Determination of Decision Boundaries for Online Signature Verification

Masahiro Tanaka[1], Yumi Ishino[2], Hironori Shimada[2], Takashi Inoue[2], and Andrzej Bargiela[3]

[1] Department of Information Science and Systems Engineering
Konan University, Kobe 658-8501, Japan
m_tanaka@konan-u.ac.jp
http://cj.is.konan-u.ac.jp/ ~tanaka-e/

[2] R & D Center, Glory Ltd., Himeji 670-8567, Japan
{y_ishino, shimada, inoue}@dev.glory.co.jp

[3] Department of Computing and Mathematics, The Nottingham Trent University
Burton St, Nottingham NG1 4BU, UK
andre@doc.ntu.ac.uk

Abstract. We are developing methods for online (or dynamical) signature verification. Our method is first to move the test signature to the sample signature so that the DP matching can be done well, and then compare the pen information along the matched points of the signatures. However, it is not easy to determine how to use the several elements of discrepancy between them. In this paper, we propose a verification method based on the discrimination by RBF network.

1 Introduction

We are developing methods for online signature verification. The data is multi-dimensional time series obtained by the tablet and an electronic pen ([1, 4, 3]). The information available in our system includes x and y coordinates, the pressure of the pen, the azimuth and the altitude of the pen. Hence the data can be seen as a sequence of vectors.

Our method is first to modify the test signature to the template signature so that the DP matching can be done well, and then compare the pen information along the matched points of the signatures.

One of the most intuitive methods for verification or recognition of hand-written characters or signatures is to extract the corresponding parts of two drawings and compare them next. For this objective, we use only the coordinates of the drawings and neglect other elements. We developed this method as a pre-processing tool before the main procedure of verification method [5–7].

However, it is not easy to determine how to use the several elements of differences between them. In this paper, we propose a verification method based on the discrimination by neural network.

2 Data Categories

Suppose q-dimensional time-series vector $\mathbf{p} \in \mathbf{R}^q$ is available from the tablet with a fixed interval, whose elements include x and y coordinates, pressure, azimuth and the altitude of the pen. We will use k as the discrete time index. For a signature verification problem, the following three kinds of data category should be prepared.

- **Template**. Template signature is the one registered by the true person. We could use more than two template signatures, but we use one template here for the simplicity of the treatment. Let $\mathbf{S} \in \mathbf{R}^{q \times n_s}$ represent it.
- **Training signatures**. We need both true and forgeries. It is possible to make a model with only genuine signatures and establish probability density function model, but the ability of discrimination is low. It is better to use both positive and negative cases. The more training signatures, the higher the accuracy. Let $\mathbf{T}_1$, $\mathbf{T}_2$, $\cdots$, $\mathbf{T}_m$ be the genuine signatures. Let $\mathbf{F}_{ij}, j = 0, \cdots, m(i); i = 1, \cdots, n$ be the forgeries, where i is the ID of the person and j is the file number.
- **Test signatures**. We also need both genuine and forgeries for checking the model of verification.

If we need to authenticate n signers, we need to prepare n-sets of the above.

3 Matching Method

By considering the diversity of genuine signatures as the partial drift of the speed during signing procedure, it is appropriate to apply DP matching. Here we will use only x and y which are elements of $\mathbf{p}$ for matching of two vectors. One sequence is the x, y elements of $\mathbf{S}$. We will denote it as $\bar{\mathbf{S}} = [\mathbf{p}(1), \cdots, \mathbf{p}(n_s)] \in \mathbf{R}^{2 \times n_s}$. Another sequence to compare with it is a 2-dimensional sequence which is denoted here as $\mathbf{Q} = [\mathbf{q}(1), \cdots, \mathbf{q}(n_q)]$.

For this problem, DP matching did not work sufficiently well. However, we considered this to be particularly important for verification. So, we proposed the matching method in [5], which is to apply DP matching and the coordinate modification iteratively.

Next we will explain the summary of DP matching and coordinate modification procedures.

