{\text{ds}}(O_i \hat{M}^j))$, $1 \leq j \leq M$

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

1: STEP 1:
2: calculate the Output Profile $OP(D(O_{\text{test}}(O_i \hat{M}^j)))$
3: STEP 2:
4: find its $K$ nearest Output Profiles by calculating the Euclidean distance between $OP(D(O_{\text{test}}(O_i \hat{M}^j)))$ and each $OP(D_{\text{ds}}(O_i \hat{M}^j))$, where $1 \leq j \leq M$
5: STEP 3:
6: for each neighbor $r$, $1 \leq k \leq K$, do
7: set $\text{count} = 1$; // number of classifier added to the ensemble
8: for each classifier $c$, $r = 1, 2, \ldots, \mathcal{R}$, in $c$ do
9: if classifier $c$ classifies $D_{\text{ds}}(O_i \hat{M}^j)$ correctly and $\text{count} \leq K$ then
10: store $c$ in the ensemble $E_k$, that is, $E_k(\text{count}) = c_i$
11: increment the variable $\text{count}$;
12: end if
13: end for
14: end for
15: STEP 4:
16: use all $E_k$, $1 \leq k \leq K$, to classify $D(O_{\text{test}}(O_i \hat{M}^j))$
17: return the final classification label $\mathcal{L}$ by majority voting

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

<table>
<thead>
<tr>
<th>Phase</th>
<th>Bank of HMMs</th>
<th>Pool of SVMs</th>
<th>OP-ELIM./UNION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>$O(\text{BL})$</td>
<td>$O(\text{RD}^N \times V)$</td>
<td>$O(K\mathcal{R})$</td>
</tr>
<tr>
<td>Test</td>
<td>$O(\text{BL})$</td>
<td>$O(\text{RD}^V \times V)$</td>
<td>$O(K\mathcal{R})$</td>
</tr>
</tbody>
</table>

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 (SVMs) with radial basis function (RBF) kernel [31] in the discriminative stage. Assume a left-to-right HMM with $S$ 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 $O(L S)$ per iteration, in terms of both time and memory [32,33]. For the SVM-RBF, the time and memory complexity during training is $O(D N^2 \times V)$, where $N$ is the size of the training set and $D$ is the number of input dimensions. During operations, the evaluation of each input sample has a time and memory complexity of $O(DV)$, 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 composed of a bank of $B$ HMMs and a pool of $R$ SVMs, respectively. In the discriminative stage, $z$ genuine samples versus $\beta$ forgery samples are used to train the SVMs (i.e., $N = z + \beta$), while only $z$ 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.

4. Experimental methodology

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

- **Scenario1**—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.
- **Scenario2**—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 four genuine signatures per writer, and new genuine samples available overtime are incrementally 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 [46,23,35], and the GPDS database [12], used by other researchers [5,36–38]. 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, 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 400 × 1000 pixels, at resolution of 300 dpi. The signatures were provided by 168 writers and are organized in two sets: the development database (DBdev) and the exploitation database (DBexp).

DBdev 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 (Tjexp,20) and 10 for validation (Vjexp,10). The remaining 10 samples, available for test, are not employed in this work.

DBexp 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 (Tjexp,20) and 10 for validation (Vjexp,10). In the second scenario, 4, 8 and 12 genuine samples are available for training, taken from Tjexp,10. The test set is the same for Scenarios 1 and 2, that is, each writer in DBexp has 10 genuine samples (Tjdev,10), 20 random forgery samples (Tjrand,10) and 10 skilled forgery samples (Tjskil,10), where the random forgeries are genuine samples randomly selected from other writers in DBexp. A similar grid resolution along with pixel density features have been successfully applied to this database in [6].

