Synthetic Handwritten Gesture Generation Using Sigma-Lognormal Model for Evolving Handwriting Classifiers

Abdullah Almaksour, Eric Anquetil, Réjean Plamondon, Christian O'Reilly

Synthetic Handwritten Gesture Generation Using Sigma-Lognormal Model for Evolving Handwriting Classifiers

Abdullah Almaksour, Eric Anquetil
INSA de Rennes, FRANCE

Réjean Plamondon, Christian O'Reilly
Ecole Polytechnique de Montral, CANADA

Abstract

We show in this paper the importance of using handwriting generation in the context of online and incremental learning of a handwriting classifier. In order to obtain realistic synthetic gestures, we apply controlled deformations on the extracted sigma-lognormal parameters of the real gesture, and we then generate synthetic gestures using the modified parameters. Results show the impact of integrating these synthetic samples generation in our learning algorithm on the classification performance.

1 Introduction

Motivated by the increasing spread of many types of devices equipped by pen-based interfaces, such as PDAs, e-book, Tablet PCs, Whiteboards ... etc, more emphasis is placed on the development of efficient recognition systems that can correctly interpret the gestures sketched by the user and then translate them either into computerized text or into some specific commands. Nowadays, the used recognition systems are always pre-trained on a fixed, predefined and almost limited group of gestures, which usually contains the Latin letters and a few specific gestures. These systems do not allow users to add new gestures in order to assign them to new commands or shortcuts, or to replace default gestures assigned to existing commands. In order to meet this important functionality, the static handwriting recognition systems that had been used so far must be replaced by new dynamic ones where the knowledge base can endlessly evolve during the use of the system. The evolving nature comes from the fact that the system must be able to integrate in any moment a new unseen class (gesture in our context), and must also continue the learning of the existing classes using the new available learning data. The latter property is very important because learning a new class can never be perfect using the very few learning samples that the user is ready to provide. So the idea is to start the learning using this few data, and then when the user starts using the system, he will be asked to validate or not the recognition system's responses In this way, new learning data samples become available and the classifier must take advantage of them to enhance its performance.

Therefore, the recognition system must be learned in incremental, online and lifelong mode. In incremental learning, the system (the classifier) must keep learning from any new available data samples without neither restarting the learning process from zero nor requiring access to preceding data samples. We present in this paper a handwriting classification system that satisfies the aforementioned needs. The challenge is to learn new unseen gestures on-the-fly, from scratch and using very few samples. Furthermore, the classification system has to remain robust and has to maintain its knowledge about the existing gestures, when introducing new unseen ones anytime during the lifelong learning process. However, the online nature of the incremental learning process requires that the classification performance of the system must be acceptable as much as possible even when adding new classes. A high performance is very hard to reach using the very few data samples that are available in the beginning of learning a new class. To cope with this problem, we integrate in our method a handwriting generation technique by which many synthetic learning data samples can be generated starting from the original real data. The handwriting generation is based on the Sigma-Lognormal representation of handwritten strokes. In this paper, we briefly describe our evolving classifier structure and learning algorithm in section 2. A short description of the Sigma-Lognormal model is presented in section 3. Then, integrating synthetic data based on Sigma-Lognormal model in the incremental learning process is explained in section 4. We present our experimental results in section 5 before concluding in section 6.

2 Evolving Fuzzy Classifier

Our system is based on first-order Takagi-Sugeno (TS) fuzzy inference system. It consists of a set of fuzzy rules of the following form:

$$\textbf{Rule}_i : \textbf{IF } \vec{x} \textit{ is close to } P_i \textbf{THEN } y_i^1 = l_i^1(\vec{x}), ..., \ y_i^k = l_i^k(\vec{x}) \quad (1)$$

where $l_i^m(\vec{x})$ is the linear consequent function of the rule i for the class m:

$$l_i^m(\vec{x}) = \vec{\pi}_i^m \vec{x} = a_{i0}^m + a_{i1}^m x_1 + a_{i2}^m x_2 + ... + a_{in}^m x_n \quad (2)$$

where n is the size of the input vector. The Prototype P is defined by a center and a fuzzy zone of influence. To find the class of $\vec{x}$, its membership degree $\beta_i(\vec{x})$ to each fuzzy prototype is first computed. After normalizing these membership degrees, the sum-product inference is used to compute the system output for each class:

