Segmental K-Means Learning with Mixture Distribution for HMM Based Handwriting Recognition

Tapan Kumar Bhowmik, Jean-Paul van Oosten, and Lambert Schomaker

Faculty of Mathematics and Natural Sciences, University of Groningen, Netherlands
tkbhowmik@ai.rug.nl, J.P.van.Oosten@ai.rug.nl, L.Schomaker@ai.rug.nl

Abstract. This paper investigates the performance of hidden Markov models (HMMs) for handwriting recognition. The Segmental K-Means algorithm is used for updating the transition and observation probabilities, instead of the Baum-Welch algorithm. Observation probabilities are modelled as multi-variate Gaussian mixture distributions. A deterministic clustering technique is used to estimate the initial parameters of an HMM. Bayesian information criterion (BIC) is used to select the topology of the model. The wavelet transform is used to extract features from a grey-scale image, and avoids binarization of the image.

1 Introduction

Hidden Markov models (HMMs) are a common classification technique for time series and sequences in areas such as speech recognition, bio-informatics and handwriting recognition. HMMs are used to model processes which behave according to the Markov property: The next state is only influenced by the current state, not by the past.

Using mixture models as the observation probabilities has been proven to be very successful in handwriting recognition. Baum-Welch and Segmental K-means are algorithms for training HMMs [1,2]. Baum-Welch, although proven [3] to converge is not ideal: the convergence claim is proven theoretically, but empirically, there are still conditions where convergence does not occur, or computation restarts are otherwise needed. These restarts can greatly increase the, already lengthy, duration of training. As a first improvement, Segmental K-means are used with single Gaussian observation distributions. However, a simple example of a character image illustrates that distributions are multivariate. This paper proposes to use Segmental K-means with multi-variate Gaussian mixture observation distributions. Segmental K-means has the advantage over Baum-Welch that not every possible path is updated, only the most likely (Viterbi) path [2], which has a positive effect on computation time.

Because of its great importance, the initialisation of model parameters is investigated in this paper as well and a deterministic method is introduced to address this problem. Ideally, all segmentation is avoided, in the x-y space, but in luminance space as well. Therefore, algorithms are needed which can handle

S.O. Kuznetsov et al. (Eds.): PReMI 2011, LNCS 6744, pp. 432–439, 2011.

grey-scale. This is important in situations with low-quality images, such as in the Monk system [4] for historical manuscript retrieval.

The primary goal of this paper is not getting the best performance, but rather to see whether the combination of segmental K-means, the solution to the initialisation problem and grey-scale features is feasible for use in handwriting recognition.

2 Features

2.1 Feature Extraction with Wavelet Transform

The wavelet transform is a tool that finds application in many areas including image processing. Due to the multi-resolution property, it decomposes the signal at different scales. For a given image, the wavelet transform produces one low frequency subband image reflecting an approximation of the original image and three high frequency components of the image reflecting the detail. The approximation component is used here as a normalization image in the present recognition problem. In our experiment, we have considered Daubechies wavelet lowpass filter with four coefficients $[0.4830, 0.8365, 0.2241, -0.1294]$ [5]. For a raw input image we first calculate as much as possible the smallest rectangle object region of the image, then normalize it to a square image of size 64×64 with an interpolation technique. The wavelet decomposition algorithm with the above lowpass filter is applied to this normalized image to get 32×32 image. To make the pixel values with range $[0, 255]$, a scaling factor is used. The 32×32 scaled image is again divided into 16 blocks each of size 8×8. The 64 pixel values of each block are considered as the initial feature vector. This pipeline is shown in Fig. 1 for the digit "0" (zero).

(a) (b) (c) (d)

Fig. 1. (a) Original image of digit "0" (b) 64×64 normalized pixel image (c) 32×32 wavelet-decomposed image and (d) corresponding image 4×4 grid of 8×8 wavelet-values

In this paper, the goal is to exploit the good character representation with the wavelet feature within a hidden Markov scheme. Therefore, a virtual time axis needs to be constructed. An observation sequence is obtained by moving through the image from top to bottom, from left to right. This order is arbitrary, but a good first approach with regards to making segments.