Given a writer i enrolled to the system, DBdev and DBexp 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 DBdev, for 1 ≤ j ≤ 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, w2, 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 DBdev and used as random forgeries to train a pool of SVMs. In this case, the indice j is not specified in the training set T (as shown in second row, fourth column). A similar procedure is applied to the validation set 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 DBexp and the remaining 140, as DBdev. For each writer in both DBexp and DBdev, there are 24 genuine signatures and 30 skilled forgeries. In the literature, only 80–160 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 DBdev, 14 genuine samples are available for training (Tjdev,14) and 10 for validation (Vjdev,10); while in DBexp, each writer j has 14 genuine samples available for training (Tjexp,14) and 10 for test (TSjexp,10). Moreover, 10 random forgeries (TSjrand,10) and 10 skilled forgeries (TSjskil,10) are used for test, where the random forgeries are genuine samples randomly selected from other writers in DBexp. During the comparative analysis performed with systems in the literature (presented in Section 5.4), 30 skilled forgeries (TSjskil,30) are used for test, instead of 10.

In the following sections, the GPDS database is referred as GPDS-160, since a set of 160 writers (that is, DBexp) is actually modeled by the proposed system.

4.2. Grid segmentation

After conversion to black and white using Otsu’s binarization method [39], the signature images of the Brazilian SV database (composed of 400 × 1000 pixels) are divided into 62 horizontal cells of 64 pixels, at resolution of 300 dpi. 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 51 × 82 pixels to 402 × 649 pixels), they are represented in a grid of 400 × 650 pixels, and segmented in 65 horizontal cells of 40 × 10 pixels.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Datasets for a specific writer i, using the Brazilian SV database.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset name</td>
<td>Task</td>
</tr>
<tr>
<td>(a) Scenario1—design</td>
<td></td>
</tr>
<tr>
<td>DBhmm</td>
<td>HMM training</td>
</tr>
<tr>
<td>DBsvm</td>
<td>SVM training</td>
</tr>
<tr>
<td>DBsvm</td>
<td>SVM grid search</td>
</tr>
<tr>
<td>DBroc</td>
<td>ROC curve</td>
</tr>
<tr>
<td>(b) Scenario2—design</td>
<td></td>
</tr>
<tr>
<td>DBhmm</td>
<td>HMM training</td>
</tr>
<tr>
<td>DBsvm</td>
<td>SVM training</td>
</tr>
<tr>
<td>DBsvm</td>
<td>SVM grid search</td>
</tr>
<tr>
<td>DBroc</td>
<td>ROC curve</td>
</tr>
<tr>
<td>(c) Scenarios1 and2—verification</td>
<td></td>
</tr>
<tr>
<td>Dataset name</td>
<td>Genuine samples</td>
</tr>
<tr>
<td>DBhmm</td>
<td>TSTjtrue,10</td>
</tr>
</tbody>
</table>
Table 3
Datasets for a specific writer i, using the GPDS database.

<table>
<thead>
<tr>
<th>Dataset name</th>
<th>Task</th>
<th>Genuine samples</th>
<th>Random forgery samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Scenario2—design</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$DB_{hmm}$</td>
<td>HMM training</td>
<td>4, 8, 12 from $T_{exp,14}^i$</td>
<td>4, 8, 12 from $T_{exp,14}^i$ for $1 \leq j \leq 140$</td>
</tr>
<tr>
<td>$DB_{svm}$</td>
<td>SVM training</td>
<td>4, 8, 12 from $T_{svm,14}^i$</td>
<td>4, 8, 12 from $T_{svm,14}^i$ for $1 \leq j \leq 140$</td>
</tr>
<tr>
<td>$DB_{grid}$</td>
<td>SVM grid search</td>
<td>4, 8, 12 from $V_{svm,14}^j$</td>
<td>$v_{svm,14}^j$ for $1 \leq j \leq 140$</td>
</tr>
<tr>
<td>$DB_{rc}$</td>
<td>ROC curve</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$DB_{ds}$</td>
<td>Dynamic selection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(b) Scenario2—verification</td>
<td>Genuine samples</td>
<td>Forgery samples</td>
<td></td>
</tr>
<tr>
<td>$DB_{est}$</td>
<td>$TST_{true,10}^t$</td>
<td>$TST_{true,10}^t$ + $TST_{false,10}^t$</td>
<td></td>
</tr>
</tbody>
</table>

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

For real-world applications, the definition of writer-adapted grid resolutions may be unfeasible, specially when dealing with a limited number of signature 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 shown in the example of Fig. 6.

