Comparison of HMM and SDTW for Tamil Handwritten Character Recognition

Shashikiran K, Kolli Sai Prasad, Rituraj Kunwar, A. G. Ramakrishnan
MILE Lab, Department of Electrical Engineering, IISc, Bangalore, India.
{shashi.reach, kollisp, kunwar.rituraj, agrkrish} @gmail.com

***Abstract*—In this paper, we compare the experimental results for Tamil online handwritten character recognition using HMM and Statistical Dynamic Time Warping (SDTW) as classifiers. HMM was used for a 156-class problem. Different feature sets and values for the HMM states & mixtures were tried and the best combination was found to be 16 states & 14 mixtures, giving an accuracy of 85%. The features used in this combination were retained and a SDTW model with 20 states and single Gaussian was used as classifier. Also, the symbol set was increased to include numerals, punctuation marks and special symbols like $, & and #, taking the number of classes to 188. It was found that, with a small addition to the feature set, this simple SDTW classifier performed on par with the more complicated HMM model, giving an accuracy of 84%. Mixture density estimation computations was reduced by 11 times. The recognition is writer independent, as the dataset used is quite large, with a variety of handwriting styles.**

I. Introduction

Human machine interface through the digital pen is becoming popular as an alternative to the keyboard with the rapid advancements in technology. Especially in hand held devices like PDA & high end mobile phones. The advent of digital pens have allowed easier interfacing for the user. In such a scenario, handwriting recognition plays a significant role in providing a natural interface for inputting text, as compared to keyboards. Handwriting recognition can be writer independent and writer dependent. A writer independent system is trained to recognize handwriting in a wide variety of writing styles, while the writer dependent system is trained to recognize handwriting of only one person.

In a country such as India, where a large section of the population still read and write in their native language, enabling interaction with computers in the native language and in a natural way allows for better technology penetration & greater inclusion of the masses.Thus arises the need for developing OHR systems for Indian Languages. One such very popular South Indian language is, Tamil. It is one of the oldest languages in the world.It is the official language of not only the Indian state of Tamil Nadu, but Also countries like Singapore, Malaysia and Sri Lanka. There are millions of speakers of this language all around the world. Given the importance & reach of the language a lot of research has been dedicated to technology development for Tamil language.

There are any number of classification techniques for this typical pattern recognition problem. Some of them are Suppot Vector Machines, Neural Networks, k-Nearest Neighbour and Dynamic Time Warping (DTW). In this work, we have experimented with Hidden Markov Models (HMM) and Statistical DTW (SDTW).

II. Survey of Tamil OHR

A. Tamil Script:

Tamil script basically has 12 vowels (V) and 23 consonants (C). The CV combinations come up to 12*23=276. Since some of the vowel modifiers are disjoint, stand-alone symbols, treating them as separate classes reduces the set of distinct symbols to 156, including all C-V combinations and the special character aydham. Most CV combinations are composed of two parts, namely the basic character and a modifier symbol. Some of the modifier symbols occur with an horizontal overlap with the base symbol and hence can be combined, the others are treated as separate symbols. Fig.1 presents the basic Tamil character set and the stand-alone vowel symbols. Also in Tamil, since there is no cursive writing each symbol can be considered as a separate class for recognition.

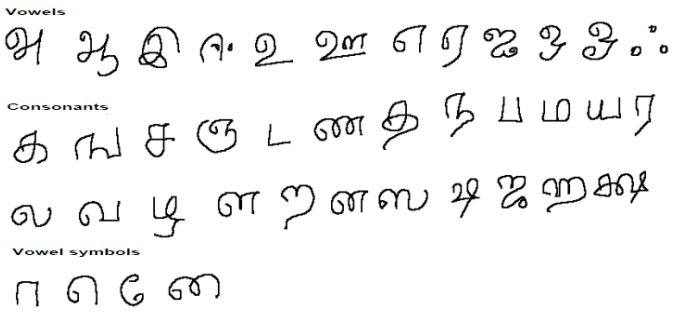

Fig. 1. Basic Tamil Character set

B. Status of Tamil OHR:

There are quite a few papers available on Tamil handwriting recognition. One of the oldest papers is by Sundaresan & Keerthi [6] which discusses feature selection based on a neural network classfier. The most common classification technique seen in the literature is DTW. The main drawback of this method is the computation time, which is proportional to

the number of prototypes. Vuurpijl et al. [4] discuss DTW along with a method of prototype learning to speed up the recognition. Similar discussion is also found in Niranjan Joshi et al. [5]. In both the cases the data sets were small. Prashanth et al. [7] have reported an accuracy of about 82% using DTW, on the HP Lab dataset. They too acknowledge the drawbacks of the DTW. Toselli et al. [8] as part of the HP Labs competition have reported 90% accuracy for the 156 classes. In this case, the data set is huge and a continuous HMM based on 16 gaussian mixtures per state, is used. None of the above works handle Indo-Arabic numerals or any other symbols.

