A Discrete HMM algorithm for On-Line Signature Verification with Pen Position Trajectories

DAIGO MURAMATSU, TAKASHI MATSUMOTO
Department of Electrical Engineering and Bioscience
Waseda University
3-4-1 Okubo Shinjuku-ku Tokyo
JAPAN

Abstract: -Authentication of identity is rapidly becoming an important issue. Signature verification is a promising biometric authentication method for resolving this issue. This paper proposes a new on-line signature verification algorithm that utilizes only pen position trajectories. The algorithm is an improvement in our previous work [9]. In preliminary experiments, the equal error rate (EER) was 1.26%, which outperforms our previous result by about 0.3%.

Key-Words: - Biometrics, On-line Signature Verification, HMM, Pen position trajectories, Writer Authentication, Handwriting

1 Introduction

Personal identity verification has many applications, including electrical commerce, access to computer terminals and buildings, and credit card verification. Algorithms for personal authentication can be roughly classified into four categories depending on whether they are static or dynamic, biometric or physical, or knowledge-based, as illustrated in Fig. 1.

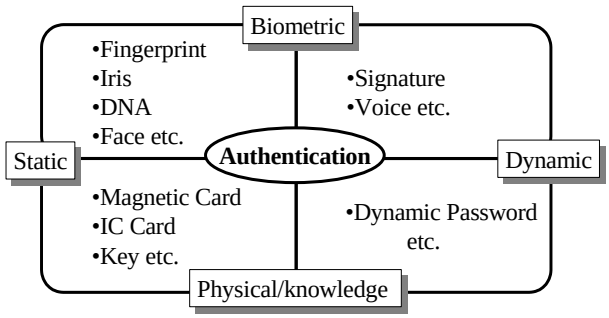

Fig.1 Authentication methods

Methods that utilize the characteristic of fingerprints, irises, DNA, or faces are considered static and biometric. Dynamic biometric methods include voice and on-line signatures. Schemes that use passwords are static and knowledge-based, whereas methods using magnetic cards and IC cards are static and physical.

There are at least two reasons that on-line pen-input signature verification is a promising scheme for personal authentication. First, signatures have a long history and are already built in among many countries. Second, the pen input environment has rapidly become a popular platform with the advent of pen-input devices such as PDAs and tablet PCs.

On-line signature verification has been studied for more than twenty years. Earlier studies are well summarized in [7, 11]. The on-line signature verification systems attempts to determine whether the input signature is a genuine signature (a signature written by the registered person) or forged signature (a signature written by an imposter) by using information derived from the on-line signature features. Thus the on-line signature verification problem is a two-class classification problem that is difficult for two main reasons related to the learning data. First, only a few data sets are available for learning. Only three to five data sets are generally available for on-line signature verification. Second, only the data sets belonging to one class are available. Data sets from both classes are generally available for a two-class classification problem. However signature verification systems must operate using data sets from only one class (genuine signatures) for learning since the systems do not have any advance information related to the forgery data that will be input by an imposter. This makes the on-line signature verification problem more difficult. Some algorithms that use forgery signature data sets for learning were recently proposed [6, 10]. These

algorithms perform well if good forgery data sets are provided, but there is a problem with obtaining good forgery data sets. We propose in this paper an algorithm for on-line signature verification using a discrete Hidden Markov Model (HMM) that incorporates only pen position trajectories with no forgery data sets are required for learning. We use only the pen position trajectories because we can obtain them from almost all pen-input devices. Therefore, our algorithm can be applied to all pen-input devices. A preliminary experiment was performed with a database consisting of 1848 genuine signatures and 3170 *skilled* forgery from fourteen individuals. There are four types of forgery: (A) Random forgery, in which the imposter has no access to the genuine signature, (B) Simple forgery, in which imposter knows the name of the person whose signature is to be authenticated, (C) Simulated forgery, in which imposter has a genuine off-line signature and can trace it, (D) Skilled forgery, in which imposter can view and train on the genuine signature. Type (D) forgery is a difficult situation to address. The preliminary experiment was performed using type (D) data for the forger. The experiment result indicates that the EER was 1.26%.

This paper is organized as follows: Section 2 describes the algorithm of on-line signature verification, and Section 3 presents the experiment results.