4.3. Training of the generative stage

Twenty-nine (29) different codebooks $q$ (1 ≤ $q$ ≤ 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 $DB_{hmm}$ (i.e., $T_{exp,20}^i$) 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, $w_1$, is composed of a variable number HMMs (i.e., 29 × ($L_{min} - 1$)) that depends on the writer’s signature size. On the other hand, to compose the impostor’s space, $w_2$, 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 algorithm [3], and at each iteration $t$, a error measure $E_t$ is calculated as

$$E_t = \frac{P(O_q^{(t)} / \lambda_{(t)}) - P(O_q^{(t-1)} / \lambda_{(t-1)})}{P(O_q^{(t)} / \lambda_{(t)}) + P(O_q^{(t-1)} / \lambda_{(t-1)})}$$

where $P(O_q^{(t)} / \lambda_{(t)})$ and $P(O_q^{(t-1)} / \lambda_{(t-1)})$ represent the joint probabilities of the training sequences $O_q^i$ have been generated by the HMM $\lambda$ 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., $V_{exp,10}$, 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 $O_q^i$, a set of HMMs is trained by varying the number of states from 2 to $\frac{1}{2}$ of the sequence’s size. The use of a single training sequence per HMM was previously 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 $\gamma$ are found through a grid search technique. For each different pair $(C, \gamma)$, 10 different SVMs are trained using a variant of the cross-validation method, where the genuine samples perform the usual validation method, where the genuine samples perform the usual rotation, and the random forgery samples are changed each time. Since the number of genuine samples in $DB_{grid}$ 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 replacement, each time that a new SVM training is performed. Finally, the error rates provided by the 10 SVMs are averaged, and the pair $(C, \gamma)$ 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, $DB_{svm}$, 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 DBi is used during the search. The value of K is defined as being half of the number of genuine samples in DBi. 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 \( D(O_{i,j}^{D}, M^t) \) to all 100 SVMs and then fusing their corresponding output labels by majority vote. If a tie vote is obtained, the final output label is randomly chosen.

The decision templates (DTs) are a well-known dynamic selection method in the multi-classifier system (MCS) community [42]. First, each DS vector \( D(O_{i,j}^{D}, M^t) \), 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 \( D(P(D(O_{i,j}^{D}, M^t))) \) is composed of two columns and 100 lines, where each cell contains the value 1 or 0. For instance, if the 100th SVM classifies \( D(O_{i,j}^{D}, M^t) \) as belonging to the genuine class \( C_i \), 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 \( C_j \)) will contain the value 0. Then, a decision template (DT) is calculated for each class \( C_j \) by averaging the DPs of the DS vectors belonging to this class. When a test vector \( D(O_{i,j}^{D}, M^t) \) is presented, its decision profile matrix \( D(P(D(O_{i,j}^{D}, M^t))) \) is calculated and compared to the decision templates \( DT(C_j) \). The comparison is done by using the Euclidean distance, and the higher the similarity between \( D(P(D(O_{i,j}^{D}, M^t))) \) and \( DT(C_j) \), the higher the support for class \( C_j \). Finally, the most likely \( DT(C_j) \) is selected and the output label the most represented in this template is assigned to \( D(O_{i,j}^{D}, M^t) \).

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] (referred 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 \( DB_{perc} \)) is computed. Then, the scores providing a same value of cumulative frequency, \( \gamma \), are used as thresholds to compute the operating points \( (TAR(\gamma), FRR(\gamma)) \). Finally, the operating points associated with the same \( \gamma \) (and related to different users) are averaged. Note that \( \gamma \) can be viewed as the true negative rate \( (TNR=\text{ratio of random forgery scores correctly classified to the total of random forgery scores}) \) 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 \( \tau_{user1}(0.3) \approx -5.6 \) and \( \tau_{user2}(0.3) \approx -6.4 \). 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 \(-6.4 \).

