Report for phase II evaluation of EE 499

ONLINE WRITER IDENTIFICATION USING GAUSSIAN MIXTURE MODELS

Submitted

By

Sameer Kumar Ghawana — **170108034**

Nishit Gaur — **170108027**

Under the supervision of

Dr. Suresh Sundaram

Department of Electronics & Electrical Engineering

Indian Institute of Technology Guwahati

April 2021

TABLE OF CONTENTS

Abstract

We present a scheme that uses on-line handwriting data taken from a whiteboard to describe the writer's name in this article. We must decide which person within a group of people is the text's author. After a few pre-processing steps, the data is translated to the correct format, and then features specifying writer attributes are extracted from the data. We use Gaussian Mixture Models (GMMs) to model each person's handwriting in the underlying population since they enable us to describe the distribution of features derived from handwritten text. To begin, all of the authors' training data is used to train a single Universal Background Model , which is then used to create a writer-specific model. A Gaussian mixture density model is used to model the distribution of feature vectors derived from a person's handwriting. The program will output a log likelihood score which correlates to the probability of that writer being the author of the text.

Introduction

As computers are now being widely used in every field of work, person identification using handwritten notes should be automated. The main principle behind writer identification is that each person has unique handwriting and can be identified by looking at certain features derived from their handwriting.

Writer identification is done primarily on two types of data namely online and offline. The two are differentiated on the basis of type of input. If the input is available in the form of scanned images then it is classified as offline data while, if spatial and temporal information about writing is available then it is classified as online data. It is important to note that online data contain more information about the user's writing style, speed etc. compared to offline data. In this project we are primarily dealing with online data.

The data about the position and velocity of pen is sent through an IR sensor present at the tip, the text written is recorded through a triangular sensor/ receiver attached at one of the corners of the whiteboard. The output of the sensor are (x , y)-coordinates of the pen tip along with time stamps of each location.

Handwritten texts from various writers are fed into the method, yielding a series of strokes, each of which contains a sequence of points. There could be gaps or factitious points within the strokes if writer’s hand comes in between the whiteboard and infrared signals from the pen, or if the pen is tilted too much while writing. Due to this reason pre-processing of the original data is required before training GMMs of every author. So when a paragraph of any random author is given as an input to our model, each model outputs a log likelihood score and the author whose GMM outputs the largest score is concluded as the author of the text.

Motivation

Automatic writer identification is one of the most troubling things in this automated era. With the growing use of the Internet, the interactions have become increasingly automated and thus the problem of identity theft has become much more serious. Most author identification work is done manually to date as many details that rely on content, such as source of documents, handwriting type, etc. are difficult to model mathematically. However, they can be easily examined by human experts. Still, an author identification system is useful as it can remove subjectivity from the handwriting recognition process.

Further iterations of this program can be used intensively by corporations in their meeting rooms to archive the meeting details which currently are performed manually. Further modifications of this algorithm can be used by people to improve their handwriting.

Literature Review

Our work is primarily based on [*Schlapbach,A., Liwicki M., and Bunke H. (2007) "A Writer Identification System for On-line Whiteboard Data" Institute of Computer Science and Applied Mathematics, Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland*].
The work in this paper deals primarily with the task of writer identification using GMMs.

We took inspiration from on-line handwriting recognition work done in the past. Specific features are chosen because they are capable of capturing almost every aspect of the person's handwriting. We have applied GMMs to model the distribution of the features for each writer. These distributions will be unique for each writer.

Firstly, a Universal Background model is trained by using all the training data from all the writers. Secondly, a writer specific model (GMM) is obtained for each writer through a process of adaptation using UBM and the training data for that writer. After the training step, we get a model for each writer. In the testing session, text of any random author is presented to each model and the model outputs a log likelihood score and then these scores are arranged in descending order (Schlapbach et al., 2007). The author with the highest GMM score is concluded as the writer of the text.

Work Plan

A. Data pre-Processing : The first step in our model is the pre-processing of data. Online handwriting data is not perfect and contains many errors like noise points which are unwanted lines or points in a paragraph and primarily occur due to transmission errors. Another type of error is gaps between stroked and they occur mainly due to loss of transmission between tip of pen and whiteboard receiver. We have several simple steps at our disposal to counter these errors.

