INITIALIZATION OF HIDDEN MARKOV MODELS FOR UNCONSTRAINED ON-LINE HANDWRITING RECOGNITION

Krishna Nathan, Andrew Senior and Jayashree Subrahmonia

IBM T.J. Watson Research Center, P.O. Box 218,
Yorktown Heights, NY 10598, USA.

ABSTRACT

In a hidden Markov model system, the initialization of the model parameters is critical to the performance of the model after retraining. This paper proposes a number of new approaches to the problem of initialization, and demonstrates that a method of *smooth alignment* results in the best performance.

1. INTRODUCTION

Hidden Markov models (HMMs) have been used effectively for modeling continuous speech data [1, 2]. The rationale behind using them for modeling handwritten data is the similarity between continuous speech and on-line cursive handwriting recognition [5, 4, 6]. Figure 2 shows a simplified schema for parameter re-estimation with the expectation-maximization (EM) algorithm. Parameters are initialized, and the resultant model used to align data with a label. The alignment can be used to estimate better parameters and the procedure iterated, converging to a local optimum [3]. Parameter initialization is an important issue. Improperly initialized models lead to poor alignments in the training phase, thus resulting in bad estimates of the HMM parameters and convergence to a local optimum significantly worse than the global optimum. This paper describes a number of new approaches to the initialization. These approaches are applicable to any problem that uses the above estimation framework.

Figure 1 shows a typical letter model. The model consists of n states. Associated with each of the states, $s_1, s_2, \ldots, s_n$, is a set of three transitions: null, next and self. The self and next transitions result in the emission of an observation feature vector. The number of states m, the state transition probabilities, $p(\{\mathsf{self}, \mathsf{next}, \mathsf{null}\}\,|\,s_i)$ and the output probability distributions, $p(x_t|s_i)$, assumed tied across next and self transitions, completely specify the model [4].

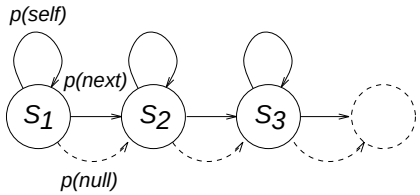

Figure 1: A hidden Markov model for one character.

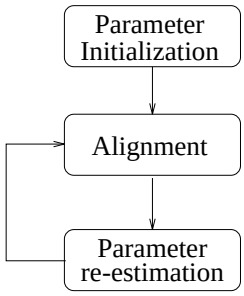

Figure 2: The schema for EM parameter re-estimation.

The HMMs use a mixture of shared Gaussian distributions as output distributions. Instead of estimating full covariance matrices we choose to approximate them with diagonal matrices. The parameters to be initialized are the means and covariances of the distributions used to represent the feature space, the mixture coefficients for each state and the state transition probabilities. We

assume that all observations start at the left-most state and end at the right-most state, so we do not estimate an initial state distribution. The means and covariances as well as the number of distributions are initialized by K-means clustering of all the input frames. In the remainder of this paper, we will assume that this yields sufficiently good initial cluster estimates. Though the parameters of the Gaussians are re-estimated, we concentrate on the task of improving the estimates of the mixture coefficients for each state and the state transition probabilities.

2. INITIALIZATION TECHNIQUES

In some HMM systems it is possible to initialize the parameters by hand-aligning the data to the model, *i.e.* by manually associating each frame with a state. This is easy to do if each state of the HMM has a significance that can be readily deduced from the observed data. In our system, each state models part of a character and given the large variability in instances of the character it is hard to associate a data point with a state with a high degree of certainty. Furthermore, different instances of identical characters often result in different number of frames rendering hand alignment difficult.

The initialization schemes that we will describe fall into two broad categories. In the first, the number of states in the model is fixed and observations of varying lengths are aligned to it. The differences in the approaches lie in the method used to choose this alignment. Given the alignments, it is straightforward to estimate the necessary parameters. In the second category, multiple model lengths are used to match the durations of the observation sequences. Alignment is trivial but the models need to be merged into one model representative of all the observation sequences for a character.