3.1 DP Matching

DP matching used here is the method to derive the corresponding points that give the minimal cost for two vector sequences, and the following recursive form is well known:

$$D(i,j) = \min \left\{ \begin{array}{l} D(i-1,j) + d(i,j) \\ D(i-1,j-1) + 2d(i,j) \\ D(i,j-1) + d(i,j) \end{array} \right\} \tag{1}$$

where $d(i,j)$ is the distance between $\mathbf{p}(i)$ and $\mathbf{q}(j)$, which is usually Euclidean distance.

3.2 Coordinate Modification

This is a procedure for moving points of the test signature to the sample signature. The following is the model:

$$x'(k) = a(k)x(k) + b(k) \tag{2}$$

$$y'(k) = a(k)y(k) + c(k) \tag{3}$$

where k denotes the time index, $x(k)$ and $y(k)$ are the elements of $\mathbf{p}(k)$. $x'(k)$ and $y'(k)$ are the transformed values which are taken as the elements of $\mathbf{q}(k)$, and $a(k), b(k), c(k)$ are time-variant transformation parameters.

Now we define the parameter vector θ as

$$\theta(k) = [a(k)\ b(k)\ c(k)]'$$

If they are allowed to change slowly and independently, the model can be written as the following:

$$\theta(k+1) = \theta(k) + w(k) \tag{4}$$

where $w(k)$ is a zero-mean random vector with covariance matrix $Q = \mathrm{diag}(q_1, q_2, q_3)$. If the diagonal elements q_1, q_2, q_3 are small, it means the elements of $\theta(k)$ are allowed to change slowly.

Now we fix the template data $\mathbf{p}(k)$. Then it is expected that the test data transformed by (2)-(3) is close to the template data $\mathbf{p}(k)$. We can write this assertion by the following equation obtained from equations (2), (3)

$$z(k) = H(k)\theta(k) + v(k) \tag{5}$$

where

$$z(k) = \begin{bmatrix} x'(k) \\ y'(k) \end{bmatrix}$$

is the test data, and

$$H(k) = \begin{bmatrix} x(k)\ 1\ 0 \\ y(k)\ 0\ 1 \end{bmatrix} \tag{6}$$

includes the elements of the model data where $v(k)$ is a 2-dimensional random vector independent from w, its mean is zero and covariance R. This absorbs the unmodelled portion of the observation data.

Based on equations (4) and (5), we can estimate the time-variant parameter $\theta(k)$ by using the linear estimation theory. If we must estimate them on-line, we would use Kalman filter or fixed-lag smoother. However besides this, the data must be matched by using DP matching, where we must use the data in an off-line way. So, we must work in an off-line manner, and "fixed-interval smoother" yields the best result for off-line processing. Hence we will use fixed-interval smoother.

4 Verification Method

We have constructed the signature verification method based on the matching of signatures. Here we will describe the method in detail.

4.1 Normalisation of Training Data

We have experimentally experienced that data normalisation is very useful before matching. Let $\mathbf{p}$ be the original 2-D signal, and let R be the covariance matrix of $\mathbf{p}$. By the orthogonalisation, we have $RV = V\Lambda$, where V is the orthonormal matrix and Λ is the diagonal matrix. Since V is orthonormal, we have $V^{-1} = V'$ and $V'RV = \lambda$.

Now, let $\bar{\mathbf{p}}$ be the mean of $\mathbf{p}$. Also define $\bar{\mathbf{r}} = V'\bar{\mathbf{p}}$ and $\mathbf{r} = V'\mathbf{p} - \bar{\mathbf{r}}$. Then we have

$$E[\mathbf{r}\mathbf{r}'] = V'E[(\mathbf{p} - \bar{\mathbf{p}})(\mathbf{p} - \bar{\mathbf{p}})']V = \Lambda$$

If we want to have the data whose standard deviation of the horizontal axis is 1000, we can normalise it by

$$\mathbf{r} = \frac{1000}{\sqrt{\lambda_1}}(V'\mathbf{p} - \bar{\mathbf{r}})$$

thus we have

$$E[\mathbf{r}\mathbf{r}'] = \begin{pmatrix} 10^6 & 0 \\ 0 & \frac{\lambda_2}{\lambda_1} \times 10^6 \end{pmatrix}$$

4.2 Matching Data

Here we apply the iterative procedure of DP matching and coordinate modification of the data as described in section 3.