B. Feature Extraction : The temporal and spatial information about the handwriting is available in the form of (x, y) coordinates of the pen tip and time stamps for each position, so we must extract useful features from the results.

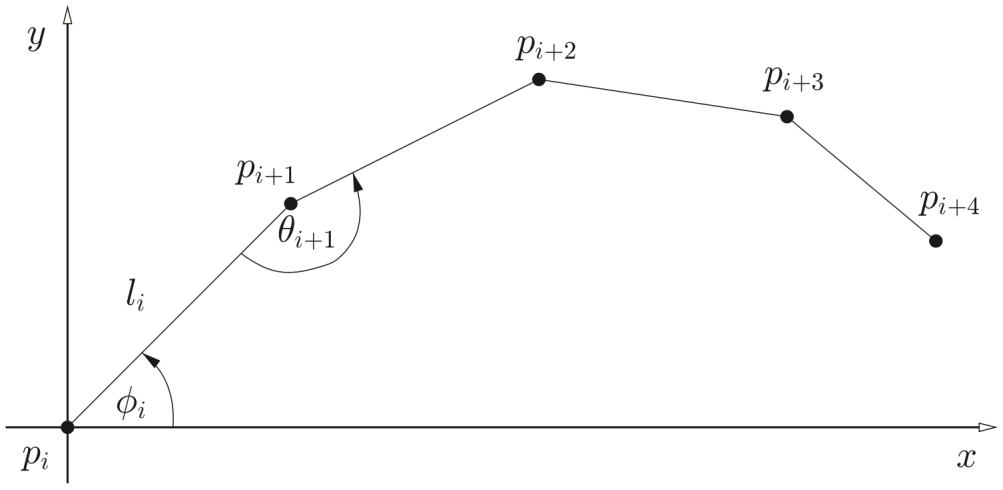


Fig. 1. Description of point-based Features

C. Feature Modelling : A Universal Background model is trained by using all the training data from all the writers. Then a writer specific model (GMM) is obtained for each writer through a process of adaptation using UBM and the training data for that writer.

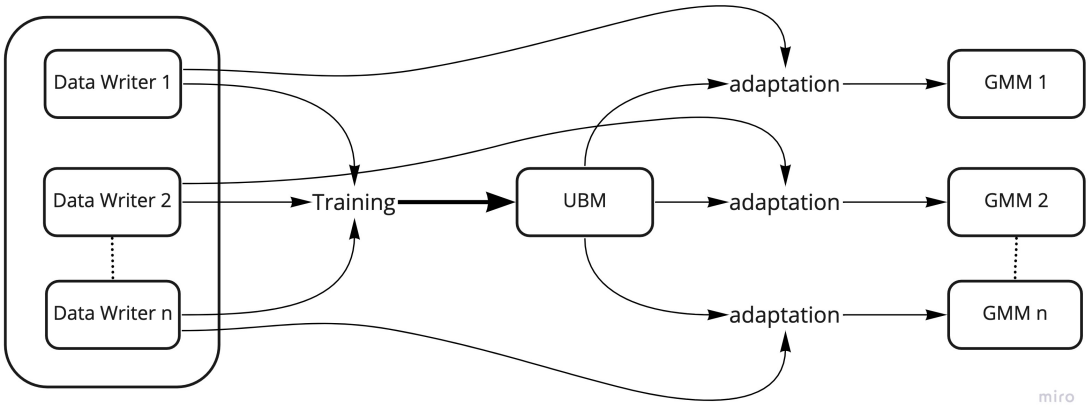


Fig.2. Illustration of the training phase.

D. Testing : A paragraph of any random author is given as an input to our model, each GMM outputs a log likelihood score and the author whose GMM outputs the largest score is concluded as the author of the text.

