A Writer Identification System for On-line Whiteboard Data

Shyamal Kejriwal
11010174
Computer Science and Engineering
Indian Institute of Technology
Guwahati, Assam 781039
Email: k.shyamal@iitg.ernet.in

Vishal Anand
11010170
Computer Science and Engineering
Indian Institute of Technology
Guwahati, Assam 781039
Email: vishal.anand@iitg.ernet.in

Vivek Poddar
11010173
Computer Science and Engineering
Indian Institute of Technology
Guwahati, Assam 781039
Email: v.poddar@iitg.ernet.in

***Abstract*—In this report we have dealt with the task of writer recognition of online handwriting captured from a writing board. A set of features were extracted from this data which was used to train a text and language independent on-line writer identification system. We describe here two systems; firstly Gaussian Mixture Models (GMMs) which provide a powerful yet simple means of representing the distribution of the features extracted from the handwritten text. The second system is based on k-means clustering process. Different sets of features are described and metrics are evaluated in this report. The system is tested using text from a set of around 50 different writers. The GMM model provided us with better identification accuracy as compared to the k-means clustering approach for the same problem.**

***Keywords*—writer identification, online handwriting, Gaussian mixture models, k-means clustering models.**

I. Introduction

In this report we have dealt with the problem of author identification with the focus on Smart Meeting Rooms. The goal of this project is to automate standard tasks that are generally done by human beings in a meeting. To record a meeting, Smart Meeting Rooms are equipped with synchronized recording interfaces to capture audio, video, and handwritten notes.

We encounter interesting and intriguing pattern recognition and classification problems using Smart Meeting Rooms. An important task in a Smart Meeting Room is to capture the handwriting rendered on a whiteboard during a meeting. We have selected the problem of identification of the author of a text written on a board. Solving this problem enables us to associate the writing with the writers identity. To add onto this, it allows us to validate the identification results of a video or audio-based person identification system within the Smart Meeting Room scenario. The text written on the whiteboard is recorded by an eBeam interface. A usual pen put in a casing sends out infrared signals to a receptor placed at a corner of the board. The output of the system is a set of (x, y)-coordinates representing the location of the writing-tip along with the time-stamp for each of the locations. The sample density ranges from 30 to 70 samples per second with a resolution of 4 points per millimeter. False points and gaps within strokes can be recorded if there is an obstruction between the pen and the receiver, or if the pen-tip is highly tilted. A mock-up of the data recording setup is shown in Figure 1.

Fig. 1. Recording session with the data acquisition device positioned in the upper left corner of the whiteboard.

With the recorded data, we use Gaussian Mixture Models (GMMs) [1] to model a person's handwriting. GMMs have previously been applied successfully for the task of speaker identification [2], [3] which is a very similar task to writer identification. GMMs provide a powerful yet simple means of representing the distribution of the features extracted from the text written by one person. GMMs consist of a weighted sum of uni-modal Gaussian densities. For each writer in the considered population, an individual GMM is trained using data from that writer only. Intuitively, each GMM can be understood as an expert specialized in recognizing the handwriting of one particular person. Given an arbitrary text as input, each GMM outputs a recognition score.

We also use k-means clustering [5] based approach for the same problem. We observe that k-means based approach gives much lesser accuracy on the same training and testing sets than GMM based approach. It makes us conclude that GMMs model the features of handwritten text better than the k-means based approach.

II. Writer Identification System using GMMs

We have extracted the features from the recorded handwriting of people and model it using one GMM for each of

the writers. The models are obtained by emulating a two-step training procedure. Initially the complete training data is used to train a single, writer independent universal background model (UBM). In the following step, corresponding to each of the writers, a writer specific model is obtained by adaptation using UBM and training data corresponding to the author. From training process, a model is obtained for each of the writers. In the testing phase, we use text of unknown author and present it to each model, which return the log-likelihood scores; and these scores are sorted. From the ranking of the scores, we assign it to the person whose model produces the highest score. An overview of the complete training and testing procedure is shown in Fig. 2. To train the models, different feature sets are extracted from the text which are described in the following section. Prior to feature extraction, a series of regularization processes are applied. The operations are designed to improve the quality of the features extracted without lowering the writer specific data.