$$y^m(\vec{x}) = \sum_{i=1}^{r} \bar{\beta}_i(\vec{x}) \ l_i^m(\vec{x}) \quad (3)$$

where r is the number of fuzzy rules in the system. The membership degree is computed by the prototype center $\vec{\mu_i}$ and its variance-covariance matrix A_i using the multivariate Cauchy probability distribution.:

$$\beta_i(\vec{x}) = \frac{1}{2\pi\sqrt{|A_i|}} \left[1 + (\vec{x} - \vec{\mu_i})^t A_i^{-1} (\vec{x} - \vec{\mu_i})\right]^{-\frac{n+1}{2}} \quad (4)$$

The incremental learning algorithm of our model consists of three different tasks: the creation of new rules, the adaptation of the existing rule's premises, and the tuning of the linear consequent parameters. These three tasks must be done in an online incremental mode and all the needed calculation must be completely recursive.

The importance of a given new sample in an incremental clustering process can be evaluated by its *potential* value. The potential of a sample is defined as inverse of the sum of distances between a data sample and all the other data samples. A recursive method for the calculation of the potential of a new sample hes been introduced in [2]. The recursive formula avoids memorizing the whole previous data but keeps - using few variables - the density distribution in the feature space based on the previous data (see [1] for more details). A Premise adaptation process allows to incrementally update the prototype centers coordinates according to each new available learning data, and to recursively compute the prototype covariance matrices in order to give them the rotated hyper-elliptical form. For each new sample, the center and the covariance matrix of the prototype that has the highest activation degree are updated recursive manner [1]. The tuning of the linear consequent parameters in a first-order TS model can be done by the weighted Recursive Least Square method (wRLS). More details are available in [1].

3 Sigma-Lognormal Model

The Sigma-Lognormal [3] model consider the neuromuscular system as a cascaded network of a large number of subsystems coupled through a proportionality law. Applying the central limit theorem to this representation shows that such a system will produce elementary movement units having a lognormal speed profile [4]. Such a unit, referred to as a neuromuscular component, is scaled and time-shifted by values labeled respectively as D and t_0, and has a lognormal shape characterized by the time delay (μ) and the response time (σ) of the neuromuscular system involved in its production. Moreover, for planar motions, it is hypothesized that each neuromuscular component is made along a fixed pivot and hence, has a circle-arc trajectory starting at an angle θ_s and ending at an angle θ_e.

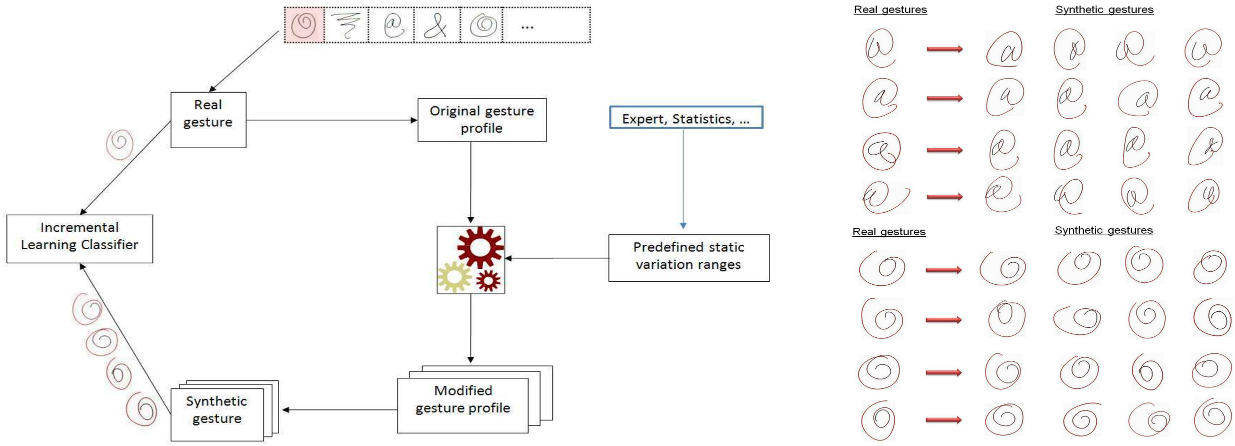
The speed profile and the angular variation of a component are described by (5) and (6) where P_j is the vector of Sigma-Lognormal parameters of the j^{th} component (i.e. $P_j = [t_{0j} \ D_j \ \sigma_j \ \mu_j \ \theta_{sj} \ \theta_{ej}]$).