2.2 Feature Reduction with PCA

In mathematical terms, feature reduction problem can be stated as: given the p-dimensional random variable $\mathbf{x} = (x_1, x_2, ...x_p)^T$, find a lower dimensional representation of it, $\mathbf{y} = (y_1, y_2, ...y_d)^T$ with $d \leq p$, that captures content in the original data, according to some criteria. Let us assume that we have n observations, each being a realization of the p-dimensional random variable $\mathbf{x} = (x_1, x_2, ...x_p)^T$ with mean $\mu = \mathbf{E}(\mathbf{x}) = (\mu_1, \mu_2, ...\mu_p)^T$, $\mu_i = \frac{1}{n}\sum x_i$ and covariance matrix $\Sigma_{p\times p} = E(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T = \frac{1}{n}\sum (x_i - \mu_i)(x_i - \mu_i)^T$ where $i = 1, 2, ..., p$. Let $\mathbf{\Phi_i} = (\Phi_{i1}, \Phi_{i2}, ...\Phi_{ip})^T$ be the eigen vector corresponding to eigen value λ_i of $\Sigma_{p\times p}$, where $i = 1, 2, ..., p$, $\lambda_i \geq \lambda_j$ for all $i < j$ and all the eigen vectors are mutually orthogonal for different λ_i. The Karhunen-Loeve linear transform on the basis of new coordinate system whose axis along $\{\mathbf{\Phi_i}\}$ is defined as $y_i = \Phi_{i1}x_1 + \Phi_{i2}x_2 + ... + \Phi_{ip}x_p$, $i = 1, 2, ..., d(d \leq p)$, where y_i is the i^{th} principal component. Since a lower component is less significant than a higher one, depending upon the value of λ_i, lower components $y_i(i > d)$ can be removed to reduce the dimension of features.

In this experiment for each block out of $8 \times 8 = 64$ features, the 16 most significant components are chosen for the actual feature vector. Since number of blocks is 16 (4×4), the total number of feature vectors is 16, each with a size of 16 features. In our HMM framework each feature vector is considered as an observation symbol (O_i). So for an input image we get an observation sequence $\mathbf{O} = O_1, O_2, ..., O_{16}$ with each observation having 16 dimensions.

3 HMM Parameter Estimation

An HMM consists of three sets of parameters $\lambda = (\pi, A, B)$, where π is the initial state probability distribution, A is the state transition probability distribution matrix and B is the observation symbol probability distribution. The modeling of handwritten characters with HMMs can be regarded as the estimation of model parameters λ_c for each character class c which represents almost the characteristic of input sequence of data of that particular class. The estimation of model parameters involves mainly three steps: selection of topology, estimation of initial parameters of model λ, based on topology and finally re-estimate λ in such a way that it maximizes the probability $P(O|\lambda)$ i.e., $P(O|\hat{\lambda}) \geq P(O|\lambda)$, where $\hat{\lambda}$ is the re-estimated model.

3.1 Topology Selection and Initial Parameter Estimation

Before going to estimate the model parameters we need to define the topology of the model which means the number of states, the number of mixture component per state (if observation probability is realized from Gaussian mixture distributions) and the transition between states for the model. Here we always consider ergodic (fully connected states transition) topology for transition between states. The number of states and the number of mixture component per

state are chosen randomly from a range of possible values of number of state and number of mixture component. Let N be the number of states and M the number of mixture components per state. Also assume that Q_t is a random variable with N possible values $\{1, 2, ...N\}$, representing a discrete state. Then we can define $A = \{a_{ij}\} = \{P(Q_t = j/Q_{t-1} = i)\}$, a hidden and time independent stochastic transition matrix and $\pi = \{\pi_i\} = \{P(Q_{t=1} = i)\}$, the probability of being in state i at time $t = 1$. Let a particular observation sequence be described as $O = (O_1 = o_1, ..., O_T = o_T)$. The probability of a particular observation at a particular time t for state j be described by: $b_j(o_t) = P(O_t = o_t/Q_t = j)$. The complete collection of parameters for all observation distributions is represented by $B = \{b_j(.)\}$. Let us assume that probability of observation for a particular state is emitted from a Gaussian mixture distribution. So we can write $b_j(o_t) = \sum_{k=1}^{M} c_{jk}\mathcal{N}(o_t|\mu_{jk}, \Sigma_{jk}) = \sum_{k=1}^{M} c_{jk}b_{jk}(o_t)$, where M is the number of components in the mixture.

