ARTICLE IN PRESS

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Pattern Recognition Letters xxx (2004) xxx–xxx

Pattern Recognition Letters

www.elsevier.com/locate/patrec

A comparison of SVM and HMM classifiers in the off-line signature verification

Edson J.R. Justino [a,*], Flávio Bortolozzi [a], Robert Sabourin [b]

[a] PUCPR—Pontifícia Universidade Católica do Paraná, Rua Imaculada Conceiçáo, 1155, Curitiba, CEP 80215-901, PR, Brazil
[b] ÉTS—École de Technologie Supérieure, 1100, rue Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Received 18 October 2004

Abstract

The SVM is a new classification technique in the field of statistical learning theory which has been applied with success in pattern recognition applications like face and speaker recognition, while the HMM has been found to be a powerful statistical technique which is applied to handwriting recognition and signature verification. This paper reports on a comparison of the two classifiers in off-line signature verification. For this purpose, an appropriate learning and testing protocol was created to observe the capability of the classifiers to absorb intrapersonal variability and highlight interpersonal similarity using random, simple and simulated forgeries.

Keywords: Classification; Support vector machine; Hidden Markov model; Signature verification

1. Introduction

There are essentially two problems underlying off-line signature verification. One is related to the number of samples to use for learning. In a real application, we are usually quite limited in the number of samples we can use for training (4–6 samples). The other is the ability of the system to discriminate among different types of forgeries (random, simple and simulated) (Sabourin et al., 1997). The random forgery is usually represented by a genuine signature sample, which belongs to a different writer not necessarily enrolled to the signature verification, as Fig. 1(b). The simple forgery is represented by a signature sample with the same semantic of the genuine writer's name without any attempt to imitate the genuine signature model, as Fig. 1(c). The simulated forgery is represented by a reasonable imitation of the genuine signature model, as Fig. 1(d). Table 1 shows database examples of genuine and forgeries

* Corresponding author. Tel./fax: +55 271 1356.
E-mail addresses: edson.justino@pucpr.br (E.J.R. Justino), flavio.bortolozzi@pucpr.br (F. Bortolozzi), robert.sabourin@etsmtl.ca (R. Sabourin).

0167-8655/$ - see front matter
doi:10.1016/j.patrec.2004.11.015

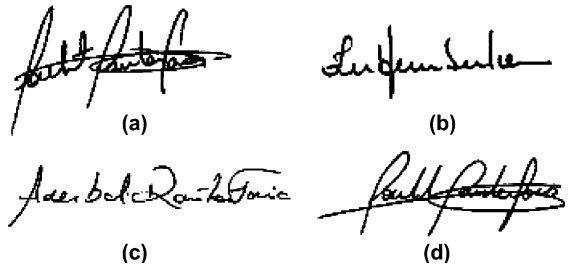


Fig. 1. Types of forgeries: (a) genuine signature; (b) random forgery; (c) simple forgery and (d) simulated forgery.

samples, for different writers. Any learning method that claims to solve those problems relies on its ability to perform the learning and classification tasks.

Usually, in signature verification, two different pattern classes are needed for the learning task, W_1 and W_2. W_1 represents a genuine signature set. W_2 represents a forged signature set. In the latter case, the genuine signatures of different writers are used like random forgeries, as Fig. 2. For real applications, like bank check authentication (Justino et al., 2001), simple and simulated forgeries are not used in the learning phase.

The main challenge in the learning task is to separate classes W_1 and W_2. In many cases, the threshold between them is not easy to find.