H. Aparna et al. [3] discuss a "soft matching " technique for stroke recognition & then go on to recognize the character. They have included the Indo arabic numerals but punctuation marks and special symbols are not handled. In this work, for the SDTW classifier we have considered a complete data set consisting of Tamil characters, Indo-Arabic numerals, punctuation and special symbols.

III. Overview of the Recognition Process

The block diagram in Fig.2 gives an overall view of the various phases of the recognition process. The details of each phase is briefly explained in the following paragraphs.

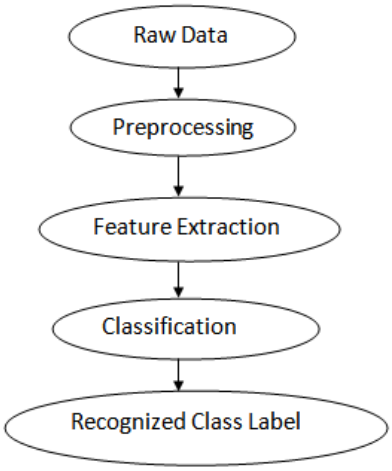

Fig. 2. Block diagram of OHR

A. Preprocessing:

It is the processing of raw data to make it more amenable for any use. It generally involves noise removal, re-sampling and normalization. Noise removal is elimination of duplicate points, since they do not contain any extra information and smoothing of strokes to remove any noise in the trajectory due to erratic pen motion. A Gaussian moving average filter was used for smoothing in our experiment. Re-sampling is done to bring about order in the raw data . Data with varying number of points can be brought to some pre determined fixed number of points, or the distance between consecutive data points can be made uniform irrespective of data sequence length. Normalization is required to compensate for the size and relative position of the data so that the patterns become comparable. In this work the raw data was resampled to 60 points and normalized to the range, 0 to 1.

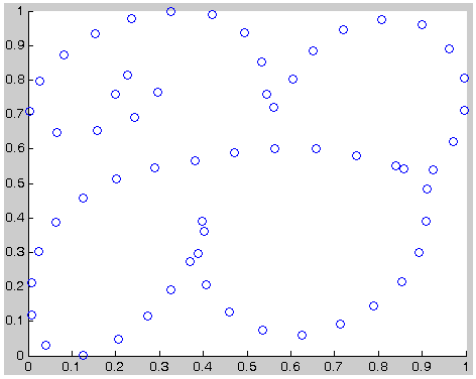

Fig. 3. Data after resampling and normalization.

B. Feature Extraction:

It is the process of extracting important characteristics of the data. Performance of any classifier depends on how well the features distinguish various data classes. The features used in this work are based on the local shape of the character and are explained below.

Preprocessed x-y co-ordinates: Using preprocessed data points as features, we retain the positional information of the original data sequence.

Pen direction angle: At each sample point, the direction of pen tip movement from that point to the next point can be used as a feature.

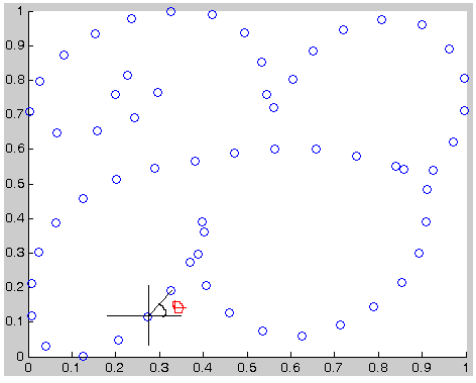

Fig. 4. Pen direction angle measurement

This can be calculated as follows. Shift the origin to the point at which, pen direction angle (PDA)= (θ_p) is to be calculated. Depending on where in the modified co-ordinate system, the next point lies we have the following cases
1st quadrant ,$(\theta_p) = (\theta)$, 2nd quadrant ,(θ_p)=$(\pi - \theta)$

