J. Basic. Appl. Sci. Res., 3(6)124-129, 2013
© 2013, TextRoad Publication
ISSN 2090-4304
Journal of Basic and Applied Scientific Research
www.textroad.com

Signature Verification Using Bayesian Model and MCMC Method

M. Behboudi[1*], E. Pasha[2], K. Shafie[3]

[1]Department of Statistics, North Tehran Branch, Islamic Azad University, Tehran, Iran
[2]Department of Mathematics and Computer Science, Kharazmi University, Karaj, Iran
[3]Department of Statistics, Shahid Beheshti University, Tehran, Iran

ABSTRACT

Bayesian model and Markov chain Monte Carlo method were used to verify a sample of signature in Iran. In the Bayesian model, parameters were considered as quantities whose variations can be described by the prior distributions. A sample is taken from a population indexed by the parameters and prior distributions were updated and then posterior distribution was calculated. Markov chain Monte Carlo method used to generate samples from the posterior distribution. The goals were (a) to define parameters of Bayesian model, obtain log-likelihood and propose the prior distribution and the posterior distribution (b) to introduce appropriate Markov chain Monte Carlo to explore the posterior (c) to propose a forgery index to determine correctness of this signature. In this method, off-line signature was representing via its curvature, and smoothness of the signature curve seems to be a necessity. In order to explore the variation in the signature samples, we used time warping functions as a random effect in the model. The performance of the proposed model was presented through a simulation example of an Iranian signature. For this signature sample, required times for preprocessing stages and signature verification stage were about 15 and 2 minutes and Type I and Type II error rate were 0.06 and 0.16 respectively. So this method seems to be appropriate to verify Iranian signatures.
KEY WORDS: biometric identification, time warping, curve registration, Strauss process, bootstrap.

1. INTRODUCTION

Off-line signatures verification is the most useful method for authentication of bank checks, questioned document examination, etc. Based on forger skills, a forgery is named random forgery, casual forgery, amateur forgery, or professional forgery. A random forgery is a random signature or a signature of other persons. A casual forgery is a kind of forgeries that forger does not access to samples of individual's actual signature. Amateur forgeries and professional forgeries are kinds of skilled forgeries in which forger access to one or more sample of individual's actual signature. A professional forgery is produced when the forger is a specialist in signature analysis. Finding appropriate methods for off-line signature verification is beloved by researchers. In this regard, several statistical methods are presented. These include the techniques based on wavelets [1], tracking of feature and stroke position [2], dynamic time warpings [3], shape matrices [4], functional data analysis [5], granulometric size distribution [6], hidden Markov models [7], Bayesian models and Markov chain Monte Carlo methods [8], etc.

Off-line signatures are often used in financial institutions and courts in Iran, and nowadays forgeries spread extensively. Also sufficient studies have not done on this subject in Iran, and now correctness of off-line signatures is determined through eye detection. So we are looking for a simple and reliable statistical method to detect Iranian forgeries especially professional Iranian forgeries. Signatures are the consequence of muscular contractions which occur because of the brain's commands that are sent to the muscles by motor neurons. The components of this process are different from one person to another and for one person in different situations. So when several people draw a signature, or an individual draws it in various situations, there are variations in the shape of those signature's samples. Thus, we are looking for a method that explores the variations in an individual's signatures. In the method [8] proposed, time warping functions were used to explore the variations in an individual's signatures. In this method signatures are represented via its curvature and it's based on Bayesian model and MCMC. Therefore we choose this method for signature verification, use it to verify an Iranian signature sample, and investigate whether this method is appropriate for Iranian signatures verification or not?

The paper is organized in 5 sections. The Bayesian model and the MCMC method for sampling from the posterior are introduced in the section 2. Signature verification is described in section 3. Analysis of an Iranian signature sample is given in section 4. In fifth section experimental result are presented in term of FRR, FAR, and the run-time of this method.