In the verification task, the challenge is to discriminate between the genuine and all types of forgeries, having in mind for training only a priori

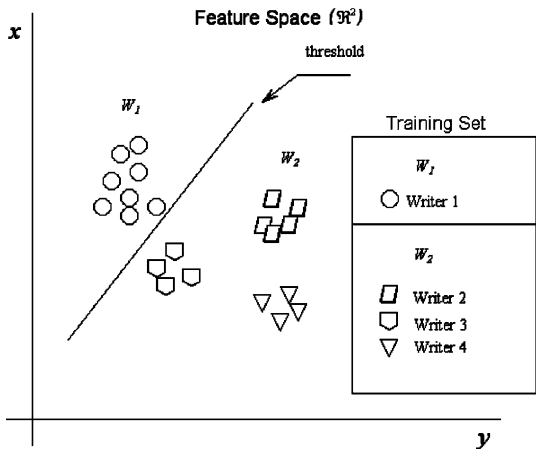
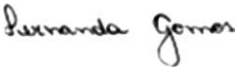
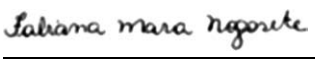
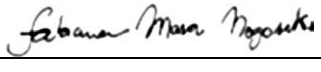
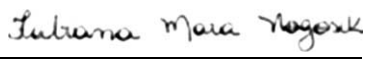


Fig. 2. Signature verification classes W_1 and W_2, which are present in the learning task. W_1 represents the class of genuine signatures and W_2 represents the class of forgeries.

knowledge of some random forgeries. This increases the verification task complexity especially when simple and simulated forgeries are challenging the verification system, as Fig. 3. Simulated forgeries, for example, are similar to genuine signatures in some ways. For this reason, some samples fit into the W_1 class and some nearly do so. This may even be the case with simple forgery samples.

In this paper, a comparison between SVM and HMM, in terms of the learning and verification tasks described above, is presented.

Table 1
Genuine and forgery signatures samples form database

Genuine signature	Simple forgery	Simulated forgery
		
		
		
		
		

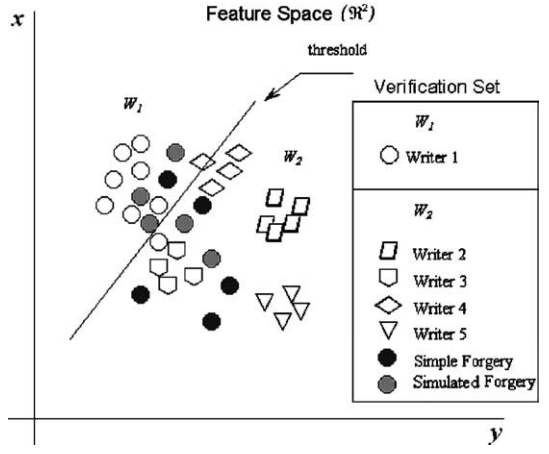


Fig. 3. The signature verification set usually present in the verification task. W_1 represents the class of genuine signatures and W_2 represents the class of forgeries.

2. Signature verification and HMM

In the last decade, the HMM has attracted the attention of many researchers in pattern recognition, and in handwriting, speech and signature verification (Elms, 1996; El Yacoubi et al., 1999; Justino et al., 2000). This statistical learning theory has the ability to absorb both the variability and the similarity between patterns. It is based on the empirical risk minimization (ERM) principle, which is the simplest of induction principles, where a decision rule is chosen. The decision rule is based on a finite number of known examples (training set).

The writer's signature model, in HMM learning process $\lambda = \{A,B,\pi\}$, is defined according to the individual writer's features. A correct topology choice is important in obtaining the best signature model. There are some topologies for the HMM models, each adapted to a particular case. In this study, a discrete left-to-right topology was chosen, because this is well adapted to the occidental handwriting motion, as Fig. 8.

In discrete models, two factors are important (Justino et al., 2000). The first is the number of states to be used. The second is the number of transitions among these states. The number of states depends on the signature length and the best results in terms of learning probability $p_t(O/\lambda)$ (Justino et al., 2002). The cross-validation procedure is another important factor in obtaining the best model in terms of the number of states.

For one specific number of states, the best validation probability $p_{cv}(O/\lambda)$ was used to define the most suitable probability model $p_t(O/\lambda)$. This model was used to define the threshold parameters, and makes it is possible to determine the acceptance and rejection thresholds taking into account a specific writer.