4.3 Extracting Feature Vectors

After the matching of two signatures $\mathbf{s}$ and $\mathbf{t}_i$, we can use the square of the distance between each elements as

$$\boldsymbol{d}(i) = [d_1(i) \;\; d_2(i) \;\; \cdots \;\; d_7(i)], \quad i = 1, \cdots, m \;\; \text{or} \;\; m(j) \tag{7}$$

The above criteria are common to the genuine and forgeries.

Each of the elements in (7) are the mean square difference between the template signature and the training signature. Each of the criteria have the following meaning.

Suppose (i, j) is the index after matching for one of the vectors. Then we trace back the original position (i', j') and calculate the velocity by

$$\begin{aligned} vel = & \sqrt{(x(i'-1) - x(i'))^2 + (y(j'-1) - y(j'))^2} \\ & + \sqrt{(x(i') - x(i'+1))^2 + (y(j') - y(j'+1))^2} \end{aligned} \tag{8}$$

Table 1. Elements of Criteria

element number	meaning
1	length of the data
2	modified coordinate x after matching
3	modified coordinate y after matching
4	pressure of matched points
5	angle of matched points
6	direction of matched points
7	velocity of matched points

This trace back is necessary because the matched index does not necessarily proceed smoothly, and will not show the actual velocity of the signature. The meaning of other elements will be straightforward, so we will omit the detail explanation.

4.4 Model Building by RBF Network

Thus we have the verification model. A training vector $\boldsymbol{d}(i)$ is 7-dimensional, and the output $o(i)$ is 1 (for genuine) or 0 (for forgery). We have m cases for $o(i) = 1$ and $m(1) + \cdots + m(n)$ cases for $o(i) = 0$. However, the scale of the criteria vary a lot, thus we need further normalisation. Hence we normalised the criteria by

1. Find the maximum max_j among the j-th elements in the training vectors, and
2. Divide all the elements of j-th elements by max_j. This max_j is again used in the verification.

Let us denote the training data by $\boldsymbol{d}(k), k = 1, \cdots, N$. By using all the samples as the kernel of the function, we have the model

$$f(\boldsymbol{x}) = \sum_{k=1}^{N} \theta_k \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_k\|^2}{\sigma^2}\right) \tag{9}$$

for a general $\boldsymbol{x} \in \mathbf{R}^7$. This can be rewritten as

$$Y = M\theta \tag{10}$$

where $Y \in \mathbf{R}^N$, $M \in \mathbf{R}^{N \times N}$, and $\boldsymbol{\theta} \in \mathbf{R}^N$. By using the training data $\mathbf{d}_k, k = 1, \cdots, N$, we have the $n \times N$ matrix M whose rank is N. So, we have a unique parameter $\boldsymbol{\theta}$. It is clear that equation (9) produces the exact values for the training data $\mathbf{d}_k, k = 1, \cdots, N$.

Various kinds of neural networks can adapt to nonlinear complicated boundary problems. However, we applied RBF network for a special reason. It has a universal approximation property like multi-layer perceptron. However, RBF network has another good property for pattern recognition like this. Due to the functional form, it intrinsically produces nearly zero if the input vector is not similar to any of the training data. Other recogniser like multi-layer perceptron does not have this property.

4.5 Verification

We can use the model for verifying whether the data is genuine or forgery based on the model. When we have the data, we have to process it as follows.

1. Normalise the original data
2. Normalize the feature vector
3. Input the feature vector to the recogniser.

5 Experimental Results

Fig. 1 shows the ranges of each criterion. The sub-figures are numbered in the order from the top-left, top-right, and to the second row. Sub-figure 1 shows the values of $d_1(i), i = 1, \cdots, 9$. The same for the sub-figure 2 with values $d_2(i), i = 1, \cdots, 9$, and so on. The first var corresponds to the genuine signature.