$$v_t(t; P_j) = \frac{D_j}{\sigma_j (t - t_{0j})\sqrt{2\pi}} e^{-\frac{\left[ln(t - t_{0j}) - \mu_j\right]^2}{2\sigma_j^2}} \quad (5)$$

$$\phi(t; P_j) = \theta_{sj} + \frac{\theta_{ej} - \theta_{sj}}{D_j} \int_0^t v_t(\tau; P_j) d\tau \quad (6)$$

The Sigma-Lognormal model generates complex motions by a vectorial summation of time overlapping neuromuscular components. Once a parameter set is defined, velocity in Cartesian space can be computed using (7).

$$v_x(t; P) = \sum_j v_{tj}(t; P_j) cos(\phi_j(t; P_j)) \quad , \quad v_y(t; P) = \sum_j v_{tj}(t; P_j) sin(\phi_j(t; P_j)) \quad (7)$$



(a) (b)

Figure 1: (a) Integrating data generation in the incremental learning process (b) some examples of generated gestures

4 Learning acceleration using synthetic data

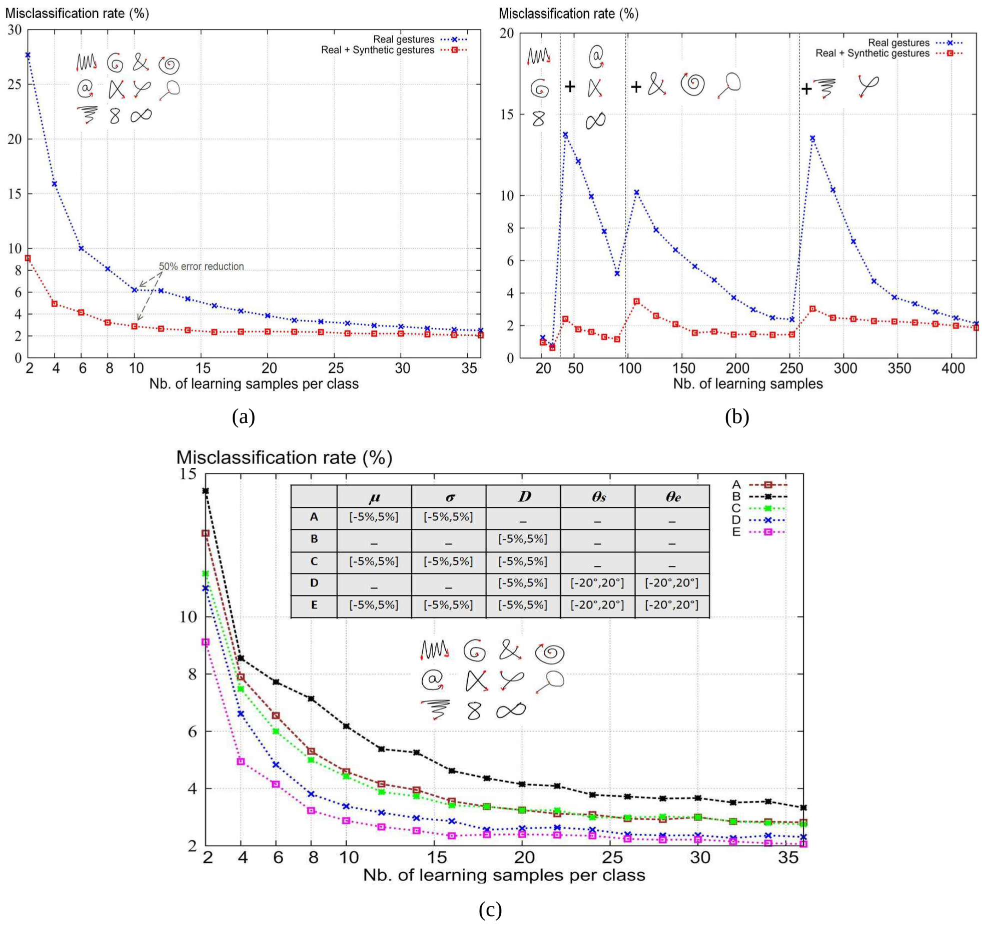
Even with a good incremental learning algorithm and an elastic classifier structure, the lack of available learning data in online learning methods remains as important restriction especially in our application where the user needs to use the new added gestures and to get an acceptable classification accuracy as fast as possible. For this reason, we focused on the idea of generating synthetic data samples in order to boost the learning process. Artificial handwriting generation had been already approached, but the deformations on the handwritten shape were always applied on coordinate's level, which produces unrealistic shapes in many cases. Obviously, applying deformations on sigma-lognormal profiles level allows obtaining mush more realistic synthetic data. The idea is to extract the sigma-lognormal profiles of a real handwritten gesture provided by the user. Then, we apply some variation on the extracted parameters within some specific ranges, and we regenerate artificial gestures using the modified profiles. Real and artificial gestures are both used in the incremental learning of our evolving classifier (see Figure 1).