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, FNR and FPR are calculated by using the user-specific thresholds associated to a given \( \gamma \) of the averaged ROC curves. The average error rate (AER), also computed for a given \( \gamma \), indicates the total error of the system, where FNR and FPR are averaged taking into account the a priori probabilities.

When the Brazilian SV database is used, the AER is calculated with respect to three forgery types: random, simple and skilled (see \( DB_{list} \) of Table 2), that is,

\[
AER = (FRR + FAR_{rand} + FAR_{simp} + FAR_{skil})/4
\]  

(3)

While for GPDS database, FAR is calculated with respect to random and skilled forgeries (see \( DB_{list} \) of Table 3), that is,

\[
AER = (FRR + FAR_{rand} + FAR_{skil})/3
\]  

(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 \( DB_{hhmm} \) (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 \( \varepsilon \) (see Eq. (2)). This resulted in an AER (\( \gamma = 1.0 \)) of 8.5% on test data. To reduce error rates and exploit the suboptimal 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 Fig. 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 $DB_{svm}$ (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 of 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 Chebyshev’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]. Fig. 9 shows the corresponding AERs curves.

5.2. Scenario 1—abundant data

In this experiment, each $DB_{i,s}$ is composed of 10 genuine samples supplied by writer $i$ (in $DB_{genuine}$) 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 $DB_{exp}$. Fig. 8 shows the averaged ROC curves obtained with scores produced from 100 different SVMs using $DB_{roc}$, while Fig. 9 presents the AERs curves on test data ($DB_{tst}$), as a function of operating points ($\gamma$).

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.

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 to the availability of partial knowledge during system design. Generally, 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.

Fig. 8. Averaged ROC curves obtained with scores produced from 100 different SVMs using $DB_{roc}$ (from Brazilian data), under Scenario 1.

![Averaged ROC curves](image)

Table 4

<table>
<thead>
<tr>
<th>Method</th>
<th>FRR</th>
<th>FAR$_{random}$</th>
<th>FAR$_{sample}$</th>
<th>FAR$_{dissim}$</th>
<th>AER</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP-UNION</td>
<td>3.33</td>
<td>1.67</td>
<td>3.83</td>
<td>34.83</td>
<td>10.92</td>
</tr>
<tr>
<td>OP-ELIMINATE</td>
<td>4.67</td>
<td>1.33</td>
<td>2.83</td>
<td>30.00</td>
<td>9.71</td>
</tr>
<tr>
<td>KNORA-UNION</td>
<td>2.17</td>
<td>2.50</td>
<td>7.00</td>
<td>45.33</td>
<td>14.25</td>
</tr>
<tr>
<td>KNORA-ELIMINATE</td>
<td>2.33</td>
<td>2.67</td>
<td>6.33</td>
<td>44.33</td>
<td>13.92</td>
</tr>
<tr>
<td>Decision templates</td>
<td>2.17</td>
<td>2.17</td>
<td>7.17</td>
<td>45.50</td>
<td>14.25</td>
</tr>
<tr>
<td>Combination of 100 SVMs</td>
<td>2.00</td>
<td>2.67</td>
<td>8.17</td>
<td>47.33</td>
<td>15.04</td>
</tr>
<tr>
<td>Static selection [35]</td>
<td>2.17</td>
<td>8.00</td>
<td>5.83</td>
<td>35.17</td>
<td>12.79</td>
</tr>
<tr>
<td>Baseline [4]</td>
<td>0.33</td>
<td>12.17</td>
<td>20.67</td>
<td>78.83</td>
<td>28.00</td>
</tr>
</tbody>
</table>