The medium threshold in Eq. (1) is defined by p_{tn}, which represents the learning probability logarithm normalized by the observation number L. The parameter L increases the personal characteristic, signature length. The α_1 and α_2 values are computed using a signature validation set, and are defined by the smallest average error rate. The error rate is correlated to the type I error rate (false rejection) and to the type II error rate (false acceptance). These α_1 and α_2 values are used to compute the acceptance borderlines p_i and p_s.

$$p_{tn}(O/\lambda) = \frac{\log p_t(O/\lambda)}{L} \tag{1}$$

$$p_i = p_{tn} - (p_{tn} \cdot \alpha_1) \tag{2}$$

$$p_s = p_{tn} + (p_{tn} \cdot \alpha_2). \tag{3}$$

In this study, the forward algorithm (Justino et al., 2000) was used to determine the verification probability p_v. The L value also normalizes the probability logarithm p_{vn}. Acceptance and rejection were defined by Eq. (5), as Fig. 4.

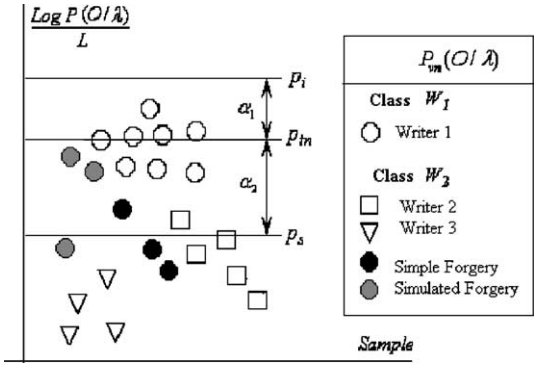


Fig. 4. Thresholds used to define the acceptance and rejection area in the verification process.

$$p_{vn}(O/\lambda) = \frac{\log p_v(O/\lambda)}{L} \quad (4)$$

$$p_s \leqslant p_{vn} \leqslant p_i. \quad (5)$$

3. Signature verification and SVM

SVM was developed by Vapinik (1998), and is a new technique in the field of statistical learning theory. It is based on the structural risk minimization principle (SRM). The SRM induction principle has two main objectives. The first is to control the empirical risk on the training data set. The second is to control the capacity of the decision functions used to obtain this risk value. The SVM's linearly learned decision function $f(x)$ is described by weight vector $\boldsymbol{w}$, a threshold b and input patterns $\boldsymbol{x}$.

$$f(x) = \text{sign}(\boldsymbol{w} \cdot \boldsymbol{x} + b). \quad (6)$$

Given a set of training vectors S_l Eq. (7) belonging to two separate classes, W_1 ($y_i = +1$) and W_2 ($y_i = -1$), the SVM finds the hyperplane with maximum Euclidian distance from the training set. According to the SRM principle, there will be just one optimal hyperplane with the maximal margin δ, defined as the sum of distances from the hyperplane to the closest points of the classes. This linear classifier threshold is the optimal separating hyperplane, as Fig. 5:

$$\boldsymbol{S}_l = ((\boldsymbol{x}_1, y_1), \ldots (\boldsymbol{x}_l, y_l)), \quad \boldsymbol{x}_i \in \Re^n, \quad y_i \in \{-1, +1\}. \quad (7)$$

In the case of non-separable training sets, the ith data point has a slack variable ξ_i, which represents the magnitude of the classification error, as Fig. 6. A penalty function $f(\xi)$ represents the sum of the misclassification errors Eq. (8).

$$f(\xi) = \sum_{i=1}^{l} \xi_i. \quad (8)$$

The SVM solution can be found by keeping the upper bound on the VC dimension small. (Burges, 1998; Weston, 1999; Vapinik, 1998) and by minimizing an upper bound on the empirical risk, i.e.