3. FIXED LENGTH INITIALIZATION

In all these methods the number of states in the model is chosen to be the mode of the length of the observations for that model.

3.1. Single-state replication

The mixture coefficients are identical for each state and are obtained by training a degenerate mixture model for each letter, consisting of a single state. The transition probabilities are assumed to be uniform *i.e.* the self, next and null transition probabilities are initialized to $1/3$. No alignment is necessary since neither the mixture coefficients nor the transition probabilities are re-estimated.

This method suffers from the following drawback: in aligning a character to a model during retraining, the data likelihoods for all alignments are equal, since the distributions are identical for all states; the only distinction between alignments is their duration probabilities. If the next transition is the most likely in each state, aligning m frames to n states will put one frame in each state if $m = n$. However, if $m > n$, the extra $m - n$ frames will be bunched in the first or final state of the model, depending on the implementation of the algorithm. Similarly, if $m < n$, the null transitions will all occur at one end of the model. This will produce poor models from which it is hard to recover with further retraining.

3.2. Mode-length alignment

Only observation sequences that are of the same length as the model are aligned. The alignment is simply that which takes the next transition at every frame. This alignment is used to calculate a new estimate of the mixture coefficients. Since only next transitions are observed, we cannot estimate the state transition probabilities and they are chosen to be uniform, *i.e.* $p(\mathsf{self}|s_i) = p(\mathsf{next}|s_i) = p(\mathsf{null}|s_i) = \frac{1}{3} \forall i$.

3.3. Random alignment

Mode-length alignment solves the problem of bunching, but at the cost of wasting much of the data in the initial estimation. A solution to this problem is to assign the frames of sequences of arbitrary length evenly to the states of a fixed-length model. Thus if there are $m = n.c + r$ frames, with $0 \leq r < n$ and $0 \leq c$, each state will be assigned c frames, and the extra r frames are randomly as-

signed among the n states. The time ordering is of course preserved, so the first state is assigned the first c or $c+1$ frames, the second state the next c or $c+1$ and so on. This frame-state alignment gives the positions of the self, next and null transitions, and from their relative frequencies in each state, we can estimate the state transition probabilities.

3.4. Smooth alignment

If we now consider the case where we repeatedly align the data by the *random alignment* procedure, sometimes a frame on a state boundary will be assigned to one state, and some times to another. This is equivalent to assigning a probabilistic count to each frame — one for frames that are consistently assigned to the same state, but less than one for frames which can fall on either side of a state boundary. We now show that it is possible to calculate these probabilistic counts explicitly to give a soft alignment by inspection, equivalent to the limit of large numbers of repeated presentations to the *random alignment* algorithm.

Suppose that there are m frames to be fitted to n states. Consider first the case $n > m$.

We assume that no self transitions will be taken, so there must be $n-m$ null transitions. The transition counts for self, next and null transitions are incremented by 0, $\frac{m}{n}$ and $\frac{n-m}{n}$ respectively.

Now consider the probability of frame j being in state k. This requires there to have been $j-1$ non-null transitions in the previous $k-1$ states. There are C_{j-1}^{k-1} ways of doing this, where C_m^n denotes the combination $\frac{n!}{m!(n-m)!}$. Similarly there are C_{m-j}^{n-k} ways of putting the remaining $m-j$ non-nulls in the remaining $n-k$ states. The total ways of arranging m non-nulls in n states is C_m^n, so

$$p(j\text{th frame in } k\text{th state}) = \frac{C_{j-1}^{k-1} C_{m-j}^{n-k}}{C_m^n}. \qquad (1)$$

Now, when $n < m$, the transition counts are incremented by $\frac{n-m}{m}$, 1 and 0 respectively.