2. BAYESIAN MODEL

2.1 Log-likelihood

Suppose that the curve $\alpha_{obs}: [a, b] \to \mathbb{R}^2$ is arc-length parameterized and smooth, where $\alpha_{obs} = (x(t), y(t))$, $t \in [0, L(b)]$. $L(t)$ is the arc length and $L(t) = \int_a^t |\alpha'_{obs}|$, where $|.|$ is Euclidean norm. Also curvature of this curve is defined as follows:

$$\kappa(t) = x'y'' - x''y' \tag{1}$$

One of the most important properties of curvature is that it is invariant under translations and rotations. Furthermore it is proportional to scale inversely, the curve $c\alpha_{obs}(t)$ has the curvature $\kappa(t)/c$, for $c > 0$. Because we have some

*Corresponding Author: Maryam Behboudi, Department of Statistics, North Tehran Branch, Islamic Azad University, Tehran, Iran, Email: m.behboudi.statistics@gmail.com

signatures in this method, we re-scale each signature sample to have a same arc-length $\tau > 0$. In this case, each curvature function belongs to the space of continuous functions on $[0,\tau]$, $C[0,\tau]$.

At first, suppose that we have one signature sample. After preprocessing the image of the signature, we numerically compute its arc length and its velocity vector $V(t) = \alpha_{obs}^{'}(t) = (x^{'}(t), y^{'}(t))$.[9]

The observed velocity is modeled as follow:

$$V(t) = \begin{pmatrix}\cos\theta(t)\\ \sin\theta(t)\end{pmatrix}, \qquad \theta(t) = \varphi + \int_0^t \kappa(s)ds + \sigma W(t) \tag{2}$$

where $\kappa \in C[0,\tau]$ is an underlying curvature function, $W(t)$ is a standard Brownian motion and φ is the angle of $V(0)$ with x-axis. In this method, white noise with variance $\sigma^2 > 0$ is superimposed to the underlying curvature. Given the availability of the numerical values of the velocity vector $V(t)$, $t \in [0,\tau]$, over a regular grid $s_j = \frac{j\tau}{N}$, $j = 0, \dots, N$, log-likelihood is obtained as:

$$\ell(\kappa|V) = \frac{1}{\sigma^2}\int_0^\tau \kappa(s)\{V_1(s)dV_2(s) - V_2(s)dV_1(s)\} - \frac{1}{2\sigma^2}\int_0^\tau \kappa(s)^2\, ds + \frac{1}{8}\sigma^2\tau$$

where we select the appropriate value for σ through experimental studies.

Now, we develop the model to the case that we have n signature samples with observed velocity vectors $V^{(i)}$, $i = 1,2,\dots,n$.

Due to the time difference in the peaks of different curvatures, we use time warping function h_i: $[0,\tau] \to [0,\tau]$.

The full log-likelihood is defined as:

$$\ell(\kappa, \boldsymbol{h}|data) = \sum_{i=1}^n \ell(\kappa_i|V^{(i)})$$

where $\kappa_i(t) = \kappa(h_i(t))$, $\boldsymbol{h} = (h_i, i = 1,\dots,n)$, and κ is the baseline curvature function [8].

2.2 Prior distributions

The parameters of this model are κ, $\boldsymbol{h}$, and we have to define proper prior distributions for them.

2.2.1 Baseline curvature process

We choose one of the signature samples as the template sample, and its curvature as a template curvature function then denote it as κ_0. We define a buffer region as:

$$\mathcal{B} = \{(x,y): \kappa_L(t) < y < \kappa_U(t), t \in [0,\tau]\}$$

where $\kappa_L(t) = \kappa_0(t) - \varepsilon$ and $\kappa_U(t) = \kappa_0(t) + \varepsilon$ are the bracketing functions and $\varepsilon > 0$ is specified by the user.

We generate a limited number of random points in $\mathcal{B}$ via Strauss process X. X has unnormalized density $f(\boldsymbol{x}) = \beta^{n(x)}\gamma^{d(x)}$ with respect to the distribution of a unit rate Poisson process in $\mathcal{B}$, where $\beta > 0$, $0 \le \gamma \le 1$ are tuning parameters, $d(\boldsymbol{x})$ is the number of unordered pairs of points in $\boldsymbol{x}$ that their maximum metric distance is ρ , and $n(\boldsymbol{x})$ is the number of points in $\boldsymbol{x}$. It is worth noting that in this method, the metric is the distance between time coordinates [10].