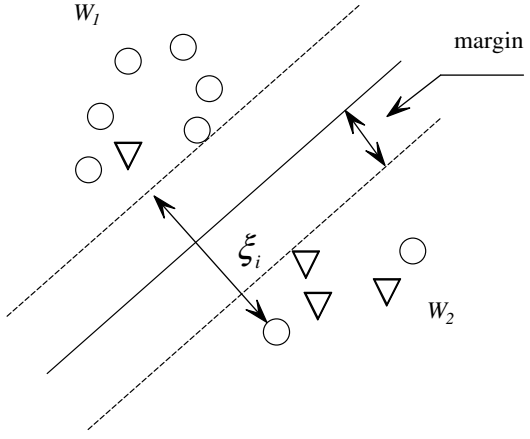


Fig. 6. The case of non-separable training sets.

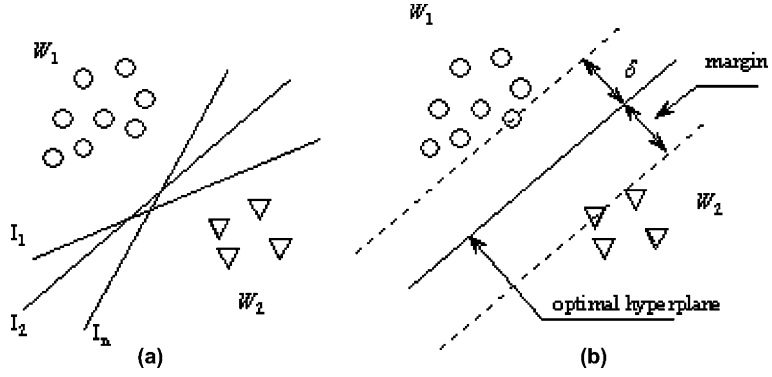


Fig. 5. Classification between two classes W_1 and W_2 using hyperplanes: (a) arbitrary hyperplanes li and (b) optimal separation hyperplane with a maximal margin.

the number of training errors, with the following minimization:

$$\min_{\bar{w},b,\bar{\xi}} = \frac{1}{2}\boldsymbol{w}\cdot\boldsymbol{w} + C\sum_{i=1}^{n}\xi_i, \tag{9}$$

where the regularization constant $C > 0$ determines the trade-off between the empirical error and the complexity term. Parameter C is chosen by the user, a large C corresponding to the assignment of a higher penalty to errors.

The bibliography presents many kernels used with success in pattern recognition problems. (Burges, 1998; Guo et al., 2002; Weston, 1999; Lee et al., 2002; Müller et al., 2001). The first kernels investigated for a pattern recognition problem were the linear kernel Eq. (10) and the polynomial kernel of degree d Eq. (11). The linear kernel was used successfully in some separable classes (Ben-Yacoub, 1998). However, in other cases (non-separable), as Fig. 6, the polynomial kernel presents a possible solution, even though the number of degrees is higher.

$$K(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}\cdot\boldsymbol{y}) \tag{10}$$

$$K(\boldsymbol{x}\cdot\boldsymbol{y}) = (\boldsymbol{x}\cdot\boldsymbol{y}+1)^d \tag{11}$$

4. Feature extraction

The advantages of grid-segmentation schemes have frequently been shown (Justino et al., 2001; Huang and Yan, 1997; Yingyong and Hunt, 1994; Sabourin and Genest, 1994). These references also demonstrate how a grid approach can be adapted to parts of the signature, according to their stability. The features obtained by the segmentation scheme are neatly correlated with the graphometric features, like signature length, signature height, spaces between signature blocks, skew and so on.

In this study, furthest from primitives obtained by the segmentation process, a set of graphometric features (static and pseudodynamic) was used to demonstrate the discrimination capability of the SVM and HMM classifiers. The pixels density X_{PD} (number of pixels inside the cell) and gravity center X_{GC} (gravity center distance in each cell) were used as static features. The stroke curvature X_{SC} (curvature angle of the bigger stroke inside de cell) and slant X_{SL} (predominant slant inside the cell) were used as pseudodynamic features, as Fig. 7.