Table 5

<table>
<thead>
<tr>
<th>Method</th>
<th>FRR</th>
<th>FAR$_{random}$</th>
<th>FAR$_{sample}$</th>
<th>FAR$_{dissim}$</th>
<th>AER</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP-UNION</td>
<td>8.17</td>
<td>0.67</td>
<td>0.67</td>
<td>14.00</td>
<td>5.88</td>
</tr>
<tr>
<td>OP-ELIMINATE</td>
<td>7.50</td>
<td>0.33</td>
<td>0.50</td>
<td>13.50</td>
<td>5.46</td>
</tr>
<tr>
<td>KNORA-UNION</td>
<td>8.17</td>
<td>0.67</td>
<td>0.67</td>
<td>14.67</td>
<td>6.04</td>
</tr>
<tr>
<td>KNORA-ELIMINATE</td>
<td>7.83</td>
<td>0.67</td>
<td>0.67</td>
<td>14.17</td>
<td>5.83</td>
</tr>
<tr>
<td>Decision templates</td>
<td>8.67</td>
<td>0.50</td>
<td>0.67</td>
<td>15.17</td>
<td>6.25</td>
</tr>
<tr>
<td>Combination of 100 SVMs</td>
<td>8.83</td>
<td>0.50</td>
<td>0.67</td>
<td>15.33</td>
<td>6.33</td>
</tr>
<tr>
<td>Static selection [35]</td>
<td>13.50</td>
<td>0.00</td>
<td>0.17</td>
<td>8.33</td>
<td>5.30</td>
</tr>
<tr>
<td>Baseline [4]</td>
<td>12.67</td>
<td>0.33</td>
<td>1.17</td>
<td>19.83</td>
<td>8.50</td>
</tr>
</tbody>
</table>

Tables 4 and 5 present the overall results for $\gamma = 0.90$ and $\gamma = 1.0$, respectively.

5.3. Scenario 2—sparse data

In this experiment, each $DB_{i,s}$ is composed of 4, 8 and 12 genuine samples supplied by writer $i$ (in $DB_{genuine}$) 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 $DB_{exp}$. Fig. 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.

Figs. 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., $DB_{trn}$ and $DB_{sym}$), and $\#ds$ indicates the number of genuine signatures in $DB_{ds}$. 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, four genuine signatures versus four 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.

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 to gradually improve the system performance by adding new genuine signatures to $DB_{ds}$.

First, $DB_{ds}$ is composed of four genuine signatures versus 1080 random forgeries from $DB_{dev}$, as performed in the previous experiment. Then, $DB_{ds}$ is updated twice, by adding four new genuine signatures each time. Figs. 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 eight genuine samples and 12 genuine samples, from the previous experiment, are also presented in these figures.

The additional newly obtained genuine samples in DB\(\text{trn}\) improved system performance in almost all operating points. With the Brazilian SV database (see Fig. 13), the performance of the system using four genuine samples for training and eight for dynamic selection is comparable to that of using eight 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 Fig. 14), the performance of the system using four 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\(\text{trn}\) 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 four genuine samples and using 8 and 12 samples for dynamic selection, respectively. Therefore, incremental updating of DB\(\text{ds}\) represents a viable measure to improve system performance, and may be used in conjunction 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.

Comparisons with other systems are 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 FNR approximately equal to the FAR - is employed. Two operating points are chosen from the test scores: one regarding genuine signatures versus random forgeries, and a second regarding genuine signatures versus 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\(\text{exp}\). Our systems are based on two independent datasets: DB\(\text{trn}\) is employed to generate codebooks and to train the impostor’s class and DB\(\text{exp}\) 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.