5 Experimental results

We led the experiments on a dataset of on-line handwritten gestures. It is composed of 11 different gestures drawn by 7 right-handed writers on a Tablet PC. Each writer has drawn 100 samples of each gesture, i.e. 1,100 gestures in each writer specific dataset. Each gesture is described by a set of 10 features. The presented results are the average of results of 7 different tests for the 7 writers. In order to get the results unbiased by the data order effect, we repeat the experiment for each writer 40 times with different random data orders and the mean results are considered. We used about 40% of the dataset for the incremental learning process and the rest is used to estimate the evolution of the performance during the learning process. We generate 10 synthetic samples (gestures) for each real learning sample. Results are shown in Figure 2. By comparing the performance obtained when considering synthetic data samples in the learning process against the one with only the few real samples, we note that the recognition error rate decreases by about 50% thanks to synthetic data samples. Figure 2 shows also that using synthetic samples, the classifier resists better when introducing new classes and it is able to re-estimate rapidly all its parameters and to improve rapidly the recognition performance for the old and the new gestures.

6 Conclusion

In an online incremental learning of handwriting classifier, the lack of learning data represents an important challenge that can sometime lead to unacceptable performance and make the application difficult to use by the user. Thanks



	μ	σ	D	θ_s	θ_e
A	[-5%,5%]	[-5%,5%]	_	_	_
B	_	_	[-5%,5%]	_	_
C	[-5%,5%]	[-5%,5%]	[-5%,5%]	_	_
D	_	_	[-5%,5%]	[-20°,20°]	[-20°,20°]
E	[-5%,5%]	[-5%,5%]	[-5%,5%]	[-20°,20°]	[-20°,20°]

Figure 2: (a)(b) Performance evaluation during the incremental learning process (c) Variation ranges of sigma-lognormal parameters and impact of each parameter variation on classification performance.

to the Sigma-lognormal representation of handwritten gestures, we could use synthetic gestures in the learning process to enhance the prediction capacity of the classifier. Experimental results shows that the deformed samples help to significantly improve the classification performance if our system in an incremental learning scenarios. One of our perspectives is to replace the predefined static variation ranges that we used to generate artificial samples by an automatic mechanism that can (incrementally) estimate the best variation ranges for each parameter and for each user.

References

[1] A. Almaksour and E. Anquetil. Improving premise structure in evolving takagi-sugeno neuro-fuzzy classifiers. *Evolving Systems*, In press.

[2] P.P. Angelov and D.P. Filev. An approach to online identification of takagi-sugeno fuzzy models. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(1):484–498, Feb. 2004.

[3] R. Plamondon and M. Djioua. A multi-level representation paradigm for handwriting stroke generation. *Hum Mov Sci*, 2006.

[4] Rejean Plamondon, Chunhua Feng, and Anna Woch. A kinematic theory of rapid human movement. part iv: a formal mathematical proof and new insights. *Biological Cybernetics*, 89(2):126–138, 2003.