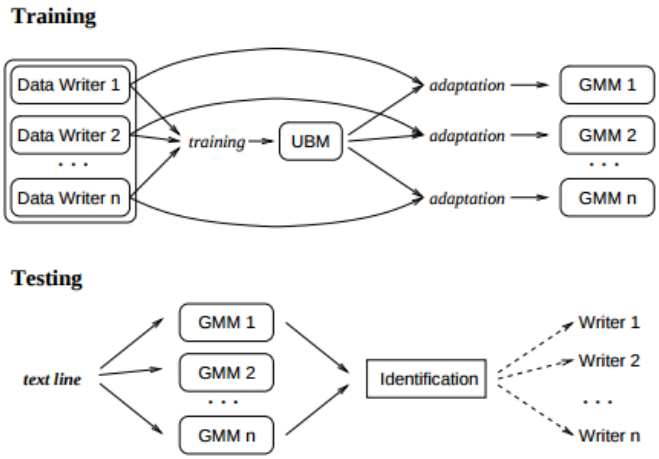

Fig. 2. Schematic overview of the training and the testing phase.

III. Writer Identification System using k-means clustering

We have extracted the features from the recorded handwriting of people and clustered the feature set of every author using k-means clustering algorithm [5]. If we have n-authors and k clusters for each of the authors, then we have a total of n*k clusters with ourselves. Given a test set of feature vectors corresponding to each author, we find the distance of each of the feature vectors from each of the clusters and assign it to the author for which we have the minimum distance. We assign the test set to the author with maximum votes.

IV. Feature Sets for Writer Identification

A. Preprocessing

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. This is done 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 describe below. One of the feature sets is based purely on point properties of strokes, while the other set of features is based on both properties of stroke points and properties of stroke as a whole.

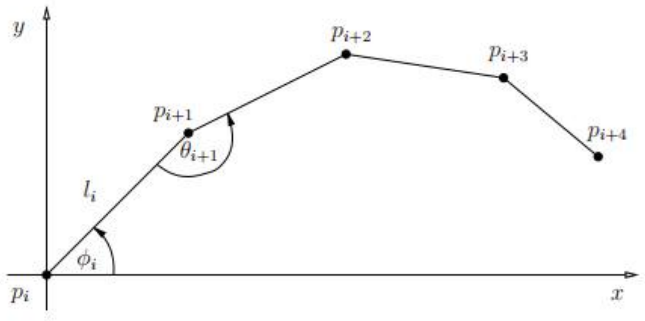

Fig. 3. Illustration of point-based features.

B. Point-based Feature Set

The features of this feature set [4] are similar to the ones used in on-line handwriting recognition systems and signature verification systems. For a given stroke s consisting of points p_1 to p_n, the following features for each consecutive pair of points (p_i, p_{i+1}) are computed. In our notation, angle ϕ denotes the angle between the horizontal line and the line (p_i, p_{i+1}), and angle θ_i represents the angle between the lines (p_{i-1}, p_i) and (p_i, p_{i+1}) (see Fig. 3 for an illustration). The following features are calculated for each point p_i :

- speed (1): the speed vi of the segment

$$v_i = \frac{\Delta(p_i, p_{i+1})}{t}$$

  where t equals the sampling rate of the acquisition device.

- writing direction (2): the writing direction at p_i, i.e., the cosine and sine of θ_i

$$cos(\theta_i) = \frac{\Delta x(p_i, p_{i+1})}{l_i}$$

$$sin(\theta_i) = \frac{\Delta y(p_i, p_{i+1})}{l_i}$$

- curvature (2): the curvature, i.e., the cosine and sine of the angle ϕ_i. These angles are derived by the following trigonometric formulas:

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

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

  The point-based feature set thus contains 5 feature values.