In the HMM low-level feature extraction procedure, the grid is placed on the signature, as Fig. 8. Depending on the signature size, however, it is possible that the peripheral parts of the signature will be lost. After that, each column of vertical cells is converted into a feature vector, one for each primitive. The low-level feature vectors were used to create the observations sequence O_L where L represents the observation sequence number. For this purpose, a k-means algorithm was used by the Vector Quantization (VQ) procedure to covert low-level feature vectors to high-level feature vectors or codebooks (Justino et al., 2000–2002). The observation sequence number is an important writer's feature used in the learn/test procedure.

In order to guarantee an efficient VQ design for the variety of features used in the system, the multiple codebook technique was used, and the feature stream was subdivided into four different streams, one for each feature. Given a HMM state, the probability of generating the feature f, by seeing the ith label of the fth VQ codebook is denote as $p_{\mathrm{s}}^{f}(i_f)$. The complete state output probability p_{s} is computed by

$$p_{\mathrm{s}} = \prod_{f=1}^{4} p_{\mathrm{s}}^{f}(i_f). \tag{12}$$

In the SVM feature extraction procedure, the grid is still placed on the signature sample and the primitives are computed. The signature image is always adjusted on the left side of the rectangular area. This procedure reduces the matching discordances observed using the gravity center.

The union of the subsets of features Eq. (12) composes the SVM feature vector $\boldsymbol{x}$, as Fig. 9.

$$\boldsymbol{x} = \cup\{\boldsymbol{x}_{\mathrm{PD}}, \boldsymbol{x}_{\mathrm{SL}}, \boldsymbol{x}_{\mathrm{SC}}, \boldsymbol{x}_{\mathrm{GC}}\} \tag{13}$$

5. Evaluation protocol

For the HMM protocol, the database of 100 writers was divided into two parts, one containing

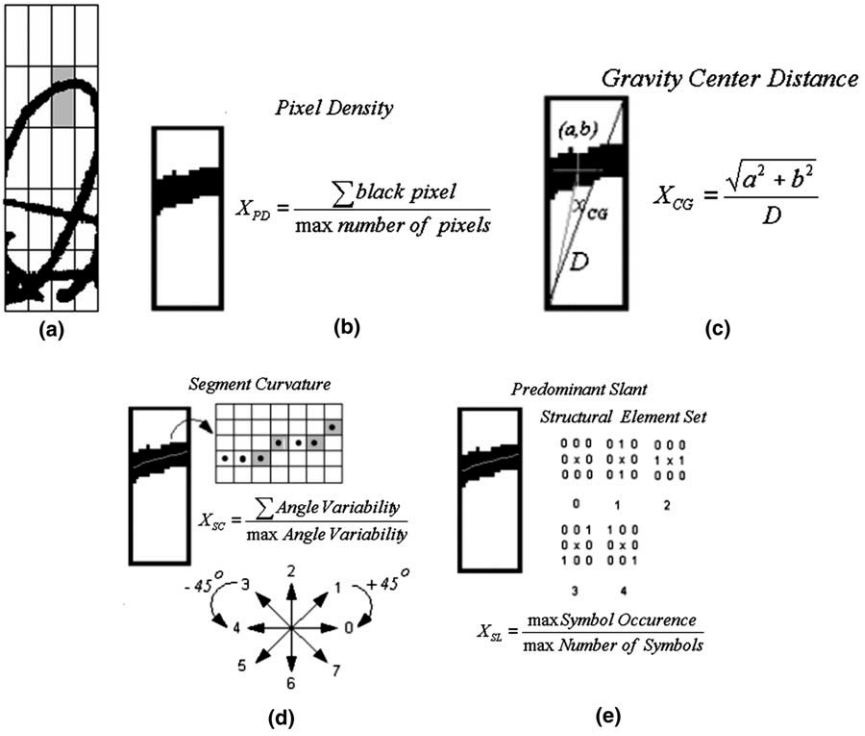


Fig. 7. The features extraction method: (a) segmented cell; (b) pixels density; (c) gravity center distance; (d) segment curvature and (e) predominant slant definition.