We add two points $(t_0, y_0) = (0, \kappa_0(0))$ and $(t_{n(X)+1}, y_{n(X)+1}) = (\tau, \kappa_0(\tau))$ to the points (t_j, y_j) , $j = 1,\dots,n(X)$, which were generated via Strauss process X and $0 < t_1 < t_2 < \cdots < t_{n(X)} < \tau$. Then we define the baseline curvature process as:

$$\kappa(t) = \left(\frac{t_{j+1}-t}{t_{j+1}-t_j}\right)\{\kappa_0(t) + y_j - \kappa_0(t_j)\} + \left(\frac{t-t_j}{t_{j+1}-t_j}\right)\{\kappa_0(t) + y_{j+1} - \kappa_0(t_{j+1})\}$$

where $t_j \le t \le t_{j+1}$, $j = 0,1,\dots,n(X)$.

2.2.2 Time warping processes

Each time warping process is modeled as follow:

$$h_i(t) = \min\{\tilde{h}_i(t), \tau\}, \quad t \in [0,\tau]$$

where $\tilde{h}_i(0) = 0$ and $\tilde{h}_i(t)$ is a continuous, increasing, piecewise-linear function that be linear between the grid points $u_j = \frac{j\tau}{\wp}$, where $\wp$ is determined by the user and $j = 0,\dots,\wp$. The parameters $\theta_{ij} = \tilde{h}_i(u_j) - \tilde{h}_j(u_{j-1})$, $j = 0,\dots,\wp$, determine h_i and then we indicate $\boldsymbol{h}$ by combining them into a vector $\boldsymbol{\theta} \in \Theta = [0,\tau]^{n\wp}$. Also it's worth that $0 \le \theta_{ij} \le \tau$ [11]. Using the method of curve registration, we determine the prior density on h_i as:

$$\pi(h_i) \propto \exp\{-\eta J(h_i|\kappa)\}$$

where

$$J(h_i|\kappa) = \int_0^\tau angle\ \left(V^{(i)}(t), V_{fitted}^{(i)}(t)\right)dt$$

is the cost function where $V_{fitted}^{(i)}(t)$ is the velocity vector of curvature function $\kappa(h_i(t))$ and $V_{fitted}^{(i)}(0) = V^{(i)}(0)$. Also $\eta > 0$ is the precision parameter defined by the user [8].

2.3 Posterior distribution

Based on Bayes formula, the posterior density is proportional to:

$$g(\boldsymbol{x}, \boldsymbol{\theta}) = \exp\{\ell(\kappa, \boldsymbol{h}|data)\} f(\boldsymbol{x}) \prod_{i=1}^n \pi(h_i|\kappa, data)$$

where $\boldsymbol{x}$ are the knots of curvature κ , and $\boldsymbol{\theta}$ are the parameters that represent the vector $\boldsymbol{h}$. Also dominating measure is the product of the unit-rate Poisson distribution on $\mathcal{B}$ and Lebesgue measure on Θ.

The MCMC scheme used in this model is "Metropolis-within-Gibbs". In order to explore this posterior, each of the components $(\boldsymbol{x}, \boldsymbol{\theta})$ is updated separately by Metropolis-Hastings algorithm. The acceptance probability for each proposal is $\min(\alpha, 1)$, where α is the Hastings ratio [12].

Each component θ of $\boldsymbol{\theta}$ is updated using a random walk Metropolis move. The proposal θ' has normal distribution with mean θ and it is restricted to the compact interval $[0, \tau]$. Here, the move is formed as $(\boldsymbol{x}, \boldsymbol{\theta}) \rightarrow (\boldsymbol{x}, \boldsymbol{\theta}')$ and its Hastings ratio is $\alpha_{rw}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\theta}') = \frac{g(x,\theta')}{g(x,\theta)}$.

The x-component is updated by Metropolis-Hastings algorithm introduced by [13]. Either a point such as ξ is born uniformly random in $\mathcal{B}$ and adds to $\boldsymbol{x}$ that in this situation the move is formed as $(\boldsymbol{x}, \boldsymbol{\theta}) \rightarrow (\boldsymbol{x} \cup \{\xi\}, \boldsymbol{\theta})$ and its Hastings ratio given by:

$$\alpha_b(\boldsymbol{x}, \boldsymbol{\theta}, \xi) = \frac{|\mathcal{B}|}{n(x)+1} \frac{g(x \cup \{\xi\}, \boldsymbol{\theta})}{g(x, \boldsymbol{\theta})},$$

or a point in $\boldsymbol{x}$ is selected randomly and killed (unless $\boldsymbol{x}$ is empty) that in this case the move is formed as $(\boldsymbol{x}, \boldsymbol{\theta}) \rightarrow (\boldsymbol{x} \backslash \{\xi\}, \boldsymbol{\theta})$ and its Hastings ratio given by:

$$\alpha_d(\boldsymbol{x}, \boldsymbol{\theta}, \xi) = \frac{n(x)}{|\mathcal{B}|} \frac{g(x \backslash \{\xi\}, \boldsymbol{\theta})}{g(x, \boldsymbol{\theta})}$$

where $|\mathcal{B}|$ is the area of the buffer region.