2 The Algorithm

The overall algorithm is depicted in Fig.2. It consists of two stages, the learning stage and verification stage. There are five sub-algorithms:

1) Preprocessing
2) HMM Generation
3) Performance Index Calculation
4) Threshold, and
5) Verification.

2.1 Preprocessing

There are three preprocessing steps in our algorithm: (i) Coordinate transformation, (ii) Quantization, and (iii) Data smoothing.

2.1.1 Coordinate transformation

Typical raw data (shown in Fig. 3) taken from the digitizer is as follows:

$$\big(x(t), y(t), p(t), \phi(t), \varphi(t)\big) \in R^5,\ t = 1, 2, \cdots, T \qquad (1)$$

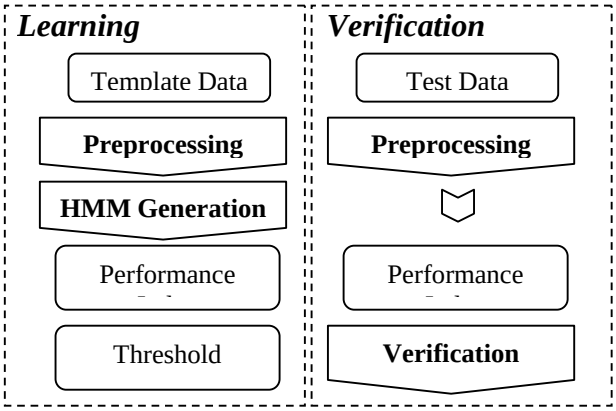

Fig. 2 Overall algorithm

Fig. 3 Typical raw data

Basically there are five features: pen position $x(t), y(t)$, pen pressure $p(t)$, pen azimuth $\phi(t)$, and pen altitude $\varphi(t)$, but the features that are used for the algorithms are differ among algorithms. The algorithms reported in [2, 3, 6, 10, 13] use all of the features, [16] uses pen position and pen pressure features, and [1, 4, and 15] use pen position and pen up/down (binary pen pressure) features. The algorithm in [9] uses only pen position features, and the algorithm in [8] uses the features that are derived from specific devices. Naturally the algorithms that use more features perform better. However the availability of these features depends on the device. Only pen position $x(t), y(t)$ and binary pen pressure features are available from many pen-input devices.

We use only the following pen position features in this paper.

$$\big(x(t), y(t)\big) \in R^2,\ t = 1, 2, \cdots, T \qquad (2)$$

We make the coordinate transformation as follows:

$$\theta = \tan^{-1} \frac{y(t+1) - y(t)}{x(t+1) - x(t)}, -\frac{\pi}{2} < \theta \le \frac{\pi}{2} \qquad (3)$$

and define θ' by

$$\theta' = \begin{cases} \theta & \big(x(t+1) - x(t) \ge 0\big) \\ \theta + \pi & \big(x(t+1) - x(t) < 0\big) \end{cases} \qquad (4)$$

2.2.2 Quantization

We considered $V(t)$, which represents the quantized angle information defined by the following equation to formulate the problem in terms of the discrete HMM.

$$V(t)=\begin{cases}1 & \left(\theta'<-\frac{\pi}{2}+\frac{\pi}{L},\theta'\geq\frac{3\pi}{2}-\frac{\pi}{L}\right)\\ n & \left(-\frac{\pi}{2}+\frac{(2n-3)\pi}{L}\leq\theta'\leq-\frac{\pi}{2}+\frac{(2n-1)\pi}{L}\right)\\ & n=1,2,\cdots L\end{cases} \quad (5)$$

Pen position $(x(t), y(t))$ is transformed into the L discretized angles in Fig. 4. Thus that data (2) is now a sequence of the discrete symbols:

$$O(t)=V(t)\in\{1,2,\cdots,L\} \quad (6)$$

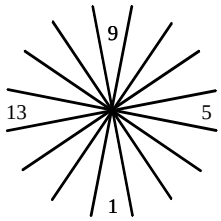

Fig. 4: Quantized directions (L=16)

2.1.3. Data smoothing

We transform the data after quantization by

$$O^{new}(t)=\alpha_{wp}O^{old}(t)+(1-\alpha_{wp}-\alpha_{wa})O^{old}(t)+\alpha_{wa}O^{old}(t) \quad (7)$$