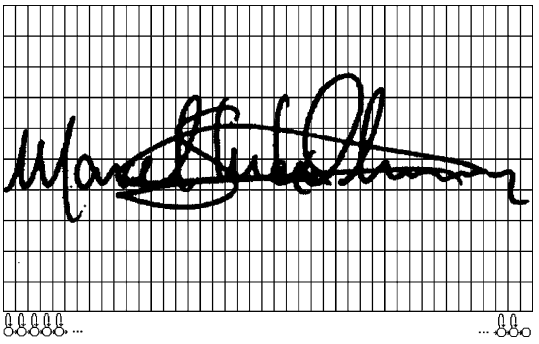


Fig. 8. The HMM segmentation example and the left–right topology.

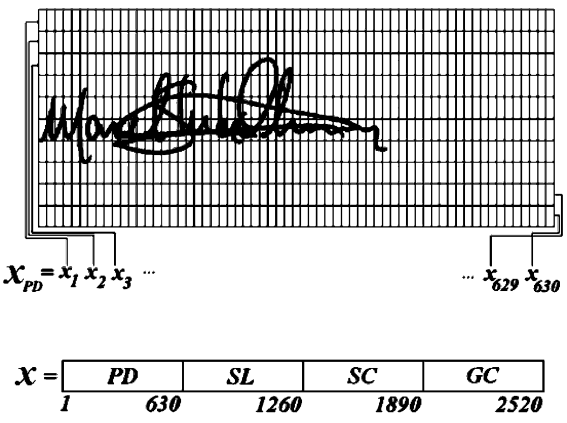


Fig. 9. SVM feature vector representation.

40 writers and the other containing the rest. The first database was used to carry out the first tests, to determine the VQ symbols, the α_1 and α_2 parameters and the best group of symbols and cells. The second was used exclusively for the tests.

For the learning database 20 signatures were used after being chosen at random. Another 10 were used as the validation database and the last 10 were chosen for verification.

The performance of the system was evaluated using the average error rate, obtained by calculating the types I and II average error rates, of all writers that participated in the experiment. In other words, for each writer 10 real signatures were tested and the sum of the remaining participant writer's signatures was used as false.

In the SVM protocol, all the databases were converted into a feature vector, as Fig. 9. The same 40 writers used in HMM protocol was used to provide a set of random forgeries by the SVM learning procedure, W_2 ($y_i = -1$), and to compute all SVM writers' models. The rest was used for the tests only (20 samples for training the model and 10 for testing). The same test samples were used to compose the random forgery test subset. For a specific writer, 59 writers were used as random forgery set, 10 genuine samples per writer.

The software SVM^{light} published by Joachims (2002) was used in the SVM experiments.

6. Experimental results

In the first HMM experiment the main objective was to find the better codebook size. For this purpose, a set of different codebook size was tested and individual features (60–150 with gap size 10). The results shown better for each feature, with codebook size equal 100, no significant error variability (~1%) was found.

In the SVM case, with the objective to establish a better empirical regularization constant value C, a first experiment was done. Using isolated feature and the linear kernel, different C values were tested (0–3500 with gap size equal 500). These results demonstrate the learning procedure's ability to discriminate between W_1 and W_2, independent of the penalty error commitment. For this reason an intermediated C value was used ($C = 1000$).

