Learning HMM Structure for On-line Handwriting Modelization

Henri Binsztok, Thierry Artières
LIP6 / Université Paris VI
8, rue du Capitaine Scott
75015 Paris - France
Henri.Binsztok@lip6.fr, Thierry.Artieres@lip6.fr

Abstract

We present an Hidden Markov Model-based approach to model on-line handwriting sequences. This problem is addressed in term of learning both Hidden Markov Models(HMM) structure and parameters from data. We iteratively simplify an initial HMM that consists in a mixture of as many left-right HMM as training sequences. There are two main applications of our approach: allograph identification and classification. We provide experimental results on these two different tasks.

Keywords: HMM Structure Learning, Allograph Clustering

1 Introduction

This paper deals with on-line handwriting signals clustering and Hidden Markov Models (HMM) structure learning. These two problems are closely related and are of great interest in the field of on-line handwriting processing and recognition. Clustering on-line signals is useful for determining allographs automatically, identifying writing styles, discovering new handwritten shapes, etc. HMM structure learning may help to automatically handle allographs when designing an on-line handwriting recognition system. Learning HMM models involves learning the structure of the model (topology) and learning the parameters of the model. Usually, learning consists in first choosing a structure *a priori* then in automatic learning of model parameters from training data with EM optimization. Learning of model structure is then implicitly performed manually through successive trials.

Fundamentally, we seek to develop learning algorithms for Markovian systems that focus on the learning of mixture models for typical writing styles, it is then very close to clustering. Such techniques were studied in speech recognition. [7] proposes an algorithm that uses probabilistic grammatical inferences techniques, which specifically addresses speech variability. A few techniques have been proposed for related tasks within the Handwriting Recognition community, e.g. automatic identification of writing styles, writer identification. For example, [9] proposes a probabilistic approach to define clusters: For each handwritten character, an approach is used to learn the probabilities that a character belongs to a given cluster. The use of HMM for clustering handwritten characters was approached by [11], but their approach depends on initialization so that some supervised information is needed to achieve good performance. Also, [18] proposes an interesting hierarchical approach. Besides, more generic approaches have been proposed for sequence clustering, for example [15] provides an algorithm to cluster sequences into a predefined number of clusters, along with a preliminary method to find the numbers of clusters through cross-validation using a Monte Carlo measure. This theoretical approach relies on iterative reestimation of parameters via an instance of the EM algorithm, which requires careful initialization. Furthermore, the structure of the model is limited to a mixture model of fixed-length left-right HMM, which may not model correctly sequences of varying lengths in the data.

Since we are interested in both HMM structure learning and on-line handwritten signals clustering we chose to tackle the problem from the HMM structure learning point of view. We present our HMM learning algorithm and its application to on-line handwritten sequence clustering in section 2. In section 3, we describe a few experimental studies on digit clustering and classification. In the following, we consider on-line signals are represented as sequences of elementary strokes (close to direction codes very popular for Asian character recognition) so that we deal with discrete HMM only. We will briefly describe in section 3.1 how this preprocessing is performed.

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 IEEE

2 HMM learning algorithm

Our goal is to define a learning algorithm for HMM that meets two requirements. First, the resulting model should describe well the training data. Second, the model should allow to identify sets of similar sequences coresponding to allographs or writing styles. Most approaches to HMM structure learning suggest to start by building a complex initial model and simplifying it iteratively [10, 3, 17]. In [3], the simplification is based on entropic prior probabilities of the transitions between states, and some transition probabilities converge towards 0, thus simplifying the structure of the model. In [17], pair of states from the initial HMM are merged iteratively as long as the loss of likelihood is not too significant. Both approaches, being generic, meet the first requirement but not the second.

We chose to adapt the approach of [17] to our goals by restricting the HMM to belong to the class of mixture of left-right HMM where each left-right HMM of the mixture corresponds to a cluster. This global HMM is then iteratively simplified by removing a left-right HMM. This ensures that at any step, the global HMM belongs to the class of mixtures of left-right HMM, which allow performing clustering. We now present our unsupervised learning algorithm. First, we detail the building of an initial HMM (section 2.1). Then, we describe the iterative simplification algorithm applied to this initial model (section 2.2). The simplification algorithm relies on a distance between left-right HMM that we present in section 2.3. In term of clustering, the initial model may find as many clusters as sequences in the data and further models will provide less clusters.

