Noctis

Centrality and Dispersion Indices
Signature Recognition System

Andrea Sergas
a.sergas@studenti.uniba.it

Donato Meoli
d.meoli1@studenti.uniba.it

June 12, 2017

University of Bari ”Aldo Moro”
Department of Computer Science

Table of Contents

1. Features Extraction

2. Train Phase

3. Tuning Phase

4. Results

5. Future Development

6. License

Online & Offline Signature Verification

Signature verification is split in two approaches, according to the input type of data:

- **offline** (static) signature verification takes as input the image of a signature;
- **online** (dynamic) signature verification uses signatures that are captured by pressure-sensitive tablets that extract dynamic properties of a signature, in addition to its shape.

As a result, online signature verification is more reliable than offline signature verification because it makes the signature more unique and difficult to forge.

Features Extraction

Noctis

Noctis - *Centrality and Dispersion Indices Signature Recognition System* - is a recognition system for online handwritten signature verification based on a statistical approach.

Why ”Statistical”?

Signature Representation

For each signature the features extracted are represented as a vector of local and global statistical features:

$$S_i = \langle\, \mu(d_i),\, \mu(v_i),\, \mu(a_i),\, \mu(\Delta p_i),\, \mu(\Delta t_i),\, \mu(an_i),\, \mu(an_{0_i}),$$

$$\sigma(d_i),\, \sigma(v_i),\, \sigma(a_i),\, \sigma(\Delta p_i),\, \sigma(\Delta t_i),\, \sigma(an_i),\, \sigma(an_{0_i}),\, t,\, l \,\rangle$$

where μ is the **mean** - *centrality* index - and σ is the **standard deviation** - *dispersion* index.
Only mean and standard deviation are considered to make a drastic reduction of redundant data to pass to the classifier by reducing each feature to only one significant and discriminating value.

Features Extraction I

For each signature of the user, the basic data which are provided by SUSIG dataset are:

- x coordinate: scaled cursor position along the x-axis
- y coordinate: scaled cursor position along the y-axis
- timestamp: system time at which the event was posted
- pressure: adjusted state of the normal pressure in milliseconds

while, starting with these, the following features are extracted:

- Displacement:

$$d_i = \begin{cases} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} & 1 \leq i \leq n-1 \\ d_n - d_{n-1} & i = n \end{cases}$$

Features Extraction II

- Velocity:

$$v_i = \begin{cases} \frac{d_i}{t_{i+1}-t_i} & 1 \leq i \leq n-1 \\ v_n = v_{n-1} & i = n \end{cases}$$

- Acceleration:

$$a_i = \begin{cases} \frac{v_i}{t_{i+1}-t_i} & 1 \leq i \leq n-1 \\ v_n = v_{n-1} & i = n \end{cases}$$

- Angle between two consecutive points:

$$an_i = \begin{cases} \arctan(y_{i+1} - y_i, x_{i+1} - x_i) & 1 \leq i \leq n-1 \\ an_n = an_{n-1} & i = n \end{cases}$$

- Angle between a point and the origin:

$$an_{0_i} = \arctan(y_i - 0, x_i - 0) \quad 1 \leq i \leq n$$

Features Extraction III

- Pressure variations between two consecutive points:

$$\Delta p_i = p_{i+1} - p_i$$

- Time variations between two consecutive points:

$$\Delta t_i = t_{1+1} - t_i$$

- Total time:

$$t = \sum_{i=1}^{n} t_i$$

- Total path length:

$$l = \sum_{i=1}^{n} d_i$$

where n is the number of the samples for each data in the user's signatures representation.

Train Phase

User's SVM

For the users' signatures modeling phase it has been used one Support Vector Machine for each user. The train set for each SVM has been mostly auto generated: from the 10 genuine signatures' features vectors other 80 features vectors have been created by adding small noises, in order to maintain consistency.
No forgery signatures' features have been used in the train set.

Adding Noises

Because of the features' statistical nature, it is possible to generate noise maintaining some certain amount of consinstency by multiplying the given genuine features by a certain number of certain multipliers.

How to choose those "certain" number and multipliers?

Linearly Spaced Vector

The solution is given by creating a linearly spaced vector. Supposing that the creation of the vector is a function with three parameters: range start, number of elements and range end, then it has been empirically proven that the best consistency for the generation of genuine features, for n genuine signatures, is given by:

- **range start** to be 1, so that the first n signatures are the ones in the given dataset;
- **number of elements** to be n - 1;
- **range end** to be the lower integer nearest to:

$$\frac{number\ of\ elements}{2} \cdot 0.1$$

Tuning Phase

User's Threshold

In order to set a specified threshold for each user right after the train phase 90 features vectors have been created by adding each feature used in the train set by the variance of that signature's features, excluding total time and total path.
The user's threshold is then obtained by getting the maximum score predicted from the trained SVM given the 90 auto-generated features vectors. A small constant is then added: the obtained threshold represents an average forgery signature's score.

Results

SUSIG Dataset

For the train and test phases it has been used an online handwritten signature database called SUSIG, which collects signatures for 94 person classified in two classes:

- 20 **genuine** signatures for user divided in two sessions:
  - *session 1* for the train phase;
  - *session 2* for the test phase.

  and vice versa in the 2-fold cross-validation approach used.
- 10 **forgery** signatures for user divided in two classes:
  - *skilled* forgery is signed by a person who has had access to a genuine signature for practice;
  - *highly skilled* forgery is signed is signed by a signatures' forgery professionist who has had access to a genuine signature for practice.

Evaluation I

In evaluating the performances of a signature verification system, there are two important factors:

- **False Acceptance Rate** of forgery signatures which is defined as:

$$FAR = \frac{Number\ of\ accepted\ forgery\ attempts}{Total\ number\ of\ forgery\ attempts} \approx 4,35\%$$

- **False Rejection Rate** of genuine signatures which is defined as:

$$FRR = \frac{Number\ of\ rejected\ legitimate\ attempts}{Total\ number\ of\ legitimate\ attempts} \approx 3,35\%$$

Evaluation II

For the sake of comprehension, another system has been evaluated without using the statistical approach. In this case the features vectors are composed by all the values given in the dataset, with no further preprocessing. The results are the following:

- **False Acceptance Rate** of forgery signatures which is defined as:

$$FAR = \frac{Number\ of\ accepted\ forgery\ attempts}{Total\ number\ of\ forgery\ attempts} \approx 1,54\%$$

- **False Rejection Rate** of genuine signatures which is defined as:

$$FRR = \frac{Number\ of\ rejected\ legitimate\ attempts}{Total\ number\ of\ legitimate\ attempts} \approx 23,94\%$$

Future Development

Adding Features & Hyper-parametrization

This recognition system can be further improved by following two approaches, independent one from the other:

- using technologically advanced devices that provide **more features** about signed signatures such as:
  - *azimuth*: clockwise rotation of cursor about the z-axis;
  - *altitude*: angle upward toward the positive z-axis;
- using **hyper-parameterization algorithms** which prevent the phenomenon of overfitting through the use of the learning rate in a preventive manner: the system, at each step of optimizing its loss function, declines the learning rate.

License

License

Get the source of this software from:

github.com/DonatoMeoli/SSRS

github.com/MrNobody1992/Noctis

The End