To calculate the frame-state correspondences, we must calculate the number of ways of putting $m-n$ self transitions among a total of m transitions, knowing that the last transition must be a next. To put frame k in state j, we must have $k-j$ selfs in the first $k-1$ transitions and $(m-n)-(k-j)$ in the remaining $m-k$ transitions. We do not care what sort of transition the kth is — self or next, the frame still goes to place j. Note that $m-k+1-(m-n-k+j) = (n-j+1)$ so $C_{m-n-k+j}^{m-k+1} = C_{n-j+1}^{m-k+1}$. Thus

$$p(j\text{th frame in } k\text{th state}) = \frac{C_{k-j}^{k-1} C_{n-j}^{m-k}}{C_{n-1}^{m-1}}. \qquad (2)$$

This alignment gives us probabilistic frame-state alignments in the manner of those generated by the forward-backward algorithm. The counts generated by the above method are used in the conventional manner to estimate the parameters of the model [1].

4. VARIABLE LENGTH INITIALIZATION

This scheme uses several models of different lengths to initialize the model for each character. We start with a set of M HMMs for each character, where M is the number of different lengths of observation sequences for that character. Each observation sequence is aligned to the HMM of the same length with a next transition being taken for each frame. This is equivalent to the *mode-length alignment* above. We model all the frames aligned to a state as a Gaussian distribution whose mean and covariance we estimate. This gives a set of n_{sum} Gaussian distributions, where n_{sum} is the total number of states in the M HMMs. We then bottom-up cluster the n_{sum} Gaussians to r clusters using a trace criterion and initialize the final model as follows. For each state s in the HMM of mode length, n, we find the cluster C_s in which the Gaussian distribution for that state lies. The mixture coefficients for state s are then computed as the weighted average of the mixture coefficients for all the states whose Gaussians are in C_s. The transition probabilities can be computed from the relative frequencies of the models of different lengths — shorter models implying some null transitions, and longer models implying self transitions.

5. RESULTS AND DISCUSSION

We present recognition results for a writer independent system after initialization using the different schemes discussed above. Table 5 shows word error rates for the initial models without subsequent retraining. The size of the lexicon is 22,000 words. The recognizer was trained with 75,000 characters from approximately 100 writers. Subjects were told to write in their natural style on an LCD tablet on which the path of the pen was displayed. The test set consisted of isolated words written by authors not represented in the training set. Since we were only interested in the character models we did not use any language modeling.

Mode-length alignment does not do as well as the other techniques. This can be explained by the fact that much of the data is unused and because the transition probabilities are not trained. *Smooth alignment* performs better than *random alignment*, reducing the error rate by six percent. Although they can be considered equivalent for repeated presentations of data, *smooth alignment* makes more efficient use of a single presentation. The variable length initialization gives the next best performance (two percent worse), and is more computationally expensive, however it provides some insight into changing the HMM topology, by means of a mechanism to identify like states and to merge models.

Initialization scheme	**Detailed match error rate (%)**
Fixed Length	
Mode-length	25.2
Random	23.9
Smooth	22.6
Variable Length	23.1

Table 1: Error rates for models initialized by different methods on the same multi-writer data.

6. REFERENCES

[1] L. R. Bahl, F. Jelinek, and R. L. Mercer. A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):179–190, March 1983.

[2] J. R. Bellegarda and D. Nahamoo. Tied mixture continuous parameter modeling for speech recognition. *IEEE ASSP Magazine*, 38, December 1990.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B*, 39(1):1–38, January 1977.

[4] K. S. Nathan, H. S. M. Beigi, J. Subrahmonia, G. J. Clary, and H. Maruyama. Real-time on-line unconstrained handwriting recognition using statistical methods. In *ICASSP95*, volume 4, pages 2619–2623, 1995.

[5] K. S. Nathan, J. R. Bellegarda, D. Nahamoo, and E. J. Bellegarda. On-line handwriting recognition using continuous parameter hidden markov models. In *ICASSP93*, volume 5, pages 121–124, 1993.

[6] T. Starner, J. Makhoul, R. Schwartz, and G. Chou. On-line cursive handwriting recognition using speech recognition techniques. In *International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 125–128, 1994.