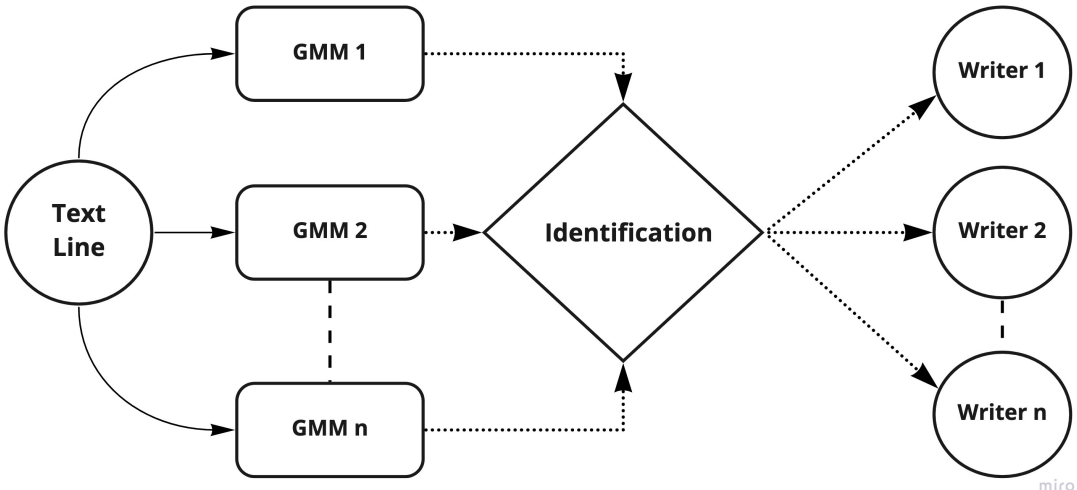
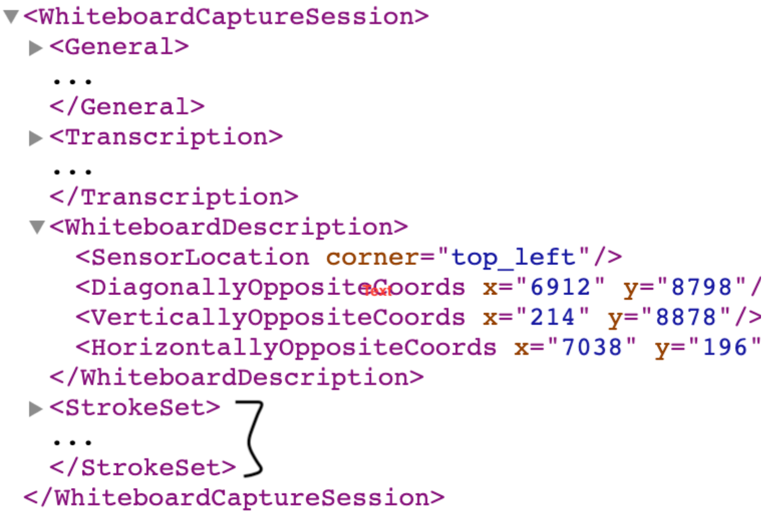


Fig.3. Schematic overview of the testing phase.

System Model

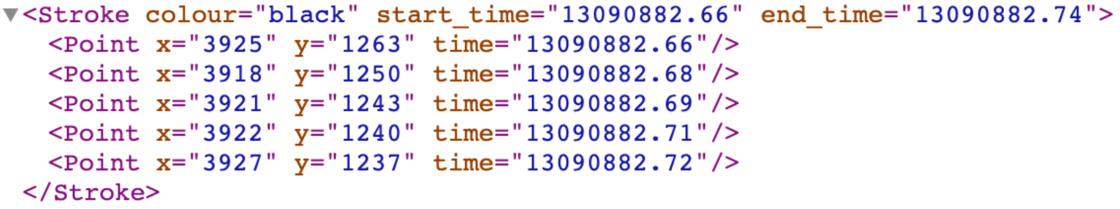
A) Brief description of the dataset

We have used The IAM On-Line Handwriting Database (IAM-OnDB)
It has data of 196 unique writers and for each of the writers, there are eight paragraphs of the text, in which each paragraph of text consists of eight text lines on an average.
For each writer data of 8 sessions is available in form of .xml files where each session is essentially a collection of multiple strokes.



```
▼<WhiteboardCaptureSession>
  ▶<General>
   ...
   </General>
  ▶<Transcription>
   ...
   </Transcription>
  ▼<WhiteboardDescription>
     <SensorLocation corner="top_left"/>
     <DiagonallyOppositeCoords x="6912" y="8798"/
     <VerticallyOppositeCoords x="214" y="8878"/>
     <HorizontallyOppositeCoords x="7038" y="196"
   </WhiteboardDescription>
  ▶<StrokeSet>
   ...
   </StrokeSet>
 </WhiteboardCaptureSession>
```