3rd quadrant,(θ_p)= $(\pi + \theta)$,4th quadrant,(θ_p) = $(2\pi - \theta)$ where,

$$\theta = \tan^{-1}\frac{|y_2 - y_c|}{|x_2 - x_c|}$$

Normalized First Derivatives: The variations in the x and y coordinates are found independent of each other which give the shape features locally at each point. At the current point, a window is taken covering the past and the future points and the derivative is calculated using the formulae given below

$$X'(j) = \frac{\sum_{i=1}^{2} i*(x(j+i)-x(j-i))}{2*\sum_{i=1}^{2} i^2} \qquad Y'(j) = \frac{\sum_{i=1}^{2} i*(y(j+i)-y(j-i))}{2*\sum_{i=1}^{2} i^2}$$

$$X'_N = \frac{X'}{\sqrt{x'^2_j + y'^2_j}} \qquad Y'_N = \frac{Y'}{\sqrt{x'^2_j + y'^2_j}}$$

Normalized Second Derivatives: This is calculated by replacing X, Y co-ordinates by the raw derivatives calculated above, and then normalized.

$$X''(j) = \frac{\sum_{i=1}^{2} i*(x'(j+i)-x'(j-i))}{2*\sum_{i=1}^{2} i^2} \qquad Y''(j) = \frac{\sum_{i=1}^{2} i*(y'(j+i)-y'(j-i))}{2*\sum_{i=1}^{2} i^2}$$

$$X''_N = \frac{X''}{\sqrt{x''^2_j + y''^2_j}} \qquad Y''_N = \frac{Y''}{\sqrt{x''^2_j + y''^2_j}}$$

Curvature: Curvature at a point on a curve is the inverse of the radius of the osculating circle at that point. The latter is defined as the circle which forms the largest circular tangent to the curve on the concave side of the point. This curvature can be found using the first and second derivatives using the formula given below.

$$K = \frac{X'_N * Y''_N - X''_N * Y'_N}{(X'_N + Y'_N)^{\frac{3}{2}}}$$

Polynomial fit features: At each point, a neighborhood is taken and a polynomial curve is fit to the segment and its parameters are used as features. The parameters of the curve are found using the least square fit procedure. That is, the curve is fit so that the square of the difference between the original data and the value given by the equation are minimized.

C. Classification:

Classification is the heart of any recognition process. As mentioned earlier, there are a variety of techniques for classification. The techniques used in the current work are briefly explained below.

Continuous Hidden Morkov Models (HMM):

One of the recognition modules used is a continuous density Hidden Markov Model (HMMs [1]). An HMM is a stochastic finite-state device used to estimate the probability of a sequence of feature vectors, which characterize how a given handwritten character evolves in time. Thus each character class is modeled by a continuous HMM. Each HMM state is assumed to be represented by an adequate mixture of probability densities. Typically Gaussian mixture is assumed. The adequate number of densities in the mixture per state, and the number of HMM states, are empirically found. Once the number of states, transitions between them and number of densities per state is fixed, the model parameters can be easily trained from samples of handwritten characters. This training process is carried out using a well known variation of the EM algorithm called Baum-Welch re-estimation [1].

The recognition of an unknown test sample sequence of feature vectors is formulated as the problem of finding a HMM character class c, that maximizes the class-posterior probability, .i.e.,

$$c = \text{argmax } c \in C \; p(x|c,\theta)P(c)$$

Where C is the number of character classes, θ is the trained set of HMM parameters and P(c) is the a-priori probability of the character class c. The optimization problem can be solved using the well known Viterbi algorithm [1].

Statistical Dynamic Time Warping (SDTW):

In SDTW [2], a reference character is represented by a sequence Q = $(Q^1, Q^2, Q^3, ..., Q^n)$ of statistical quantities (state). These statistical quantities include

1) Discrete probabilities say $\alpha^j : \Omega \rightarrow [0,1]$ for statistical modeling of transitions $\Delta\phi\epsilon\Omega$ reaching the sequence's state *j*. i.e. state transition probabilities.

2) A continuous probability density function $\beta_j : R^d \longrightarrow R$ that models the feature distribution at sequence's state *j*. In our work we modeled β_j by a unimodal, multivariate Gaussian distribution.

While testing, the SDTW distance of test pattern to the reference model of each class is computed and the test pattern is assigned the label of the class giving minimum SDTW distance. The definition of SDTW distance is different from that of DTW distance.

The models in SDTW frame work are similar to HMMs of particular type with state prior probabilities π=$(1, 0, 0, ...0)^T$ and are of left to right models with step size of at most 1 and with null transitions (transitions that allow change in state without observation change i.e. transitions (0,1) in Ω). So the models in SDTW frame work can be trained by algorithms used for training HMMs. In our work, we used segmental K-means algorithm [1] for training SDTW model parameters.

IV. Dataset for our experiments

We have totally 188 classes composed of the 156 classes of distinct Tamil symbols as explained earlier, 10 Indo Arabic numerals, special symbols and punctuation marks totalling 22.