Initialization of Mixtures through PCA based K-means Clustering[6]: Let $X = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\}$ be the set of n observations, each of dimension d. For initializing N mixtures with each having M components, the data X has been partitioned into $K(= NM)$ clusters $\{C_{11}, C_{12}, ..., C_{1M}, C_{21}, ..., C_{NM}\}$. The goal of the K-means clustering is to find an exclusive partition in such a way that the criterion value $SSE = \sum_{j=1}^{N}\sum_{k=1}^{M}\sum_{\mathbf{x_i}\in C_{jk}} \|\mathbf{x_i} - \mu_{jk}\|$ (where $\mu_{jk} = \frac{1}{n_{jk}}\sum_{\mathbf{x_i}\in C_{jk}} \mathbf{x_i}$ denotes the mean of cluster C_{jk} and n_{jk} denotes the number of instances in C_{jk}) is minimized. Since the first principal direction (the eigen vector corresponding to the largest eigen value of the covariance matrix $\Sigma_{d\times d}$) is the direction which contributes the most to the SSE, it is a good candidate direction for splitting the cluster. Starting from a single cluster, divide it into two sub-clusters, choose the sub-cluster with the largest within-cluster SSE_{jk} as the next cluster to partition, repeat the process until K clusters are produced. At each split stage, for the selected cluster C_{jk}, we divide it into two sub-clusters $C_{jk}^{(1)}$ and $C_{jk}^{(2)}$ according to the following rule: for any $\mathbf{x_i} \in C_{jk}$, if y_i (the projected value of $\mathbf{x_i}$ in first principle direction of $\mathbf{x_i} \in C_{jk}$) $\leq \bar{y}_i$ (mean of y_i), assign $\mathbf{x_i}$ to $C_{jk}^{(1)}$, otherwise, assign $\mathbf{x_i}$ to $C_{jk}^{(2)}$. The mean value of each cluster is considered as the initial cluster center. Once we get the initial cluster centers, run K-means clustering to get the final clusters. For each cluster C_{jk} we then calculate mean μ_{jk} and covariance matrix Σ_{jk}. And c_{jk} has been calculated by $c_{jk} = \frac{n_{jk}}{\sum_k n_{jk}}$. These are the initial parameters for the mixtures.

Initialization of initial state probability and state transition probability: Initial state probabilities and state transition probabilities have been chosen randomly in such a way that they satisfy the following criteria: $\sum_i \pi_i = 1$ and $\sum_j a_{ij} = 1$.

3.2 Re-estimation of Model Parameter through Segmental K-Means

We now define the following variables before describing the Segmental K-Means algorithm:

$$\alpha_t(i) = P(O_1 = o_1, ..., O_t = o_t, Q_t = i|\lambda), \tag{1}$$

is the probability of seeing the partial observation sequence, $o_1, o_2, ..., o_t$, (until time t) and ending up in state i at time t, given the model λ.

$$\beta_t(i) = P(O_{t+1} = o_{t+1}, ..., O_T = o_T|Q_t = i, \lambda), \tag{2}$$

is the probability of the ending partial observation sequence, $o_{t+1}, o_{t+2}, ..., o_T$ given that we started at state i at time t and the model λ.

$$\xi_t(i,j) = \frac{P(Q_t = i, Q_{t+1} = j|O, \lambda)}{P(O|\lambda)} = \frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_i^N \sum_j^N \alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}, \tag{3}$$

is the probability of being in state i at time t and being state j at time $t+1$, given the model λ and observation sequence O.

$$\gamma_t(i) = P(Q_t = i|O, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^N \alpha_t(j)\beta_t(j)} = \sum_{j=1}^N \xi_t(i,j), \tag{4}$$

is the probability of being in state i at time t, given the observation sequence O, and the model λ.

Segmental K-Means Algorithm:

The Baum-Welch algorithm defines $\xi_t(i,j)$ for all states i and j. In contrast, the segmental k-means algorithm finds the most likely state sequence using the Viterbi algorithm and calculates $\xi_t(i,j)$ only over the states found in the most likely state sequence. The algorithm works as follows:

For an input observation sequence $\mathbf{o} = (o_1, o_2, ..., o_T)$

1. Calculate $\alpha_t(i)$ and $\beta_t(i)$ through the Forward-Backward algorithm, which is described explicitly in [1]. To prevent rounding errors in implementations, $\alpha_t(i)$ and $\beta_t(i)$ need to be scaled, as discussed in [7].
2. Estimate $\xi_t(i,j)$
   Find the optimal state sequence for the input sequence $\mathbf{o}$ with the Viterbi algorithm [1] and then assign each observation o_t to a particular state according to the Viterbi state sequence. Then,

   $$\xi_t(i,j) = \begin{cases} 1 \text{ if observation } o_t \text{ assigned to state } i \text{ and } o_{t+1} \text{ assigned to state } j \\ 0 \text{ otherwise} \end{cases}$$

3. Estimate $\gamma_t(i)$ (the probability of being in state i at time t) from $\xi_t(i,j)$ (the probability of being in state i at time t and state j at time $t+1$), according to equation (4) and in particular, $\gamma_T(i) = \sum_{j=1}^N \xi_{T-1}(i,j)$.