2.1 Building an initial HMM from data

This first stage consists in building an HMM M_0 (a mixture of left-right HMMs) summarizing the whole training sequences. Let $D = (s_1 \dots s_n)$ be these sequences. Each input sequence s_i of length l_i is a sequence of strokes $s_i = (\sigma_{i,1} \dots \sigma_{i,l_i})$ where each stroke $\sigma_{i,j}$ belongs to an *alphabet* Σ (i.e. the set of possible strokes) and let $|\Sigma| = \text{card}\ \Sigma$ be the different number of strokes. We start by building a left-right HMM from each training sequence, we detail this step a little further. Once every training sequence has been transformed into a left-right HMM, the global initial model M_0 is defined as a mixture of all these left-right HMM with uniform priors. This model implements a probablistic model of the form:

$$P(s|M_0) = \sum_{i=1}^{n} w_i P(s|\lambda_i)$$

where s is an observed sequence, λ_i the i^{th} left-right HMM built from s_i and for each i, $w_i = \frac{1}{n}$.

We now come back to the building of a left-right HMM from a trainig sequence. The HMM built from s_i is a left-right HMM with l_i states, one for each stroke in s_i. There is then a correspondance between a state and a symbol in Σ. We explain now how emission probability laws associated to the states of this HMM are defined.

There are a few solutions for the initialization of the discrete emission probability laws defined on the alphabet Σ. Ideally, the emission probability law in a state corresponding to a stroke σ should give high probability to strokes similar to σ. [17] suggests to learn these laws with a standard Maximum Likelihood criterion using an EM algorithm. However, this strategy did not appear relevant to us since training is delicate insofar as it requires to find a good initialization. Suppose that we have 1000 training sequences, each of length 30, with the alphabet size $|\Sigma|$ equal to 50. We therefore have 30 000 symbols to estimate 1000x30 emission probability laws. If we choose to estimate all probability laws without sharing parameters, we would have to estimate 30 000 probability laws, each defined on Σ with 50 parameters. This is practically impossible to achieve with only 30 000 observed symbols in the trainng set. We rather chose to share emission probability laws between all states corresponding to the same stroke of Σ so that there are only $|\Sigma|$ probabilitiy laws to estimate. For instance, if the first state and the last state of a left-right HMM correspond to the same stroke σ, both states share the same emission probability law. Then, our strategy requires to estimate only 2500 parameters (50 laws, each one defined with 50 parameters) from the same number of observations. In addition, we will show in our experiments that a Maximum Likelihood Estimation scheme is not necessarily an optimal method from the clustering point of view.

We chose to define probability emission laws by estimating, with countings from D, the similarity between strokes in Σ. We consider as similar two strokes which appear in the same context: Let s be a sequence of strokes ($s \in \Sigma^*$), and let $P_s(\sigma)$ be the probability of seeing stroke σ after subsequence s. An estimate for $P_s(\sigma)$ is:

$$P_s(\sigma) = \frac{w(s\sigma)}{w(s)}$$

where $w(s)$ represents the number of occurrence of the subsequence s in D. We may then characterize a *profile* for a stroke σ as the distribution:

$$P_\sigma = \{P_s(\sigma), s \in sub(D)\},$$

where $sub(D)$ stands for all subsequences of sequences in D. We then define the similarity κ between two strokes $(\sigma_1, \sigma_2) \in \Sigma^2$ by the correlation between the *profiles* P_{σ_1} and P_{σ_2}:

$$\kappa(\sigma_1, \sigma_2) = \text{corr}(P_{\sigma_1}, P_{\sigma_2}).$$

Finally, the emission probability law associated to a state s_σ (built from a stroke σ) is given by:

$$b_{s_\sigma} = \left\{ \frac{\kappa(\sigma, \sigma_2)}{\sum_{\sigma_2 \in \Sigma} \kappa(\sigma, \sigma_2)}, \sigma \in \Sigma \right\}$$

Note that we could use prior knowledge on strokes since they correspond to particular shapes (angle or curvature). We will provide, as a reference, experimental results using manually tuned emission probability laws.