HP dataset was used for the 156 classes, the other symbols were collected in house from about 40 writers and 10 samples of each class. We used 250 files for training and 100 files for testing. Data were collected on Toshiba tablet PC using HP data collection tool.

V. Results

Initially, experiments were conducted on 156 classes [9]. First we experimented with a 10-state and 10-mixture continuous HMM using preprocessed X Y co-ordinates as features. This gave an accuracy of 60.3%. Various combinations of the features, states and mixtures were tried and found that the 16-state, 14-mixture model gave the best result of 85%. The results are given in Table 1.

We then included all the Indo arabic numerals, punctuation marks etc, and formulated SDTW as an 188-class problem. Experiments were conducted with two sets of features, and the number of states was empirically fixed at 20. Each state was modeled as a single Gaussian. The best accuracy obtained was 84%, which was comparable to the best HMM result.

Table.1 Results of HMM on 156 classes and SDTW on 188 classes, with different feature combinations

Classifier	Features used	Symbol Accuracy (%)
HMM (156 class)	Normalized X & Y states=10 mixtures=10	60.3
	Normalized X & Y, curvature and third order polynomial fit coefficients states=14 mixtures=14	78.5
	Normalized X & Y, Derivatives X & Y, Curvature states=16 mixtures=14	85.2
	Normalized X &Y, Derivatives X & Y, states=18 mixtures=18	81.9
SDTW (188 class)	Normalized X & Y, Derivatives X & Y, Pen Direction Angle states=20 mixtures=1	82.2
	Normalized X & Y, Derivatives X & Y, Second Derivatives X & Y Pen Direction Angle states=20 mixtures=1	83.9
	Normalized X & Y, Derivatives X & Y, Second Derivatives X & Y, Pen Direction Angle, Curvature states=20 mixtures=1	81.3

VI. Conclusion

We have compared the experimental results for HMM and SDTW classifiers on Tamil online handwritten characters. We found that results comparable to the best result of HMM could be achieved using a simpler SDTW model. The best HMM combination was 16 states and 14 mixtures, giving an accuracy of 85%, whereas the best SDTW with 20 states and single Gaussian, achieved 84%. Moreover, in the SDTW case the number of classes were higher(188) than the HMM case (156). The SDTW reduced the number of mixtures to be estimated from 16*14=224 to 20 only and handles 32 more symbols.

VII. Acknowledgment

We thank Mr. Suresh Sundaram, Ms.Nethra Nayak and Ms.Archana C.P. for their efforts which made these experiments possible. Special thanks to Technology Development for Indian Languages (TDIL), DIT, Govt. of India, for sponsoring this research.

References

[1] Lawrence R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition", Proc. of the IEEE, Vol.77, No.2, pp 257-286, February 1989.

[2] Clauss Bahlmann and Hans Burkhardt , "The writer independent online handwriting recognition system Frog On Hand and cluster generative statistical dynamic time warping", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.26, No. 3, pp 299-310, March 2004.

[3] KH. Aparna, V. Subramanian, M. Kasirajan, V. Prakash, V. Chakravarthy, and S. Madhvanath. Online handwriting recognition for Tamil. In F. Kimura and H. Fujisawa, editors, Proc. IWFHR9, pages 438-442, Tokyo, October 2004.

[4] R.Niels and L.Vuurpijl, "Using Dynamic Time Warping for Intuitive Handwriting Recognition", Proc. of 12th Conference of the International Graphonomics Society, Salerno, 2005, pp. 217-221..

[5] Niranjan Joshi , G. Sita , A. G. Ramakrishnan , Sriganesh Madhvanath, Comparison of elastic matching algorithms for online tamil handwritten character recognition, Proc ninth international workshop on frontiers in handwriting recognition, p.444-449, October 26-29, 2004

[6] C. S. Sundaresan and S. S. Keerthi, A study of representations for pen based handwriting recognition of Tamil characters, " Fifth International Conference on Document Analysis and Recognition, Sep. 1999, pp.422-425.

[7] L Prasanth, V Jagadeesh Babu, R Raghunath Sharma, G.V. Prabhakara Rao, Dinesh , Elastic Matching of Online Handwritten Tamil and Telugu Scripts Using Local Features,,HPL-2007-104

[8] Alejandro H. Toselli1 , Moiss Pastor2 and Enrique Vidal2 On-Line Handwriting Recognition System for Tamil Handwritten Characters, Pattern Recognition and Image Analysis, 2007

[9] Kolli Sai Prasad "Online Recognition of Tamil on Tablet PC using HMM" M.E Project Report, Dept. of EE, IISc, 2008.