In this study, the polynomial kernel with different degree was tested, but the result shown

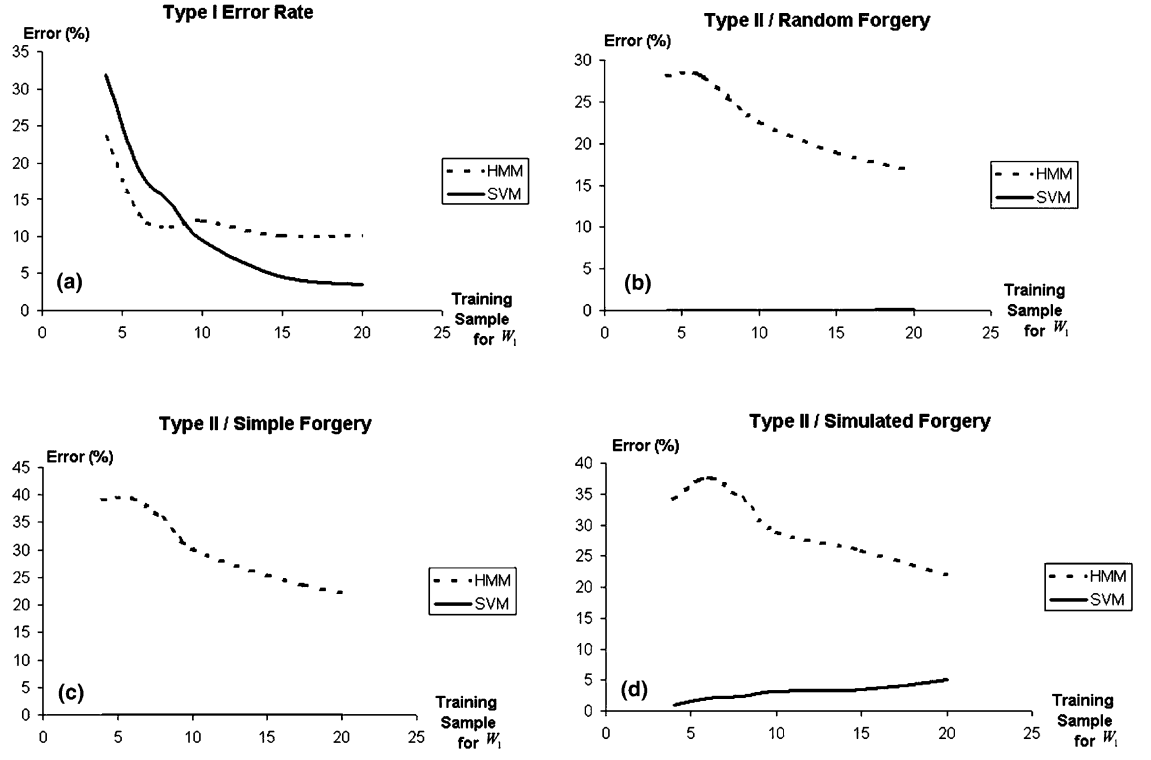


Fig. 10. Comparison results between SVM and HMM: (a) false rejection error rate; (b) false acceptance for random forgery; (c) false acceptance for simple forgery and (d) false acceptance for simulated forgery.

unsatisfactory compared with linear kernel. For this reason, only the linear kernel was shown.

Fig. 10 shows the results obtained with classifiers. Fig. 10(a) presents the rejection rate for both cases. The classifiers exhibited about the same behavior when the number of samples used for training went down. For a real application (4–6 samples), the results presented by both classifiers were still high, about 13% for HMM and 19% for SVM, using six genuine samples. The HMM classifier shown reductions on the forgeries error rate, when the number of training samples was increased, but still high compared with SVM. The SVM presented better stability when the number of samples went up, with small increases on simulated forgery (4%). This demonstrates the SVM's capability to absorb the intrapersonal variability and the interpersonal similarity, without previous knowledge (simple and simulated forgeries). The SVM classifier increases the type I error rate and reduces the type II error rate, especially in the simulated forgery case, as, when the number of samples used to training was decreased, as Fig. 10(d). This occurs because the model's capability to absorb the intrapersonal variability was reduced.

When the size of the training set is increased, the simulated forgery error rate grows. This phenomenon occurs because the model becomes more adapted to absorb the intrapersonal variability. In other words, increasing the number of training samples, the model is going to absorb the intrapersonal variability. Thus, this effect produces a reduction on the false rejection error rate, as depicted in Fig. 10(a). However, this adaptability allows the model to accept reasonable imitations (simulated forgeries), as illustrated in Fig. 10(d). A possible solution for this problem is to introduce new graphometric features to discriminate simulated forgeries more carefully.