5.5. System complexity

In Scenario 1, \(Q(L_{\text{min}}-1)\) HMMs are trained per writer, where \(Q\) is the number of codebooks (i.e., 29) and \(L_{\text{min}}\) is the size of the

<table>
<thead>
<tr>
<th>Reference</th>
<th>FRR</th>
<th>FAR(_{\text{random}})</th>
<th>FAR(_{\text{simple}})</th>
<th>FAR(_{\text{skilled}})</th>
<th>AER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batista et al. [4]</td>
<td>9.83</td>
<td>0.00</td>
<td>1.00</td>
<td>20.33</td>
<td>7.79</td>
</tr>
<tr>
<td>Bertolini et al. [23]</td>
<td>11.32</td>
<td>4.32</td>
<td>3.00</td>
<td>6.48</td>
<td>6.28</td>
</tr>
<tr>
<td>Justin et al. [48]</td>
<td>2.17</td>
<td>1.23</td>
<td>3.17</td>
<td>36.57</td>
<td>7.87</td>
</tr>
<tr>
<td>Santos et al. [47]</td>
<td>10.33</td>
<td>4.41</td>
<td>1.67</td>
<td>15.67</td>
<td>8.02</td>
</tr>
<tr>
<td>OP-ELIMINATE ((\gamma = 1.0))</td>
<td>7.50</td>
<td>0.33</td>
<td>0.50</td>
<td>13.50</td>
<td>5.46</td>
</tr>
</tbody>
</table>
On the other hand, $aQ(\frac{1}{3}L-1)$ HMMs are trained per writer in Scenario 2, where $a$ 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 $a=3$ 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 $L_{\text{min}}$ 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, $w_1$, $29 \times (33-1)$ HMMs were trained per writer in Scenario 1, and $4 \times 29 \times (12-1)$ HMMs were trained per writer in Scenario 2, when $a=4$. 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.

## 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 (composed 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.

### Table 8

<table>
<thead>
<tr>
<th>Reference</th>
<th>Database</th>
<th>Technique</th>
<th>#trn</th>
<th>FRR (%)</th>
<th>FAR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Genuine signatures versus random forgeries</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ferrer et al. [5]</td>
<td>GPDS-160 HMM classifiers</td>
<td>4</td>
<td>4.30</td>
<td>3.80</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>2.50</td>
<td>2.40</td>
<td></td>
</tr>
<tr>
<td>Ferrer et al. [5]</td>
<td>GPDS-160 HMM classifiers</td>
<td>12</td>
<td>2.20</td>
<td>3.30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SVM classifiers</td>
<td></td>
<td></td>
<td>3.23</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Euclidean distance</td>
<td></td>
<td></td>
<td>2.65</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.13</td>
<td></td>
</tr>
<tr>
<td>Vargas et al. [38]</td>
<td>GPDS-100 LBP + GLCM features</td>
<td>5</td>
<td>3.75</td>
<td>3.75</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LBP features</td>
<td></td>
<td></td>
<td>4.59</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GLCM features</td>
<td></td>
<td></td>
<td>6.40</td>
<td></td>
</tr>
<tr>
<td>Vargas et al. [38]</td>
<td>GPDS-100 LBP + GLCM features</td>
<td>10</td>
<td>1.76</td>
<td>1.76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LBP features</td>
<td></td>
<td></td>
<td>2.41</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GLCM features</td>
<td></td>
<td></td>
<td>4.31</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.31</td>
<td></td>
</tr>
<tr>
<td>OP-UNION</td>
<td>GPDS-160 HMM + SVM classifiers</td>
<td>4</td>
<td>7.75</td>
<td>6.56</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>8</td>
<td>5.38</td>
<td>5.44</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>12</td>
<td>4.50</td>
<td>5.19</td>
<td></td>
</tr>
</tbody>
</table>

### Table 9

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

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$w_1$</th>
<th>HMMs per writer</th>
<th>HMM states</th>
<th>SVM inputs</th>
<th>SVs per SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20 928</td>
<td>2–33</td>
<td>$15+15$</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 1276</td>
<td>2–12</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8 2552</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12 3828</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The system proposed in this paper combines 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 strategies (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, four genuine signatures versus four 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 \( \gamma \) 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 \( \gamma \) value close to 1. Lower amounts would translate to operating points related to low FARs, 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.

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

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 Department 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 Ph.D. 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 250 scientific publications including journals and conference proceedings. He was co-chair of the program committee of CIFED’98 (Conférence Internationale Francophone sur l’Écrit et le Document, Québec, Canada) and IFHWR04 (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.