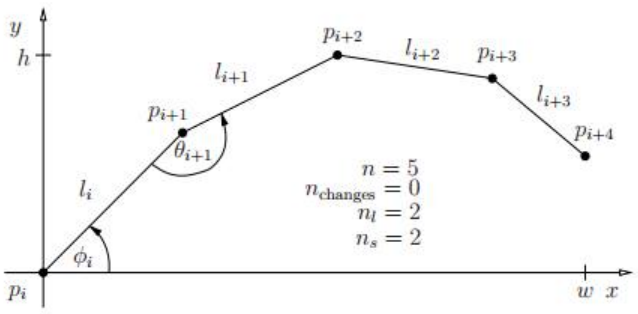

Fig. 4. Illustration of point-based features.

C. Stroke-based Feature Set

In this set, the individual features are based on strokes [4]. For each stroke s = $p_1, \ldots, p_n$ we calculate the following features (for an illustration see Fig. 4):

- accumulated length (1): the accumulated length l_{acc} of all lines l_i
$$l_{acc} = \sum_{i=1}^{n-1} l_i$$
- accumulated angle (1): the accumulated angle θ_{acc} of the absolute values of the angles of the writing directions of all lines:
$$\theta_{acc} = \sum_{i=1}^{n-1} |\theta_i|$$
- width and height (2): the width w = $x_{max} - x_{min}$ and the height h = $y_{max} - y_{min}$ of the stroke
- duration (1): the duration t of the stroke
- time to previous stroke (1): the time difference Δt_{prev} to the previous stroke
- number of points (1): the total number of points n
- number of curvature changes (1): the number of changes $n_{changes}$ in the curvature
- number of up strokes (1): the number of angles n_l of the writing direction larger than zero
- number of down strokes (1): the number of angles n_s of the writing direction smaller than zero

V. Gaussian Mixture Model

We use Gaussian Mixture Models (GMMs) to model the handwriting of each person of the underlying population. The distribution of the feature vectors extracted from a persons handwriting is modeled by a Gaussian mixture density. For a D-dimensional feature vector x the mixture density for a specific writer is defined as

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

where the mixture weights w_i sum up to one. The mixture density is a weighted linear combination of M uni-modal Gaussian densities p_i(x), each parametrized by a D x 1 mean vector μ_i and a D x D covariance matrix C_i

$$p_i(x) = \frac{1}{(2\pi)^{D/2}|C_i|^{1/2}} exp\{-\frac{1}{2}(x-\mu_i)'(C_i)^{-1}(x-\mu_i)\}$$

The parameters of a writers density model are denoted as $\lambda = \{w_i, \mu_i, C_i\}$ for all i = 1, . . . , M. This set of parameters completely describes the model and enables to concisely model a persons writing on the whiteboard.

The GMM is trained using the Expectation-Maximization (EM) algorithm. The EM algorithm follows the Maximum Likelihood (ML) principle by iteratively refining the parameters of the GMM to monotonically increase the likelihood of the estimated model for the observed feature vectors. The algorithm starts with a data set X of T feature vectors x_t, an initial set of M uni-modal Gaussian densities, $N_i = N(\mu_i, C_i)$, and M mixture weights w_i.

Then, in the first step, for each training data point x_t the responsibility $P(i|x_t)$ of each component N_i is determined. In the second step, the component densities, i.e., the mean vector μ_i and the variance matrix C_i for each component, and the weights w_i are re-estimated based on the training data. The models parameters are updated as follows:

$$\mu_i = \frac{\sum_{t=1}^{T} P(i|x_t) * x_t}{\sum_{t=1}^{T} P(i|x_t)}$$

$$\sigma_i^2 = \frac{1}{d} \frac{\sum_{t=1}^{T} P(i|x_t) * ||x_t - \mu_i||^2}{\sum_{t=1}^{T} P(i|x_t)}$$

$$w_i = \frac{1}{T} \sum_{t=1}^{T} P(i|x_t)$$

where σ_i is the diagonal standard deviation.

The two steps are repeated until the likelihood score of the entire data set does not change substantially or a limit on the number of iterations is reached.