2.2 Iterative simplification algorithm

The general outline of the algorithm is to iteratively merge the two *closest* left-right HMM using a distance δ between left-right HMM, which is presented in next subsection:

1. For each sequence of the database, build the corresponding left-right HMM.
2. Build the initial HMM model $M = M_0$ as detailed in previous section. Using N data sequences, M is a mixture of N left-right HMM.
3. Repeat until the stopping criterion is satisfied:
   (a) Compute the distances δ between all left-right HMM of M.
   (b) Select the pair of left-right HMM (u, v) closest with respect to δ.
   (c) Remove the left-right HMM u (resp. v) from the mixture model if prior $P(u) < P(v)$ (resp. $P(u) > P(v)$). The prior for the left-right HMM that is kept is $P(u) + P(v)$.

In the present implementation, the stopping criterion is satisfied when a given number of left-right HMM is obtained. Although it does not exist satisfying methods to determine automatically an optimal number of left-right HMM (i.e. clusters), we are working on a few alternative methods. Also, note that step 3.(c) is a trivial case of merging left-right HMM. A more general case would be to compute the merged left-right HMM m between u and v and to perform the iterative step $M \leftarrow M \setminus \{u, v\} \cup \{m\}$.

2.3 Left-right HMM distance measure

Let M_1 and M_2 be two left-right HMM of respective lengths l_1 and l_2. The straightforward approach proposed in [12] is too costly and we searched a distance $\delta(M_1, M_2)$ between these two models that takes account of their left-right topology. It is based on an alignment between the states of M_1 and M_2 using a Dynamic Time Warping algorithm ([14], [2]) where local costs are distances between states (i.e. between their emission probability laws). We use a commonly used distance between distributions, the symmetrized distance from Kullback-Leibler d_{KL}. Hence the algorithm seeks to find an optimal alignment between the states of M_1 and the states of M_2, that minimises the cost:

$$J = \sum_{k=1}^{p} d_{KL}(i_k, j_k)$$

where $d_{KL}(i_k, j_k)$ is the symmetrized Kullback-Leibler distance between the probability distributions of the state i_k of M_1 and the state j_k of M_2 and where the sequence of the indices $\{(i_k, j_k), k \in [1, p]\}$ corresponds to an authorized alignment of pair of states of the two models (using classical DTW path constraints):

$$(i_k, j_k) \in \{(i_{k-1}, j_{k-1} + 1), \\ (i_{k-1} + 1, j_{k-1}), (i_{k-1} + 1, j_{k-1} + 1)\}$$

As we are interested in left-right HMM, we impose the limiting conditions $(i_1, j_1) = (1, 1)$ and $(i_p, j_p) = (l_1, l_2)$ so that δ is defined as:

$$\delta(M_1, M_2) = \hat{J} = \min_{\substack{p, \{(i_k, j_k), k \in [1,p]\} \\ (i_1, j_1) = (1,1); (i_p, j_p) = (l_1, l_2)}} \sum_{k=1}^{p} d_{KL}(i_k, j_k)$$

3 Experimental results

In this section, we first present the data used (section 3.1) then we show results related to emission probability estimation (section 3.2), clustering (section 3.3) and classification (section 3.4).

3.1 Database

We carried out our experiments on on-line handwritten digits written by about 100 writers, extracted from the Unipen database [6]. The rough on-line signal is a temporal sequence of pen coordinates and is first preprocessed as in [1] using a kind of direction coding. A handwritten signal is represented as a sequence of *strokes*, each one is characterized by a direction and a curvature. The strokes belong to a finite dictionnary Σ of 36 elementary *strokes*, including 12 straight lines in directions uniformly distributed between 0 and 360°, 12 convex curves and 12 concave curves. Such a sequence of strokes represents the shape of the signal and