3. Signature verification

The main goal of this research is signature verification. The signature verification problem determines whether a particular signature is written by the same person or not.

At first, we re-scale the new signature to have the same arc-length. We denote its velocity vector as V^{new}. We determine the forgery index as:

$$F = \frac{1}{\pi\tau} min_{i=1,\dots,n} \int_0^\tau angle\ \left(V_i^{new}(t), V^{(i)}(t)\right) dt$$

Using this forgery index can be verified the correctness of new signatures, where V_i^{new} is the velocity vector of new signature which has been rotated (clockwise) to have the initial direction $V^{(i)}(0)$, $i = 1, \dots, n$. We normalize F to range between 0 and 1. In fact, we have hypothesis testing in which null hypothesis is that the new signature is genuine and F is the test statistic.

We generateV^* by using a bootstrap method and replacing κ by $\hat{\kappa}(\hat{h}_I(.))$ in Formula 2, where I is randomly selected from $= 1, \dots, n$, and $\hat{\kappa}$, $\hat{h}_i$ are the posterior means of the baseline curvature, and the time warping h_i , respectively. Then we define F^* by replacing V^{new} by V^* in F . Next, by assessing p-value $P(F^* > F)$, we verify the correctness of new signature [8].

4. An iranian signature verification

In this section, we verify the correctness of an Iranian signature sample by using the method proposed in section 2 and Matlab software. We have 4 signature samples. At first, we scan images of signatures.

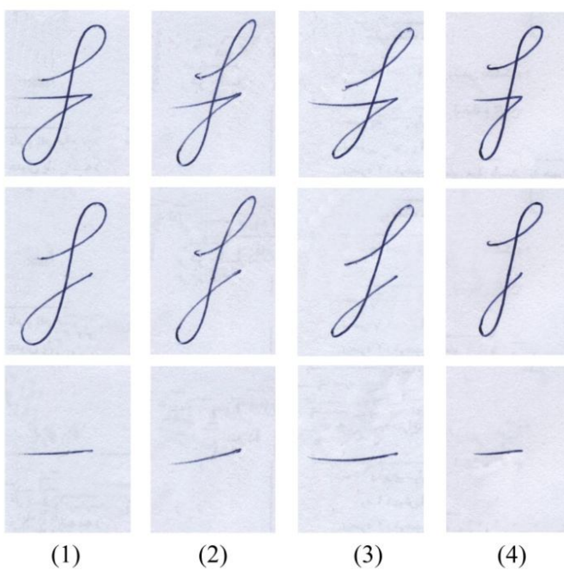

Figure 1. Images of 4 signature samples (first row); the first segment of the signatures (second row); the second segment of the signatures (third row)

As you see in Figure (1), each signature has one singular point. Because there is no tangent in a singular point, in each signature we separate 2 segments of the signature from the singular point. Also in each signature, the second segment is a straight line so its curvature is equals to 0. Thus we just to verify the first segment.

At first, the images are loaded into the Matlab software with the command "imread". By clicking on signature points located at regular distances from each other along signature curve, we record the coordinates of these points. Then cubic spline interpolation over a regular grid of 1001 points and separately to x- and y-components is used to get smooth and accurate approximation for the signatures. Next, we select 100 points from these 1001 points by regular rule, and by detecting the componentwise derivatives in these 100 points (with the command "gradient") we compute the velocity vector and the arc length of each signature. The arc lengths of signatures are 2.6970, 2.5136, 2.7398, and 2.3629, respectively. Because the arc lengths of the signatures are different, we choose signature (2) as the template sample and re-scale other signatures to have the same arc length $\tau = 2.5136$. Figure (2) shows the velocity vector of signature samples after re-scaling.

