The University of Calgary
Department of Electrical and Computer Engineering

ENCM 509 - Fundamentals of Biometric Systems Design

Laboratory Experiment #2
Biometric data matching.

1 Introduction

The purpose of this laboratory exercise is to learn handling and processing the acquired biometric data (the authentic and forged signatures of an individual), preparing this data for usage in 1:1 matching. The data acquired in Lab 1 and in this lab are on-line signatures, represented by the pre-extracted features: coordinate, time and pen pressure information. This means that no feature extraction (the essential step in data mining) is required in this lab. The pre-extracted features will be used for matching demo, which use a statistical method called *Gaussian mixture model (GMM)*. In this exercise, we will also look how to handle multiple-file input data in Matlab, and how to graph some statistics in Matlab.

The signature can be characterized as a random process: no two signatures of the same person are exactly alike. The same is true for the signals such as on-line signature coordinates, pressure and speed. Having a periodic signal repeated in the time domain, in the frequency domain there are harmonics related to the period of the signal. What we need is the envelope of these harmonics to characterize the signature. This can be achieved by taking the peaks of the harmonics in the frequency domain. Such an envelope is represented by a mixture of Gaussian components.

The GMM classifier is a Bayesian classifier (meaning statistical one, based on the Bayes formula for posteriori probability), which uses Gaussian mixture probability density function models, and maximum likelihood parameter estimation methods. Gaussian mixture model is a weighted sum of Gaussian probability density functions which are referred to as Gaussian components of the mixture model, describing a class. The goal is to implement a Bayesian classifier that can handle any feasible number of variables (data dimensions), classes and Gaussian components of a mixture model. Data used in classification are assumed to come from a proper feature extraction, to be real-valued, and not to have any missing values in samples. A simple classifier separate the data into two classes. The classifier system contains two main components: a training function and a classification function, both implemented in Matlab. The training is based on the expectation maximization (EM) algorithm.

The main idea is to represent the biometric features as the probability density function of one signee, with a mixture of Gaussians. A Gaussian mixture density function is a weighted sum of M constituent functions,

$$p(x|\lambda) = \sum_{i=1}^{M} a_i f_i(x)$$

where x is a D-dimensional random vector, $f_i(x)$, $i = 1, 2, ..., M$ are the constituent Gaussian density functions, and a_i are the mixture weights. Each constituent Gaussian density is a D-variate normal density function. Note that for probabilities $p(x|\lambda)$, the constraint $\sum_{i=1}^{M} a_i = 1$ must be satisfied. Each individual has its own density function, that is $p(x|\lambda_s)$ where λ_s represent the different individuals. A maximum likelihood classifier can be used for this model. Given several individuals, it is required to find the individual with maximum posterior probability for the feature vectors X. The probability of being the individual λ_s, given the feature vectors of the submitted signature, is calulated using the Bayes formula

$$P(\lambda_s|X) = \frac{P(X|\lambda_s)P(\lambda)}{P(X)}$$

We assume equal prior probabilities $P(\lambda)$ for all writers. $P(X)$ is deemed to be a constant for all writers. Also, $P(X|\lambda)$ is a joint, or total, likelihood of the feature vector for a given individual, and is calculated as the product of the likelihoods at each point, x_t,

$$P(X|\lambda) = \prod P(x_t|\lambda)$$

The criterion for the selection of one individual as the correct one is maximum likelihood,

$$max_s P(X|\lambda) = max_s \prod P(x_t|\lambda)$$

For convenience, we take the logarithm of this product:

$$max_s log \prod P(x_t|\lambda) = max_s \sum log P(x_t|\lambda)$$

in which $P(x_t|\lambda)$ is calculated as shown above for $p(x|\lambda)$.

2 The laboratory procedure

2.1 Preparing the single .mat file in Matlab

Use the data you collected on Lab 1 exercise or have a new set of data containing:

- 10 to 30 samples of your own signatures,
- 10 to 30 samples of your "forged" signature.