α_{wp},α_{wa} : *parameters*

2.2. HMM Generation

HMM is a general doubly stochastic structure that is applicable to a broad class of problems in which time evolution is important [12, 14]. Every general discipline must be tailored before being applied to a specific type of problem, which is a basic engineering function, HMM is no exception. The general framework of HMM must be carefully tuned to the on-line signature verification problem. An HMM is formally described by the following:

- N : The number of states
  $q_1, q_2, \cdots, q_N$
- $\{a_{ij}\}$: State transition probabilities where
  $a_{ij}=P(Q(t+1)=q_j \mid Q(t)=q_i)$
- $\{b_{jk}\}$: Output emission probability where
  $b_{jk}=P(O(t)=k \mid Q(t)=q_j)$
- $\{\pi_i\}$: Initial state probabilities where
  $\pi_i=P(Q(1)=q_i)$

Learning in HMM amounts to an estimation of parameters $\{a_{ij}\}$, $\{b_{jk}\}$, $\{\pi_i\}$, and N, whereas verification computes the likelihood given a test signature and template HMM and attempts to make a determination. We will use the *left to right model* to tune HMM to our present problem.

2.2.1 State clarification

The number of the states in an HMM application is one of the most difficult parameters to estimate [12]. Therefore, the number of states in [15] was changed from two to nine, and [10] fixed it as four. We first attempted to associate a clear meaning to the states with the data sets for leaning given in our algorithm. We make a division between $O(t)$ and $O(t+1)$, if $V(t)\neq V(t+1)$ in given data sets, i.e., if the angle values change. Doing this enable us to divide the data into N groups. We then assume the groups as states and define N as the number of states. Each state consists of $n(q_i)$ symbols that have the same V_{q_i}. However, the algorithms is an HMM since it has nontrivial $\{b_{jk}\}$

2.2.2 Learning $\{a_{ij}\},\{\pi_i\}$

Some algorithms have been proposed to estimate the parameters. The Baum-Welch algorithm is a well-known method that estimates the HMM parameters $\{a_{ij}\}$, $\{b_{jk}\}$ and $\{\pi_i\}$ with a fixed N [10, 15]. The algorithm in [5] uses a segmental k-means iterative procedure and the algorithm in [2] applies Viterbi approximation. We use a simpler and faster algorithm to estimate the parameters, by defining

$$a_{ij}=0 \quad (i\neq j, i\neq j-1) \quad (8)$$

$$a_{ii}=\frac{n(q_i)-1}{n(q_i)} \quad (1\leq i\leq N-1) \quad (9)$$

$$a_{ii+1}=\frac{1}{n(q_i)} \quad (1\leq i\leq N-1) \quad (10)$$

$$a_{NN}=1 \quad (11)$$

Smoothing is performed to avoid overfitting.

$$a_{ii}^{new} = \alpha_a a_{ii}^{old} + (1-\alpha_a) a_{ii+1}^{old}$$
$$a_{ii+1}^{new} = (1-\alpha_a) a_{ii}^{old} + \alpha_a a_{ii+1}^{old} \quad (12)$$

The initial state probability is set as

$$\pi = \{1, 0, \cdots, 0\} \quad (13)$$

2.2.3 Learning $\{b_{jk}\}$

Each signature data in our algorithm generates one HMM. Again, care must be exercised to learn output emission probabilities in order to circumvent overfitting.

$$b_{jk} := \frac{\alpha}{Z(\sigma_v)} \int_{x=k-0.5}^{x=k+0.5} e^{\left(-\frac{(V(q_j)-x)^2}{2\sigma_v^2}\right)} dx + \frac{1-\alpha}{L} \quad (14)$$

Here. $Z(\sigma_v)$ is the normalization constant.

2.3 Performance Index Calculation

The purpose of signature verification is to determine whether a given test signature was written by the registered person, using learning data sets obtained in advance.

The given learning data sets

$$D := \{D_1, D_2, \cdots, D_m, \cdots D_M\}, D_m = \{O(t)\}_{t=1}^{T_m - 1} \quad (15)$$

consist of M signature trajectories from a registered person. We generate the associated HMMs using the above algorithm. Note that each learning data D_m in the above HMM generation generates one HMM H_m. Thus, we can have m pieces of HMMs.