Each stroke is a collection of points whose data is available as 3 different parameters (*x, y, t*). Each session contains multiple strokes of varying lengths.



```
▼<Stroke colour="black" start_time="13090882.66" end_time="13090882.74">
   <Point x="3925" y="1263" time="13090882.66"/>
   <Point x="3918" y="1250" time="13090882.68"/>
   <Point x="3921" y="1243" time="13090882.69"/>
   <Point x="3922" y="1240" time="13090882.71"/>
   <Point x="3927" y="1237" time="13090882.72"/>
 </Stroke>
```

The data (.xml files) is extracted as 8 different sessions (.mat files) for each author and these sessions are used to train the GMMs.

B) Data Preprocessing and Feature Extraction

We have taken the strokes as n-tuple of x, y coordinates and time stamps and have broken the strokes when they turn by more than right-angles to ensure that the writing styles are accurately captured by the models. Apart from this preprocessing, we have also split the longer strokes into same sized strokes by resampling so as to generate feature vectors of same sizes. For the purpose of our project, we have used two sets of features which we have described below. One of the feature set is based purely on point based properties of the strokes while the other set of features is based on both point based properties and the properties of a stroke as a whole.

❑ Point-based properties of a stroke

The following features for and consecutive pair of points was computed for a given stroke 's' consisting of points p_1 to p_n. Angle Φ denotes the angle formed by the horizontal line and the line (p_i, p_{i+1}), while angle θ_i denotes the angle formed by the lines (p_{i-1}, p_i) and (p_i, p_{i+1}).

The following features are calculated for each point p_i :

- **Speed :** The speed v_i of the segment is given by $v_i = \frac{\Delta\left(p_i, p_{i+1}\right)}{t}$ where t denotes the acquisition device's sampling rate.

- **Writing Direction :** The writing direction at p_i , i.e., the cosine and sine of θ_i

$$\cos\left(\theta_i\right) = \Delta x\left(\frac{p_i, p_{i+1}}{l_i}\right) \;\&\; \sin\left(\theta_i\right) = \Delta y\left(\frac{p_i, p_{i+1}}{l_i}\right) \text{ respectively.}$$

- **Curvature :** The curvature i.e. the cosine and sine of angle Φ. The following trigonometric formulas are used to calculate these angles:

$$\cos\left(\phi_i\right) = \cos\left(\theta_i\right)\cos\left(\theta_{i+1}\right) + \sin\left(\theta_i\right)\sin\left(\theta_{i+1}\right)$$

$$\sin\left(\phi_i\right) = \cos\left(\theta_i\right)\sin\left(\theta_{i+1}\right) - \sin\left(\theta_i\right)\cos\left(\theta_{i+1}\right)$$

Thus, the **Feature Set-1** includes **5** attribute values.

```
%Writing Direction
costheta = zeros(29,1);
sintheta = zeros(29,1);
for i=2:30
    l = norm(rs(i,1:2)-rs(i-1,1:2));
    costheta(i-1) = (rs(i,1)-rs(i-1,1))/l;
    sintheta(i-1) = (rs(i,2)-rs(i-1,2))/l;
end

%Stroke Velocity
stroke_speed = zeros(29,1);
for i=2:30
    stroke_speed(i-1)=(norm(rs(i,1:2)-rs(i-1,1:2)))/(rs(i,3)-rs(i-1,3)
end

%Stroke Curvature
cosphi = zeros(28,1);
sinphi = zeros(28,1);
for i=2:29
    cosphi(i-1)=costheta(i-1)*costheta(i) + sintheta(i-1)*sintheta(i);
    sinphi(i-1)=costheta(i-1)*sintheta(i) - sintheta(i-1)*costheta(i);
end
```