4. Update the model parameters

$$\bar{\pi_i} = \gamma_1(i), \tag{5}$$

the expected relative frequency spent in state i at time $t = 1$.

$$\bar{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \tag{6}$$

the expected number of transitions from state i to state j relative to the expected total number of transitions away from state i. We now define the probability that the k^{th} component of the i^{th} mixture generated observation o_t as $\gamma_t(i,k) = \gamma_t(i)\frac{c_{ik}b_{ik}(o_t)}{b_i(o_t)}$, then

$$\bar{c_{ik}} = \frac{\sum_{t=1}^{T} \gamma_t(i,k)}{\sum_{t=1}^{T} \gamma_t(i)} \tag{7}$$

$$\bar{\mu_{ik}} = \frac{\sum_{t=1}^{T} \gamma_t(i,k) o_t}{\sum_{t=1}^{T} \gamma_t(i,k)} \tag{8}$$

$$\bar{\Sigma_{ik}} = \frac{\sum_{t=1}^{T} \gamma_t(i,k)(o_t - \bar{\mu_{ik}})(o_t - \bar{\mu_{ik}})^T}{\sum_{t=1}^{T} \gamma_t(i,k)} \tag{9}$$

In case of multiple observation sequences, let us assume that if L be the number of sequences with T_l be the length of the sequence l, then the equations above become:

$$\bar{\pi_i} = \frac{\sum_{l=1}^{L} \gamma_1^l(i)}{L}$$

$$\bar{a_{ij}} = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T_l-1} \xi_t^l(i,j)}{\sum_{l=1}^{L} \sum_{t=1}^{T_l-1} \gamma_t^l(i)}$$

$$\bar{c_{ik}} = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T_l} \gamma_t^l(i,k)}{\sum_{l=1}^{L} \sum_{t=1}^{T_l} \gamma_t^l(i)}$$

$$\bar{\mu_{ik}} = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T_l} \gamma_t^l(i,k) o_t^l}{\sum_{l=1}^{L} \sum_{t=1}^{T_l} \gamma_t^l(i,k)}$$

$$\bar{\Sigma_{ik}} = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T_l} \gamma_t^l(i,k)(o_t^l - \bar{\mu_{ik}})(o_t^l - \bar{\mu_{ik}})^T}{\sum_{l=1}^{L} \sum_{t=1}^{T_l} \gamma_t^l(i,k)}$$

From Step-1 to Step-4 is repeated until any observation o_t is reassigned a new state in Step-2. Step 1 to step 4 is repeated as long as any observation is reassigned to a new state in step 2.

4 Experimental Results

The following experiments are performed on three datasets. The first two are handwritten Bangla numerals and basic characters, the third is a MNIST dataset of 70 000 handwritten arabic numerals. The Bangla numerals dataset consists of 10 000 training instances and 2 000 instances for testing. The Bangla basic characters dataset has 45 classes and the number of training instances are 500 per class, and 100 for testing. Finally, the MNIST dataset is used as well, with 60 000 training instances and 10 000 testing instances. The following accuracies can be reported on the previously mentioned test sets, using the selected models as described in the Model Selection section: 96.25% for Bangla numerals, 80.00% for Bangla basic characters and 97.19% for MNIST dataset. The accuracy on same dataset of basic characters is relatively better than the accuracy is found previously on single stage classification with SVM classifiers [5].