We interested in the model likelihood when the test signature is given:

$$P(D_{test} \mid H_m) = \sum_{\substack{all\ possible\ paths \\ \{Q(t)\}_{t=1}^{T_{Test}-1}}} P\left(\{O(t), Q(t)\}_{t=1}^{T_{test}-1} \mid H_m\right)$$
$$= \sum_{\substack{all\ possible\ paths \\ \{Q(t)\}_{t=1}^{T_{Test}-1}}} \pi_{Q(1)} \prod_{t=1}^{T-2} a_{Q(t)Q(t+1)} \prod_{t=0}^{T-2} b_{Q(t+1)O(t+1)} \quad (16)$$

where $Q(t)$ represents the hidden state at time t.

This paper proposes performance index derived from

$$\Theta_{test\,m} = \frac{\ln P(D_{test} \mid H_m)}{T_{test} - 1} \quad (17)$$

2.4 Threshold

Each model in our algorithm has a threshold value. The threshold value of the m-th model is calculated in the following manner:

$$\lambda_m := \frac{c1}{M} \sum_{n=1}^{M} \Theta_{n,m} - c2 \sqrt{\frac{1}{M} \sum \left(\Theta_{n,m} - \frac{1}{M} \sum_{l=1}^{M} \Theta_{l,m} \right)^2}$$
$$m = 1, 2, \cdots, M \quad (18)$$

where Θ_{nm} is the performance index of the n-th registered signature in place of the test signature calculated in (17).

2.5 Verification

We calculate the performance indices by (17) when the test signature is given and compare them with threshold values. We then determine whether the test signature was written by the registered person as follows:

D_{test} is a genuine signature if $\sum_m f(\Theta_{test,m}, \lambda_m) \geq G$

D_{test} is a forged signature if $\sum_m f(\Theta_{test,m}, \lambda_m) < G$

where G is the empirical value and

$$f(\Theta_{test,m}, \lambda_m) = \begin{cases} 1 & (\Theta_{test,m} \geq \lambda_m) \\ 0 & (\Theta_{test,m} < \lambda_m) \end{cases} \quad (19)$$

3 Experiment

This section reports our preliminary experiment using the algorithm described above. Fourteen individuals participated in the experiment. The data was obtained for a period of three months. There were 1848 genuine signatures and 3170 *skilled* forgery. Table 1 lists the details of our database. Forgery data was not used for HMM learning in this experiment, it was used for the test only.

Figure 6 reveals the total average verification error as a function of parameter $c2$ described above, in which the EER is 1.26%. The EER will naturally improve if $c1$ and $c2$ are adjusted for each individual.

Many algorithms have been proposed and their experimental results reported. However, each algorithm is evaluated using a private database since there is no public database. Exact comparisons with other algorithms are therefore difficult. The results of this proposed algorithm outperforms our previous work [9], conducted using the same database.

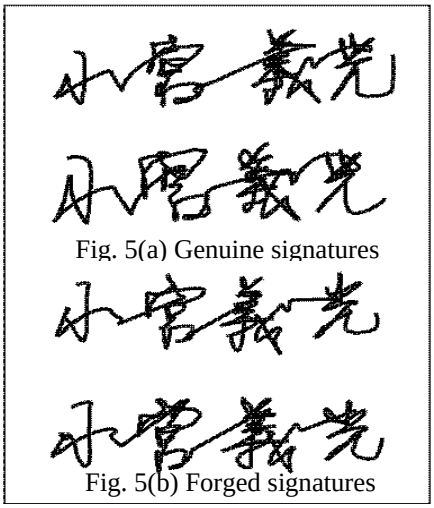

Fig. 5(a) Genuine signatures

Fig. 5(b) Forged signatures

Fig. 5 Test data for the experiment

Table 1 Data used for the experiment

Individuals	Genuine		Forgery	Total
	Test	Template generation	Test	
A	204	25	585	814
B	45	5	81	131
C	141	15	237	393
D	25	5	68	98
E	187	25	435	647
F	153	15	357	525
G	56	5	71	132
H	54	5	73	132
I	205	10	288	503
J	210	20	396	626
K	73	5	69	147
L	102	10	156	268
M	94	5	81	180
N	134	15	273	422
Total	1683	165	3170	5018