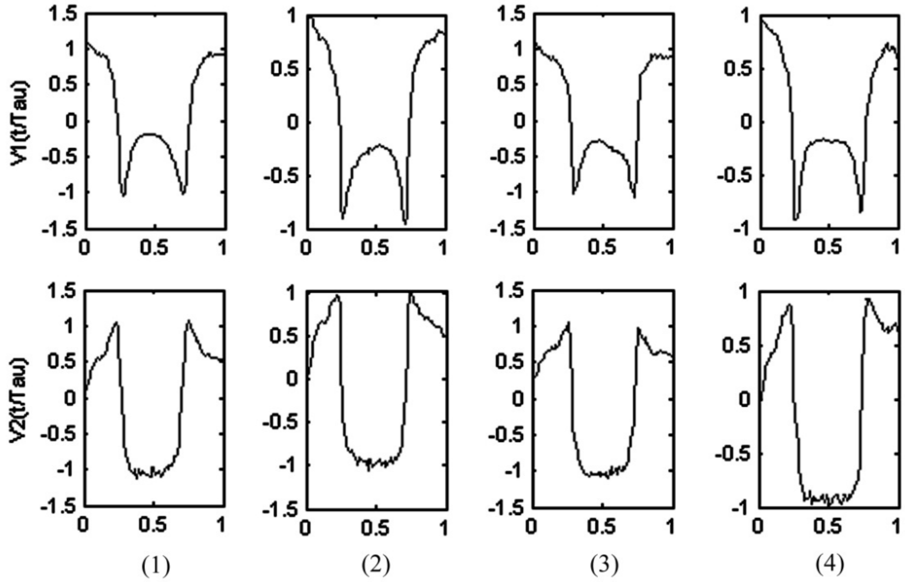

Figure 2. The first and second components of the velocity vector for 4 signature samples

We choose the curvature function of the signature (2) as the template curvature function κ_0 . $\varepsilon = 1.5$, $\rho = 0.005\tau$, $\beta = 6$, and $\gamma = 0.04$ are the proper values for the parameters of the baseline curvature process.

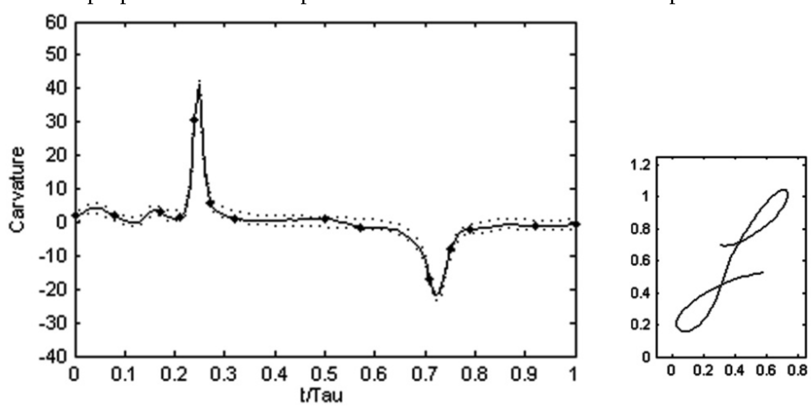

Figure 3. Bracketing functions (dotted line), generated points using Strauss process, and approximated curvature (solid line)(left); obtained trace from approximated curvature (right)

As you see in Figure (3), the observed trace from approximated curvature is very similar to the signature samples. The MCMC scheme needs to carefully accuracy so $\varepsilon = 0.2$, $\rho = 0.0001\tau$, $\beta = 6$, $\gamma = 0.01$, $\wp = 20$, $burn = 500$, and $afterburn = 1000$ are the best value for the parameters of the MCMC scheme.

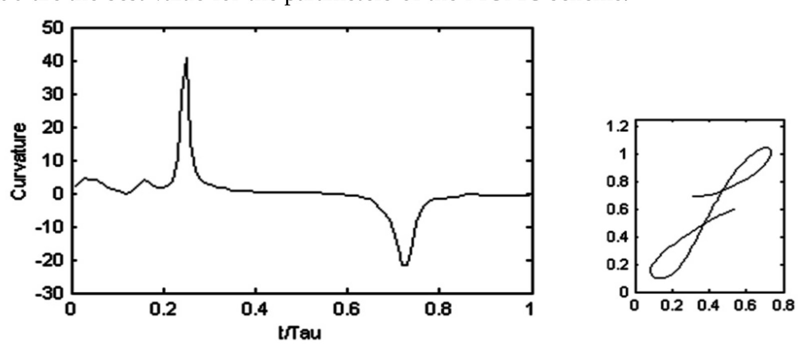

Figure 4. Posterior mean baseline curvature (left); obtained trace from posterior mean baseline curvature (right)

Figure (4) shows the posterior mean baseline curvature regardless of the effects of time warping functions. As you see, obtained trace from posterior mean baseline curvature is very similar to the signature samples.