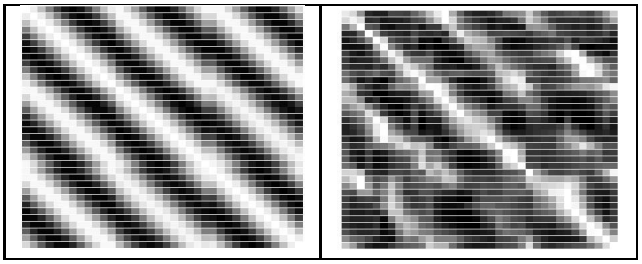

Figure 1. 36x36 matrixes representing probability laws attached to strokes of Σ, fixed manually using prior knowledge (left) and learned from the data (right).

may be efficiently used for recognition. All handwriting signals are converted into sequences of elementary strokes. We used several subsets of the database: 1000 samples of digits '0' and '9', 1000 samples of all ten digits, and about 6000 samples of all ten digits for classification.

3.2 Emission probability estimation

Figure 1 represents emission probabilities laws over alphabet Σ. The dimension of the matrixes are 36x36: The pixel at the intersection of the ith row and jth colunn is the probability of the ith stroke by a state corresponding to the jth stroke in Σ ; Gray levels are proportional to probabilities (white = close to 1, black = close to 0). The left matrix is fixed manually according to prior knowledge [1] while the right matrix is estimated with the method presented in section 2.1. We note a strong correlation between these matrixes, which shows that our estimation method allows to capture efficiently the information of similarity between symbols contained in the sequences of D.

3.3 Clustering experiments

3.3.1 Evaluation criteria

Evaluating unsupervised methods (e.g. clustering) is an open problem and we do not have any label information about allographs in the database. Then, although our approach can be used to identify a digit's allographs, we chose to perform clustering experiments on databases including signals of various but close digits (e.g. '0' and '9') and other experiments with all ten digits. In all experiments, after learning a global HMM, all sequences in D are clustered using remaining left-right HMM as cluster models and we use criteria relying on a labeling of samples with class information (digits) to evaluate our algorithms.

A few criteria may be used to evaluate clustering results [16]. Among these, we chose the *precision* measure that is also used in classification. In the following, we name *clusters* the result of our clustering and *classes* the labelling of the data. For a cluster j, P_{ij} is the probability that an element of the cluster j belongs to class i. This probability is estimated by counting: Let n_j be the number of sequences in cluster j and n the total number of sequences in the data. Then,

$$precision = \sum_j \frac{n_j}{n} \max_i P_{ij}.$$

We compare our approach with a recent approach for sequences clustering proposed by [4], based on EM algorithm, that we applied both with Markov chains (as in the article) and with HMM. As this algorithm strongly depends on initialization (a random initialization providing bad results), we initialized this algorithm with the result of our approach.

3.3.2 Results

In a first set of experiments, we use 100 samples of digits '0' and '9' whose drawings are very similar. As an illustration, the resulting clusters from one experiment using our model are drawn in figure 3: The discovered clusters are homogeneous (including either '0' or '9' samples). The two clusters for digit '0' include indeed slightly different drawings since samples from the smaller set are drawn the other way round. In this figure, the drawing is generated from our model representation. Therefore, characters do not display as nicely as the fine-grained original representation. It further illustrates that our approach does not require a precise information on characters, and could be used, for instance, on low-resolution devices.

We then compare our algorithm with the EM reestimation method described in [4][1] and a variant (stochastic EM) using the CEM algorithm, as it may outperform EM in unsupervised learning, especially when dealing with too few data to estimate the likelihood correctly [5]. This experiment was conducted using 1000 samples of digits '0' and '9'. The results are presented in figure 2. By nature, our approach does not require initialization, this is not the case for EM and CEM approach. Therefore for each experiment (number of clusters) we initialized both EM and CEM algorithm with the resulting clustering of our method.

As may be seen, EM and CEM reestimation tend to weaken the results. We think that our learning strategy by nature is more adapted to discover typical cluster of sequences. The reason lies in that a left-right HMM built from a training sequence, as detailed in section 2.1, cannot handle much variability around this training sequence. At the opposite, performing EM reestimation may result in less specific left-right HMM, thus in less precise clusters.