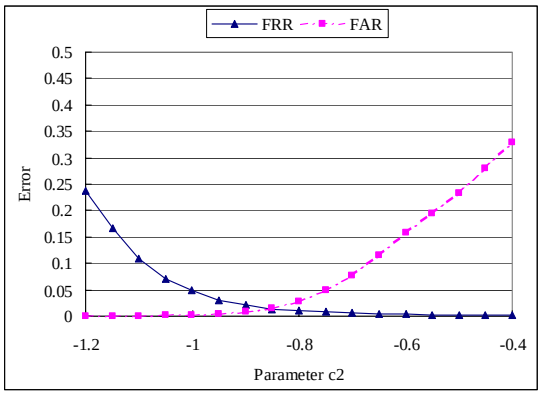

Fig. 6 Total average verification error

References:

[1] L. Bovino, S. Impedovo, G. Pirlo and L. Sarcinella, “Multi-Expert Verification of Hand-Written Signatures”, Proc. ICDAR 2003, Vol. 2, pp. 932-941, 2003.

[2] J. G. A. Dolfing, E. H. L. Aarts and J. J. G. M. Van Oosterhout, ”On-line Signature Verification with Hidden Markov Models”, Proc. ICPR 1998, Vol. 2, pp. 1309-1312, 1998.

[3] S. Hangai, S. Yamanaka and T. Hashimoto, “On-Line Signature Verification Based on Altitude and Direction of Pen Movement”, Proc. ICME 2000, pp. 319-322, 2000.

[4] A. K. Jain, F.D. Griess, and S.D. Connell, "On-line Signature Verification", Pattern Recognition, Vol.35, pp.2963-2972, 2002.

[5] R. S. Kashi, J. Hu, W. L. Nelson and W. Turin, “On-line Handwritten Signature Verification using Hidden Markov Model Features”, Proc. ICDAR 1997, vol. 1, pp. 253-257, 1997.

[6] M. Kondo, D. Muramatsu, M. Sasaki and T. Matsumoto, "A Bayesian MCMC Algorithm for On-line Signature Verification", Proc. Practical Bayesian Statistics 5, July 2003.

[7] F. Leclerc and R. Plamondon, “Signature Verification – The State of the Art 1989-1993”, IJPRAI, Vol. 8, no. 3, pp.643-660, 1994.

[8] P. Mautner, O. Rohlik, V. Matousek and J. Kempf, "Signature Verification Using ART-2 Neural Network", Proc. ICONIP 2002.

[9] D. Muramatsu and T. Matsumoto, "An HMM On-Line Signature Verification with Pen Position Trajectories", Proc. IC-AI 2003, Vol. 1, pp. 299-303, June 2003.

[10] J. Ortega-Garusia, j. Fierrez-Aguilar, J. Martin-Rello and J. Gonzalez-Rodriguez, "Complete Signal Modeling and Score Normalization for Function-Based Dynamic Signature Verification", Proc. AVBPA 2003, pp.658 -667, June 2003.

[11] R. Plamondon, G. Lorette, “Automatic Signature Verification and Writer Identification – The State of Art”, Pattern Recognition, Vol.22, no. 2, pp. 107-131, 1989.

[12] L. R. Rabiner, ”A Tutorial on Hidden Markov Models and Selected applications in speech Recognition”, Proc. IEEE, Vol. 7, No. 2, February, 1989.

[13] D. Sakamoto, T. Ohishi, Y. Komiya, H. Morita and T. Matsumoto, "On-line Signature Verification Algorithm Incorporating Pen

Position, Pen Pressure and Pen Inclination Trajectories", Proc. IEEE ICASSP 2001, Vol. 2, pp. 993-996, 2001.

[14] H. Yasuda, T. Takahashi and T. Matsumoto, "A Discrete HMM For Online Handwriting Recognition", IJPRAI, Vol. 14, No. 5, pp. 675-688, 2000.

[15] L. Yang, B. K. Widjaja and R. Prasad, "Application of Hidden Markov Models for Signature Verification", Pattern Recognition, Vol. 28, No. 2, pp. 161-170, 1995.

[16] M. Yoshimura, Y. Kato, S. Matsuda and I. Yoshimura, "On-line Signature Verification Incorporating the Direction of Pen Movement", Trans IEICE Japan, Vol. E74, No. 7, pp. 2083-2092, 1991.