Also the acceptance rate of the MCMC scheme to accept θ' , born a point in x , and kill a point in x are %2.6, %11.5, and %11, respectively.

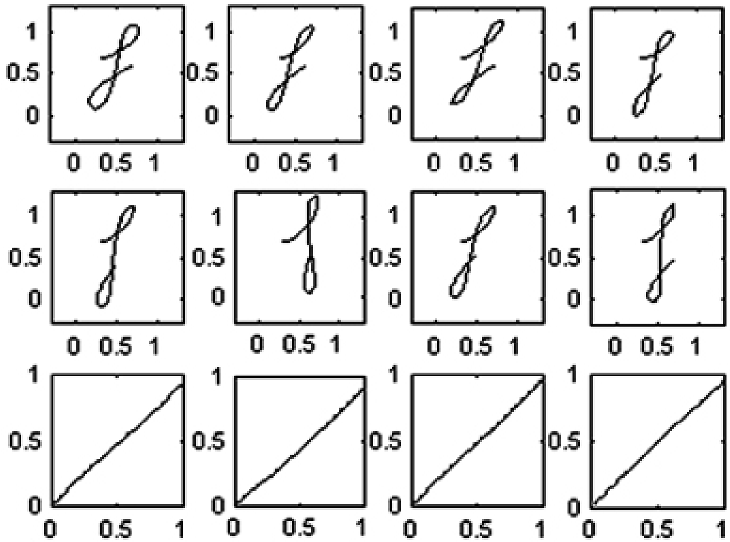

Figure 5. Traces of signature samples (first row); trace derived from the posterior mean baseline curvature adjusted by each posterior mean time warping (second row); posterior mean time warping functions (third row)

As we can see in Figure (5), trace derived from the posterior mean baseline curvature give acceptable images, and it means that this MCMC scheme forecast signature images. Now we verify the correctness of a new signature sample. This new signature sample can be seen in Figure (6).

Figure 6. New signature sample image (left); the first segment of this signature (right)

As mentioned, we just need to verify the first segment of the signature. At first we re-scale this signature to have the same arc length $\tau = 2.5136$. Its velocity vector and curvature function are shown in Figure (7) and Figure (8), respectively.

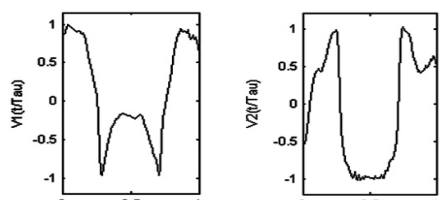

Figure 7. The first and second components of the velocity vector for the new signature

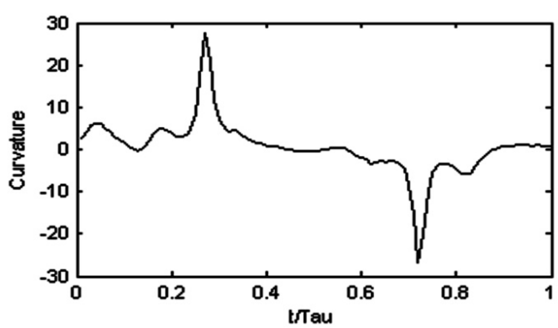

Figure 8. The curvature function of the new signature

The forgery index F is equal to 0.0762 for this signature. Also after simulating the values of F^* using bootstrap, the amount of p-value $P(F^* > F)$ is equal to 0.0004. Therefore, the null hypothesis is rejected and the new signature is a forgery.