An individual's on-line signature acquisition using SigGet software results in a single `.mat` file (for example, `Sig1.mat`). When loaded into Matlab with the `load` command, such as

```
>> Sig1m=load('Sig1.mat'),
```

the variable `Sig1m` is a 1×1 structure array. If you have 10 or more signatures, then loading 10 files (`sig1.mat, sig2.mat, sig3.mat`,...) can be be managed by combining those into one `.mat` file. You can use structure array. For example,

```
sigAll = {'sig1.mat','sig2.mat',,'sig3.mat','sig4.mat','sig5.mat',...
          'sig6.mat','sig7.mat','sig8.mat','sig9.mat','sig10.mat'};
       for i=1:10,
           s(i) = load(sigAll{i}) ;
           sAll_authentic{i} = double(struct2array(s(i))'); %transpose
       end
save sigAllm sAll_authentic;
```

Now, `sigAllm.mat` is a 1×10 structure array with fields `coord`, `prs`, and `time`.

The same procedure can be repeated for your set of 10 forged signatures.

A testing set of 10 original and 10 forged signatures in 2 .mat files can also be found on N drive
N:\\ENCM\509\lab2\authSig.mat and *forgSig.mat*.

2.2 Demo of a signature verification

The demo for this lab includes the main file called Lab2sigVerif_demo.m, as well as files gmm_estimate.m, graph_gmm.m, histn.m, lmultigauss.m, and lsim.m. When running the file Lab2sigVerif_demo, it suggest to choose the values for:

- the number of signatures used for training (usually half of the collected set size),
- the number of Gaussian components (usually between 1 and 80; for example, use 20, 40 and 60).

The demo produces the plots of:

- five graphs, each for one of available coefficients (5 for the sample set of signature), and each showing histogram of data, multigauss coefficients vs. data, and Gauss components vs data on the same graph,
- the normalized probability density functions `aProb` (green line) and `fProb` (red line) within the range -45:0.01:-15, as well as `aScores` (green squares) and `fScores` (red triangles).

2.3 Lab procedure step by step

- Create the two `.mat` files: one with your authentic and one with forged signatures (10 to 30) ; convert these data into `double` format as described in section "Preparing the single `.mat` file". You may later study the performance of the matching algorithm depending on the size of the sample (for example, 20 vs 30 signatures), so you may create several `.mat` files of various sizes (f.e. 10, 15, 20, etc.)

- On the N drive, in ENCM/509, you will find directory 'lab2'. You will see files: Lab2sigVerif_demo.m, gmm_estimate.m, graph_gmm.m, histn.m, lmultigauss.m, and lsim.m. Run the file Lab2sigVerif_demo.m and use your input data `.mat` files. Repeat for 3-4 various values of the parameters (the number of signatures used for training, and the number of Gaussian mixture components). Save the displayed figures for your report if necessary. Plot the graph showing `aMu` versus the number of signatures used fo training.

- Consider the matching score (note that this is a logarithmic value) as your normally distributed data (plot the probability density function vs matching score). Find (do not use Matlab) the mean (average) and standard deviation for the authentic and forgery scores (Matlab variables `aMu, fMu,` and `aStd, fStd`, respectively). In your report, you will need to answer the question: What do these values represent?

- Consider your plots for 40 (or other reasonable number) Gaussian components. Formulate a hypothesis H_0 about the μ and σ of the entire population of the authentic signatures, and analytically test this hypothesis based on your sample (your 10 or more signatures) given the level of the test significance equal to 0.05. Now, change the critical values (choose reasonalbe values) and evaluate the FRR value. Do the same for the forged signatures.

- Consider your plots for authentic signature distribution, given 40 (or other reasonable number) Gaussian components. Formulate a hypothesis H_1 with μ and σ close to those of the forged signatures, and evaluate the FAR value. Find the power of the test.

3 Laboratory Report

Include the following in your report:

- The verification results in the form of Matlab plots created in demo, for selected values of the parameters (number of signatures used for training samples, and the number of Gaussian mixture components)

- Your analysis of how varying the above parameters influence on the classification results. This may include graphs or tables illustrating the dependance (for instance, `aMu` vs. the number of signatures in the training set).

- Your analysis of the mean and standard deviation values for the authentic and forgery scores, as well as your hypothesis testing given the level of the test significance, and FRR and FAR evaluation. Answer the questions: How to reduce both FRR and FAR? How to increase the power of test?

Attach `.m` files containing your code, or include them in your report document at the end.

4 Acknowledgments

We wish to thank research students who contributed to the projects implemented in the Biometric Technologies Laboratory. The implementation of the GMM in Matlab are credited to J. Richiardi, EPFL. We acknowledge the support of the Department of Electrical and Computer Engineering and IT staff.

Svetlana Yanushkevich
September 22, 2015