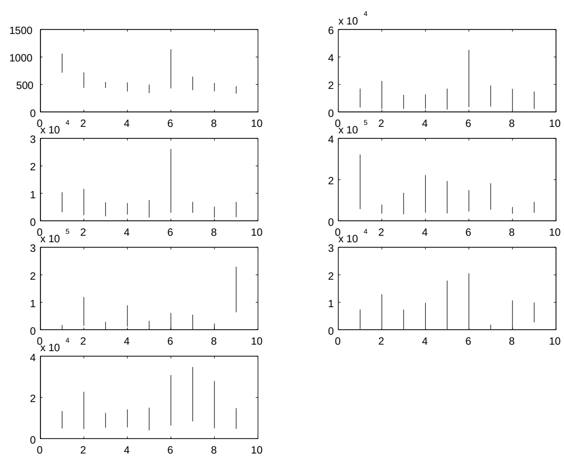

Fig. 1. Ranges of the criteria

By using each criterion alone, it is almost impossible to divide the space clearly. However, by using the RBF network, we have obtained a good separation result as shown in Fig. 2. The first 15 cases are the genuine signatures and all the others are forgeries.

If we put the threshold at 0.4, the errors of the first kind is 3 out of 15 and the errors of the second kind is 3 out of 135. This is fairly a good result.

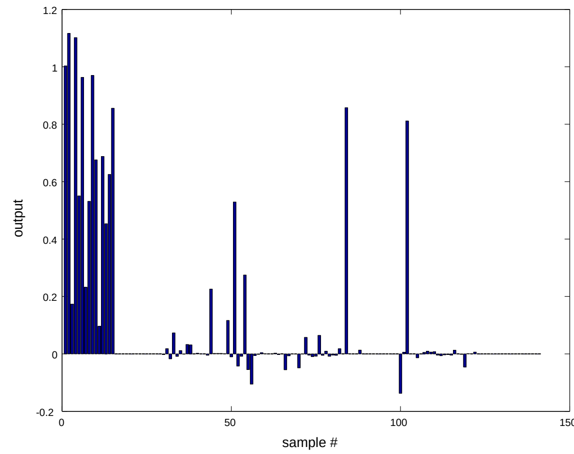

Fig. 2. Output of RBF network for test data

6 Conclusions

The performance of the algorithm will be upgraded if we apply the matching procedure more times. We will have to apply the algorithm for other persons' data as the template. Further comparison will be our future work.

References

1. L. L. Lee, T. Berger and E. Aviczer, "Reliable On-line Human Signature Verification Systems," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 18, pp. 643-647, 1996.
2. M. L. Minsky and S. A. Papert, *Perceptrons – Expanded Edition*, MIT, 1969.
3. V. S. Nalwa, "Automatic On-line Signature Verification," *Proceedings of the IEEE,* Vol. 85, pp. 215-239, 1997.
4. R. Plamondon and S. N. Srihari, "On-line and Off-line Handwriting Recognition: A Comprehensive Survey," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* Vol. 22, pp. 63-84, 2000.
5. M. Tanaka, Y. Ishino, H. Shimada and T. Inoue, "DP Matching Using Kalman Filter as Pre-processing in On-line Signature Verification," *8th International Workshop on Frontiers in Handwriting Recognition,*, pp. 502–507, Niagara-on-the-Lake, Ontario, Canada, August 6-8, 2002.
6. M. Tanaka, Y. Ishino, H. Shimada and T. Inoue, Dynamical Scaling in On-line Handwritten Characters' Matching, 9th International Conference on Neural Information Processing, Singapore, November 19-22, 5 pages (CD ROM), 2002.
7. M. Tanaka, Y. Ishino, H. Shimada, T. Inoue and A. Bargiela, "Analysis of Iterative Procedure of Matching Two drawings by DP Matching and Estimation of Time-Variant Transformation Parameters," The 34th International Symposium on Stochastic Systems Theory and Its Applications, accepted.