[1] though applied with HMM instead of Markov Chains, the latter providing weaker results

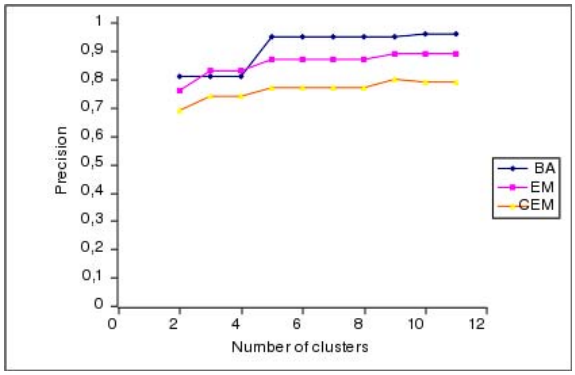

Figure 2. Clustering precision for 3 algorithms (our approach, BA ; reestimated via EM, and via CEM)

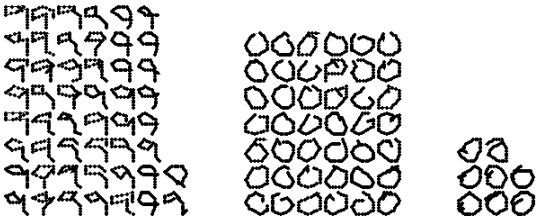

Figure 3. The three discovered clusters for a database of on-line handwriting samples of digits '0' and '9'

To further validate our approach, we performed another comparison on a set of 1000 samples of the ten digits (table 1). The results first outline that clustering is indeed a difficult task since for a reasonnable number of clusters (20) precision does not exceed 80% whereas classification results on such handwriting signals may reach about 95% [1]. Also, our approach outperforms other methods provided there are enough clusters but performance falls sharply when the number of clusters decreases. Note that a reasonnable number of clusters should be greater than 10 since there exists several allographs for every digit. From this point of view, experiments show that our approach significantly outperforms EM in this range. As discussed above, reestimation with EM maximizes likelihood so that a single left right HMM may generate with high probability different signals corresponding to different allographs, and still achieve a minimal performance.

Clusters	**BA**	Cadez MC	Cadez HMM
25	**80%**	62%	64%
20	**80%**	60%	64%
17	**71%**	58%	64%
16	59%	52%	**63%**
10	35%	48%	**58%**

Table 1. Clustering performance for our approach (BA) compared to [4] with Markov Chains (MC) and HMM

3.4 Classification experiments

We present here preliminary experiments on classification. We use our learning algorithm for every digit resulting in a mixture of left right HMM as digit models. Experiments were performed on a bigger subset of Unipen, about 6000 samples of the ten digits (from '0' to '9') with 800 samples for training and the remaining for test. Recognition rates are displayed in figure 4 as a function of the number of components in mixture models. Without simplification of initial HMM (i.e. about 80 left-right HMM per digit) the classification reach an asymptotic performance of 92.5%. By learning a model for each digit, we can achieve same or better performance while simplifying the models. One main advantage of having compact character models is a significant speedup on recognition speed. For example, using 10 allograph models per character, we achieve a recognition speed of 1300 characters per second on a Pentium IV system. Note that our performance does not match state of the art recognition rates [13]. The main reasons lie in that, in this preliminary study, our model do not integrate an adequate duration model and diacritics are not taken in account. However, in sight of these shortcomings, the results may appear to be very promising since we obtain the same level of performance than by using the approach described in [8] using the same data definition.