7. Conclusion and future work

The main objective of this study was to compare SVM and HMM classifiers under two specific conditions, the first being the number of samples used for training, and the second being the use of different types of forgeries. Under both conditions, the SVM showed better results. However, in terms of random forgery acceptance and small number of samples used to training, the SVM showed promising results, demonstrating SVM's ability to identify simple and simulated forgeries without previous knowledge.

In a future work will be increased the database, implemented new graphometric features and a new protocol with the main objective to reduces the number of signature training samples and reduces the acceptation error rate in simulate forgery type.

References

Ben-Yacoub, S., 1998. Multi-modal data fusion for person authentication using SVM. IDIAP Research Report, Martigny—Valais—Suisse. Available from: http://www.idiap.ch, pp. 1–9.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. Data Mining Knowl. Discovery 2, 121–167.

Elms, A.L., 1996. The representation and recognition of text using hidden Markov models. Ph.D. Thesis, University of Surrey, UK.

El Yacoubi, A., Gilloux, M., Sabourin, R., Suen, C.Y., 1999. Unconstrained hand-written word recognition using hidden Markov models. IEEE Trans. Pattern Anal. Machine Intell. 2 (8), 752–760.

Guo, G., Jain, A.K., Zhang, H., 2002. Learning similarity measure for natural image retrieval with relevance feedback. IEEE Trans. Neural Networks 13 (4), 811–819.

Huang, K., Yan, H., 1997. Off-line signature verification based on geometric feature extraction and neural network classification. Pattern Recognition 30 (1), 9–17.

Joachims, T., 2002. Optimizing search engines using clickthrough data. ACM Conf. on Knowledge Discovery and Mining (KDD), pp. 1–10.

Justino, E.J.R., Bortolozzi, F., Sabourin, R., 2002. The interpersonal and intrapersonal variability influences on off-line signature verification using HMM, SIBGRAPI 2002. XV Brazilian Symposium on Computer Graphics and Image Processing. vol. 1. Fortaleza, Ceará, Brazil. pp. 197–202.

Justino, E.J.R., Bortolozzi, F., Sabourin, R., Justino, S., 2001. Off-line signature verification using HMM for random, simple and skilled forgeries, ICDAR 2001. Internat. Conf. on Document Analysis and Recognition. vol. 1. Seattle, USA. pp. 105–110.

Justino, E.J.R., Bortolozzi, F., Sabourin, R., 2000. An off-line signature verification system using HMM and graphometric features, DAS 2000. 4th IAPR Internat. Workshop on Document Analysis Systems, Rio de Janeiro, Brazil. pp. 211–222.

Lee, K., Chung, Y., Byun, H., 2002. SVM-based face verification with feature set of small size. 18th Electronics Lett. 38 (15), 787–789.

Müller, K., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B., 2001. An introduction to kernel-based learning algorithms. IEEE Trans. Neural Networks 12 (2), 181–202.

Sabourin, R., Genest, G., Prêteux, P., 1997. Off-line signature verification by local granulometric size distributions. Pattern Anal. Machine Intell. (PAMI) 19 (9), 976–988.

Sabourin, R., Genest, G., 1994. An extended-shadow-code based approach for off-line signature verification: Part I—Evaluation of the bar mask definition. 12th IAPR Internat. Conf. on Pattern Recognition, Jerusalem, Israel, pp. 450–455.

Vapinik, V., 1998. Statistical Learning Theory. Wiley, NY.

Weston, J.A.E., 1999. Extensions to the support vector method. Ph.D. Thesis. Royal Holloway College, University of London.

Yingyong, Q., Hunt, B.R., 1994. Signature verification using global and grid features. Pattern Recognition 22 (12), 1621–1629.