4.1 Model Selection

An important but difficult task is to choose a relevant number of components or a number of hidden states in the HMM when observation probabilities are emitted from a finite mixture distribution. Many criteria (BIC, ICL, PML, etc.) or procedures [8] have been proposed to answer this open question. But there is no guarantee that any particular criterion will work satisfactory on a particular dataset. In this study we have used BIC criterion[9] for selecting the number of states and the number of mixture per state in HMM. If the number of state in the model is N and the number of components in the mixture is M, then according to BIC criterion $BIC(\lambda) = logP(X|\hat{\lambda}) - \frac{1}{2}\nu(\hat{\lambda})log(n)$, where $\nu(\hat{\lambda})$ is the number of free parameters in the model and n is the size of the observation set X generating model $\{\lambda\}$ and $\hat{\lambda}$ is the Maximum Likelihood estimate of the model λ. Here $\nu(\hat{\lambda}) = N(M\frac{d(d+1)}{2} + Md + (M-1)) + (N^2-1)$, d is the dimension of the observation. The first term $M\frac{d(d+1)}{2}$ is the number of free parameters for covariance matrix of a mixture, the second term Md is for mean, the third term $(M-1)$ is for the coefficient of each mixture and the last term (N^2-1) is for state probabilities (in the context of ergodic model).

As an example, Fig. 2 shows the BIC values for different numbers of states and mixture components for the model of the Bangla numeral zero and Fig. 3 shows the corresponding model accuracy in the Bangla numerals test set when other competitive models (except zero) are kept constant. From this figure, we can see that the first maximum value of the BIC curve always gives the highest accuracy. In this case, the models $\lambda(4,8)$, $\lambda(6,5)$ and $\lambda(8,4)$ are the models with the highest accuracy, on the basis of the first maxima point of the BIC curve. Since $\lambda(4,8)$ produces the maximal BIC value with respect to other two models ($\lambda(6,5)$ and $\lambda(8,4)$), $\lambda(4,8)$ has been chosen as the final model for the numeral zero.

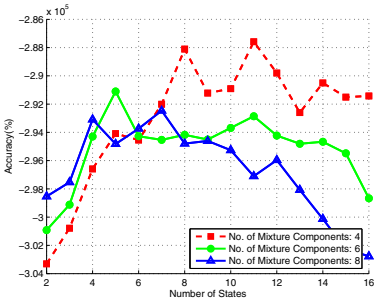

Fig. 2. BIC for different number of states

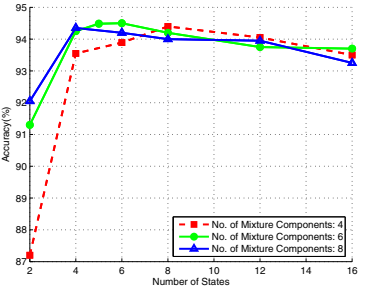

Fig. 3. Accuracy for different number of states

5 Conclusion

The learning framework of the Segmental K-Means algorithm for HMMs with Gaussian mixture observation densities has been described. A deterministic method to initialize the model parameter has been presented as well. The main advantage of this framework is that it can easily be switched to the Baum-Welch learning algorithm, which is used conventionally to learn the HMM parameters. The only change in the Segmental K-Means algorithm is step 3. A grey-scale feature has been implemented for recognition of isolated handwritten characters. Initial recognition results show that using the segmental k-means learning method for HMMs is quite efficient for handwriting recognition.

References

1. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Trans. on. IEEE 77(2), 257–286 (1989)
2. Juang, B.H., Rabiner, L.: The segmental k-means algorithm for estimating parameters of hidden markov models. IEEE Trans. on ASSP 38(9), 1639–1641 (1990)
3. Wu, C.F.J.: On the convergence properties of the EM algorithm. The Annals of Statistics 11(1), 95–103 (1983)
4. van der Zant, T., Schomaker, L., Haak, K.: Handwritten-word spotting using biologically inspired features. IEEE Trans. on PAMI 30(11), 1945–1957 (2008)
5. Bhowmik, T.K., Ghanty, P., Roy, A., Parui, S.K.: Svm-based hierarchical architectures for handwritten bangla character recognition. International Journal on Document Analysis and Recognition(IJDAR) 12(2), 97–108 (2009)
6. Su, T., Dy, J.: A deterministic method for initializing k-means clustering. In: Proc. of The 16th IEEE Int. Conf. on Tools with Artificial Intelligence, pp. 784–786 (2004)
7. Rahimi, A.: An erratum for a tutorial on hidden markov models and selected applications in speech recognition. In: Online article (2000)
8. Celeux, G., Durand, J.B.: Selecting hidden markov model state number with cross-validated likelihood. Computational Statistics 23(4), 541–564 (2008)
9. Biem, A.: A model selection criterion for classification: Application to hmm topology optimization. In: Proceedings of the 7th ICDAR, pp. 104–108 (2003)