Fig. Point based features (MATLAB)

❑ Properties of stroke as a whole

Specific characteristics are dependent on strokes in this collection. The following features are calculated for each stroke s $= p_1, \ldots, p_n$

- **Accumulated length :** The accumulated length l_{acc} of all lines l_i

$$l_{acc} = \sum_{i=1}^{n-1} l_i$$

```
function l = getStrokeLength(s)
    l = 0;
    for i = 2:size(s,1)
        prev_point = s(i-1,1:2);
        this_point = s(i,1:2);
        l = l + norm(this_point - prev_point);
    end
end
```

Fig. MATLAB code for l_{acc}

- **Accumulated angle :** The angle formed by adding the absolute values of the angles of all lines' writing directions.

$$\theta_{acc} = \sum_{i=1}^{n-1} \left| \theta_i \right|$$

- **Height and Width :** The height $h = y_{\max} - y_{\min}$ and the width $w = x_{\max} - x_{\min}$ of the stroke.

- **Stroke Duration :** The duration t of the stroke

The Feature Set-2 contains **9 feature values** (5 Point based features + 4 stroke based features)

C) Feature Modelling using GMMs

A Gaussian Mixture is a function which comprises of several Gaussians, each identified by $i \in \{1,\ldots,N\}$, where N is the number of clusters. Each Gaussian is comprised of the parameters (μ_i, Σ_i, π_i) where

μ_i = Mean (Centre of the Gaussian)

Σ_i = Covariance (Width of the matrix)

π_i = Mixing Probability (π_k) defined for each gaussian in a distribution

$$\sum_{i=1}^{N} \pi_i = 1$$

For a D dimensional feature vector **x** the mixture density for a specific author is defined as

$$p(\mathbf{x} \mid \lambda) = \sum_{i=1}^{N} \pi_i p_i(\mathbf{x})$$

Where $\lambda = \left\{\pi_i, \mu_i, \Sigma_i\right\}$ for all i = 1,…, N.

A weighted linear combination of N distinct gaussian densities is the Gaussian mixture density p_i (x), each characterised by a D x 1 mean vector μ_i and a D x D covariance matrix Σ_i and represented as:

$$\mathcal{N}(\mathbf{x} \mid \mu, \Sigma) = \frac{1}{(2\pi)^{D/2} \mid \Sigma \mid^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)\right)$$

The MATLAB built in function which we have used to train our GMMs is $fitgmdist(X, K)$. It fits a Gaussian mixture distribution with K components to the data in X. X is an N x D matrix, rows of X corresponds to observations and the columns corresponds to variables. **fitgmdist** fits the model by maximum likelihood, using the Expectation- Maximisation (EM) algorithm.

```
function model = gmmTrain(X, k)
    %MATLAB built in function
    model = fitgmdist(X,k,'CovarianceType','Diagonal', 'Regularize',0.01);
end
```

This function allows you to specify optional parameters and to control the iterative EM algorithm used to fit the model. We are using two of those parameters :

Covariance Type : The value is set to ‘Diagonal’ if the covariance matrices are restricted to be diagonal; otherwise it is 'full' by default.

Regularisation Value : To guarantee that the calculations are positive-definite, a non-negative regularisation value should be applied to the diagonal of each covariance matrix. The default value is 0.

```
>> execute1
Training model for author with id: 26
Model trained for author with id: 26
Training model for author with id: 27
Model trained for author with id: 27
Training model for author with id: 28
Model trained for author with id: 28
Training model for author with id: 29
Model trained for author with id: 29
Training model for author with id: 30
Model trained for author with id: 30

ans =

  1×5 cell array

  Columns 1 through 4

    {1×1 gmdistribution}    {1×1 gmdistribution}    {1×1 gmdistribution}    {1×1 gmdistribution}

  Column 5

    {1×1 gmdistribution}
```

Fig. GMM Training completed for set of five authors

D) Setup for GMM- Based System

We have trained models for three sets of different count of authors containing 5, 15 and 30 of them respectively. For each of the models, we have trained one each Gaussian Mixture Model for every author. We have taken covariance matrix of each Gaussian distribution to be diagonal and we have used variance flooring factor to be 0.01. As we have 8 paragraphs of text for every author, we have used 7 paragraphs for training and 1 for testing. We have conducted experiments for two feature sets described below:

Feature Set 1 : To construct this feature set, we have sampled every stroke at 30 points. The features consist of all the point-based features described in the above section. Every feature vector has a dimension size of 140.