4 Conclusions and future work

We presented a model-based approach to cluster sequences that we tackled through unsupervised HMM learning. We propose to learn, from the data, the structure and parameters of a global HMM within a subclass of HMM (mixture of left-right HMM) that seems more appropriate for sequence clustering and allographs identification. The learning consists in building from data an initial mixture model of left-right HMMs and then simplifying it by removing iteratively the less significant left-right HMM. This algorithm relies on an original estimation of emission probability laws and on the definition of a distance

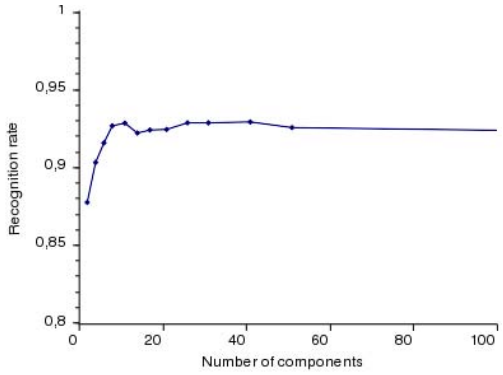

Figure 4. Recognition rate (digits) as a function of the number of components per mixture model

between left-right HMM. We provide experimental results for clustering and classifcation of on-line handwritten digits. These prelimary results show that our algorithm outperforms Maximum Likelihood approach and seems promising, we look forward to extend and validate our approach on larger datasets.

References

[1] T. Artières and P. Gallinari. Stroke level HMMs for on-line handwriting recognition. In *8th International Workshop on Frontiers in Handwriting Recognition (IWFHR-8)*, Niagara, Aug. 2002.

[2] C. Bahlmann and H. Burkhardt. Measuring HMM similarity with the bayes probability of error and its application to online handwriting recognition. In *ICDAR*, pages 406–411, 2001.

[3] M. Brand. Structure learning in conditional probability models via an entropic prior and parameter extinction. *Neural Computation*, 11:1155–1182, 1999.

[4] I. V. Cadez, S. Gaffney, and P. Smyth. A general probabilistic framework for clustering individuals and objects. In R. Ramakrishnan, S. Stolfo, R. Bayardo, and I. Parsa, editors, *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-00)*, pages 140–149, N. Y., Aug. 20–23 2000. ACM Press.

[5] G. Celeux and J. Diebolt. A random imputation principle : the stochastic em algorithm. Technical report, Rapport de recherche de l'INRIA- Rocquencourt, 1988.

[6] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. Unipen project of on-line data exchange and benchmarks. In *International Conference on Pattern Recognition, ICPR'94*, pages 29–33, Jerusalem, Israel, 1994. IEEE Computer Society Press.

[7] P. Lockwood and M. Blanchet. An algorithm for the dynamic inference of hidden markov models (dihmm). In *Proc. ICASSP93*, pages 251–254, 1993.

[8] S. Marukatat, R. Sicard, T. Artières, and P. Gallinari. A flexible recognition engine for complex on-line handwritten character recognition. In *7th International Conference on Document Analysis and Recognition (ICDAR 2003)*, Edinbourgh, Scotland, Aug. 2003.

[9] A. Nosary, L. Heutte, and T. Paquet. Unsupervised writer adaptation applied to handwritten text recognition. *Pattern Recognition*, 37:385–388, 2003.

[10] S. M. Omohundro. Best-first model merging for dynamic learning and recognition. In J. E. Moody, S. J. Hanson, and R. P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4, pages 958–965. Morgan Kaufmann Publishers, Inc., 1992.

[11] M. Perrone and S. Connell. K-means clustering for hidden markov models. In *In Proceedings of the Seventh International Workshop on Frontiers in Handwriting Recognition*, pages 229–238, Amsterdam, Netherlands, September 2000.

[12] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series, 1993.

[13] E. H. Ratzlaff. Methods, report and survey for the comparison of diverse isolated character recognition results on the unipen database. In *Seventh International Conference on Document Analysis and Recognition*, Edinburgh, Scotland, August 2003.

[14] D. Sankoff and J. B. Krushkal. *Time warps, string edits, and macromolecules: The theory and practice of sequence comparison*. Addison-Wesley, 1983.

[15] P. Smyth. Clustering sequences with hidden markov models. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 648. The MIT Press, 1997.

[16] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques, 2000.

[17] A. Stolcke and S. Omohundro. Hidden Markov Model induction by bayesian model merging. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5, pages 11–18. Morgan Kaufmann, San Mateo, CA, 1993.

[18] L. Vuurpijl and L. Schomaker. Finding structure in diversity: A hierarchical clustering method for the categorization of allographs in handwriting, 1997.