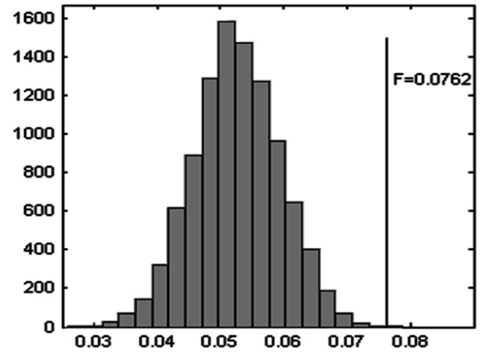

Figure 9. Histogram of bootstrap simulations of F^* and forgery index F

5. DISCUSSION AND CONCLUSION

We computed Type I and Type II error rate to evaluate the validity of this method for this individual's signature. The false reject rate (FRR) also called Type I error is the ratio of the number of genuine test signatures rejected to the total number of genuine test signatures submitted, and false accepted rate (FAR) also called Type II error is the ratio of all numbers of forgeries accepted to the total number of forgeries submitted [14].

To compute Type I and Type II error rate, at first we got 30 genuine signatures from the individual and 30 forgeries which included 4 types of forgeries, and then computed FRR and FAR. At the significance rate 0.025, 2genuine signatures in 30 genuine signatures were rejected, and 5 forgeries in 30 forgeries were accepted. So the FRR and FAR were 0.06 and 0.16, respectively.

Also required runtime for programs of preprocessing stages was less than 15 minutes and required runtime for the signature verification program was less than 2 minutes.

According to the obtained FRR and FAR, this method was proper to verify this individual's signature. Also because the programs of preprocessing for each person's signature run only once, these runtimes were good. To evaluate the validity of this method for all Iranian signatures, we must compute Type I and Type II error rate for numerous signatures included many types of Iranian signatures. Also some Iranian signatures have one or more singular point. Our goal is to verify this kind of signatures using this method and evaluate the velocity of this method for these signatures.

It is worth noting that these run-times were similar to the run-times of Shakespeare's signature verification so this method run-time is relatively stable. Also in earlier study, Type I and Type II error rates were not calculated and we calculate these errors for the first time.

REFERENCES

1. Deng, P. S., Liao, H.-Y.M., Ho, C.W., and Tyan, H.-R., 1999. Wavelet-based off-line signature verification. Computer Vision and Image Understanding., 76: 173–190.

2. Fang, B., Leung, C. H., Tang, Y. Y., Tse, K. W., Kwok, P. C. K., and Wong, Y. K., 2003. Off-line signature verification by the tracking of feature and stroke positions. Pattern Recognition., 36: 91–101.

3. Piyush Shanker, A., Rajagopalan, A.N., 2007. Off-line signature verification using DTW. Pattern recognition letters., 28(12): 1407-1414.

4. Sabourin, R., Drouhard, J.-P., and Wah, E. S., 1997. Shape matrices as a mixed shape factor for offline signature verification. In Proceedings of the Fourth International Conference on Document Analysis and Recognition, pp: 661–665.

5. Ramsay, J. O., 2000. Functional components of variation in handwriting. Journal of the American Statistical Association., 95: 9–15.

6. Sabourin, R., Genest, G., and Prˆeteux, F., 1997. Off-line signature verification by local granulometric size distributions. IEEE Trans. On Pattern Analysis and Machine Intelligence., 19 (9): 976–988.

7. El-Yacoubi, A., Justino, E. J. R., Sabourin, R., and Bortolozzi, F., 2000. Off-line signature verification using HMMs and cross-validation. In IEEE International Workshop on NeuralNetworks for Signal Processing., 2: 859–868.

8. Mckeague, I., 2005. A statistical model for signature verification. American Statistical Association., 100: 231-241.

9. Carmo, M. P. do., 1976. Differential Geometry of Curves and Surfaces. Prentice–Hall New Jersey publishers, pp:1-30.

10. Van Lieshout, M. N. M., 1995. Markov point processes and their applications in high-level imaging. Bulletin of the international statistical institute., 56: 559-576.

11. Ramsay, J. O., and Silverman, B. W., 1997. Functional Data Analysis. Springer New York publishers, pp: 73-80.

12. NtZoufras, I., 2009. Bayesian Modeling Using WinBUGS. Wiley New Jersey publishers, pp: 31-76.

13. Geyer, C. J., and Møller, J., 1994. Simulation and likelihood inference for spatial point processes. Scandinavian Journal of Statistics 21: 359–373.

14. Munich, M. E., and Perona, P., 2003. Visual identification by signature tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence., 25: 200–217.