Feature Set 2: To construct this feature set, we have sam- pled every stroke at 30 points. The features consist of relative x-coordinates of points, relative y-coordinates of points, speed for every point, curvature for every point, accumulated length of every stroke and stroke duration. In total, every feature vector has dimension size of 118.

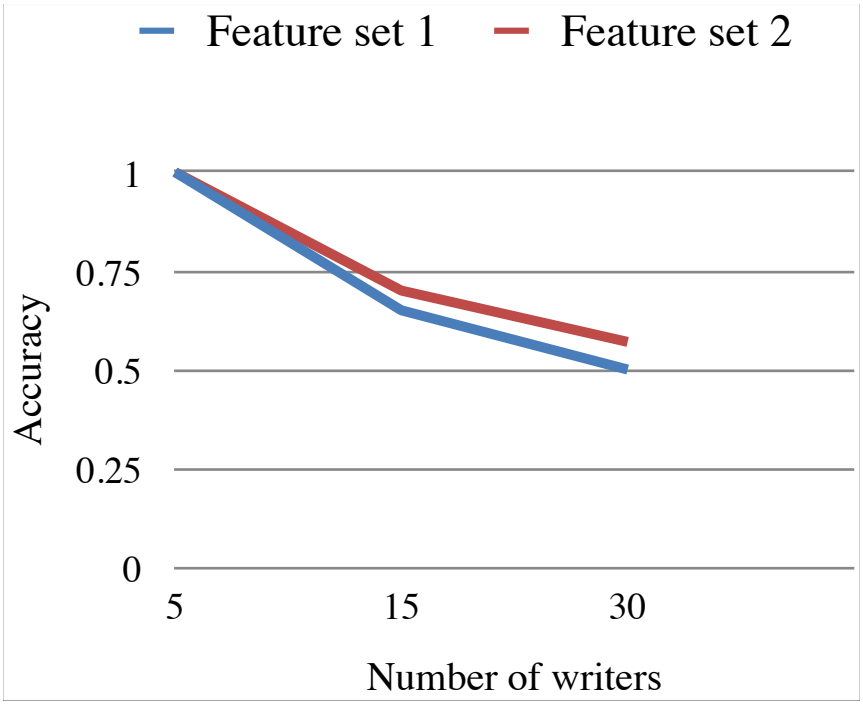
```
features = zeros(5,28);

%Set Feature Vector
features(1,1:28) = costheta(1:28);
features(2,1:28) = sintheta(1:28);
features(3,1:28) = stroke_speed(1:28);
features(4,1:28) = cosphi;
features(5,1:28) = sinphi;
```

Fig. Feature Vector for Set 1 (MATLAB)

Results and Conclusion

We have tested our model for three groups containing different number of authors and compiled our observations in following table. We have observed that the accuracy values are better for feature set 1. This result was expected because in the feature set 2, we have considered only point based properties. While in the feature set 1, we have considered both point based and stroke based properties. Also, we have observed that as we increase the number of authors (classes), the accuracy results go down as expected.

Feature Set	No of Writers	Accuracy
1	5	1.00
1	15	0.65
1	30	0.50
2	5	1.00
2	15	0.70
2	30	0.57



References

[1] Schlapbach,A., Liwicki M., and Bunke H.(2007) "*A Writer Identification System for On-line Whiteboard Data*" *Institute of Computer Science and Applied Mathematics, Universität Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland*

[2] Namboodiri,A., Gupta,S.(2006) "*Text Independent Writer Identification from Online Handwriting*" *Tenth International Workshop on Frontiers in Handwriting Recognition, Université de Rennes*, *La Baule (France).*