VI. k-means Clustering

K-means clustering algorithm is a well known algorithm which classifies a given data set through a certain number of clusters (= k clusters) fixed before-hand. The prime concept is defining k centroids corresponding to each of the clusters. These centroids are placed as far away from each other as possible. The next step is to taking each of the points belonging to the data-set and assign it to the nearest centroid based on the distance from each of the centroids. When no point is left, the first step is completed and an early grouping is done. Now, re-calculation of the k new centroids as barycenters of the clusters resulting from the previous step is done. After we have the k new centroids, a new binding has to be done between

the same data set points and the nearest new centroid. A loop is generated due to which we may notice that the k centroids change their location step by step until no more changes are done, i.e. the centroids do not move any more. Ultimately, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function is:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^{(j)} - c_j||^2,$$

where $||x_i^{(j)} - c_j||^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centres.

The algorithm is composed of the following steps:

1) Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2) Assign each object to the group that has the closest centroid.
3) When all objects have been assigned, recalculate the positions of the K centroids.
4) Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centres. The k-means algorithm can be run multiple times to reduce this effect. K-means is a simple algorithm that has been adapted to many problem domains.

VII. Experimental Setup

A. Brief description of the dataset

The dataset we have used is the IAM On-line English Handwritten Text Database (IAM-OnDB). We have 196 unique authors and for each of the writers, there are eight paragraphs of text, in which each paragraph of text consists of eight text lines on an average. Thus, it consists of a total of approximately 1,600 paragraphs consisting of around 11,170 text lines. A text line contains 627 points and 24 strokes on an average.

B. Setup for GMM-based system

We have trained models for three sets of different count of authors containing 5, 20 and 50 of them respectively. For each of the models, we have trained one each Gaussian Mixture Model for every author containing 50 components. 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:

1) Feature set 1: To construct this feature set, we have sampled 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 119.

2) Feature set 2: 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 143.

C. Setup for k-means based system

We have trained a model for a set of 50 authors. For each of the author, we have clustered his/her set of feature vectors into 50 groups. 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 the feature set 1 described above.

VIII. Results and Conclusion

A. Results for GMM-based system

1) Feature set 1: .

Number of Authors	Percentage Accuracy
5	1.00
20	0.80
50	0.64

2) Feature set 2: .

Number of Authors	Percentage Accuracy
5	1.00
20	0.75
50	0.60

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.

B. Results for k-means based system

We have already mentioned that we have evaluated k-means based system for set of 50 authors. The accuracy of this system turned out to be 20% which is much lesser than that compared to GMM-based system. Hence, we can conclude that GMM-based systems are able to model the feature sets of authors more accurately than simple clustering of the feature sets.

Acknowledgments

We are indebted to Dr. Rashmi Dutta Baruah for guiding us throughout the span of the semester and lending exhaustive insights into the intricacies of the project. We are also grateful to Mr. Gautam Singh, an undergraduate of IIT Guwahati for providing us useful guidance in feature extraction and k-means clustering based approach for writer identification. We are

also thankful to the teaching assistants of the Spring course Artificial Intelligence, CS561 for their guidance and keeping us updated with the timeline of the project.

REFERENCES

[1] A. Schlapbach, M. Liwicki and H. Bunke, *A Writer Identification System for On-line Whiteboard Data*, 2007.

[2] D. A. Reynolds, *Speaker identification and verification using Gaussian mixture speaker models, Speech Communication 17*, 1995.

[3] D. A. Reynolds, T. F. Quatieri, R. B. Dunn, *Speaker verification using adapted Gaussian mixture models, Digital Signal Processing 10*, 2000.

[4] M. Liwicki, A. Schlapbach, H. Bunke, S. Bengio, J. Mari'ethoz and J. Richiardi, *Writer identification for smart meeting room systems, in: Proc. 7th IAPR Workshop on Document Analysis Systems*, 2006.

[5] T. Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, R. Silverman and Angela Y. Wu, *An Efficient k-Means Clustering Algorithm: Analysis and Implementation*, 2002.