Cyclic Linear Hidden Markov Models for Shape Classification[⋆]

Vicente Palazón, Andrés Marzal, and Juan Miguel Vilar

Dept. Llenguatges i Sistemes Informàtics. Universitat Jaume I de Castelló. Spain
{palazon,amarzal,jvilar}@lsi.uji.es

Abstract. In classification tasks, shape descriptions, combined with matching techniques, must be robust to noise and invariant to transformations. Most of these distortions are relatively easy to handle, particularly if we represent contours by sequences. However, starting point invariance seems to be difficult to achieve. The concept of cyclic sequence, a sequence that has no initial/final point, can be of great help. We propose a new methodology to use HMMs to classify contours represented by cyclic sequences. Experimental results show that our proposal significantly outperforms other methods in the literature.

Keywords: Cyclic Sequences, Hidden Markov Models, Shape Classification.

1 Introduction

Shape classification is a very important problem with applications in several areas such as industry, medicine, biometrics and even entertainment.

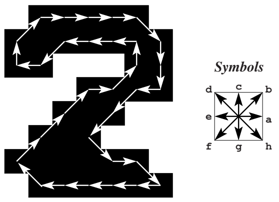
The first step towards the design of a shape classifier is feature extraction. Shapes can be represented by their contours or by their regions [1]. However, contour based descriptors are widely used as they preserve the local information which is important in the classification of complex shapes.

The next step is shape matching. Dynamic Time Warping (DTW) based shape matching is being increasingly applied [2,3,4]. A DTW-based dissimilarity measure seems natural for optimally aligning contours, since it is able to align parts instead of points and is robust to deformations. Hidden Markov Models (HMMs) are also being used as a possible shape modelling and classification approach [5,6,7,8]. Hidden Markov Models are a general approach to model sequences. They are stochastic generalizations of finite-state automata, where transitions between states and generation of output symbols are modelled by probability distributions. HMMs have the properties of DTW matching, but also provide a probabilistic framework for training and classification.

Shape descriptors, combined with shape matching techniques, must be invariant to many distortions, including scale, rotation, noise, etc. Most of these

⋆ Work partially supported by the *Ministerio de Educación y Ciencia* (TIN2006-12767), the *Generalitat Valenciana* (GV06/302) and *Bancaixa* (P1 1B2006-31).

distortions are relatively easy to deal with, particularly when we convert contours in sequences and use DTW or HMMs to match them. However, no matter what representation is used, starting point invariance seems to be difficult to achieve. A contour can be transformed into a sequence by choosing an appropriate starting symbol, but its election is always based on heuristics which seldom work in unrestricted scopes. The most suitable solution to this problem is to measure distances between every possible initial symbol of the sequences. The concept of cyclic sequence arises here. Figure 1 depicts the coding of a character contour as a cyclic sequence with an 8-direction code. A cyclic sequence is a sequence of symbols or values that has neither beginning nor end, i.e., a cyclic sequence models the set of every possible cyclic shift of a sequence, thus, to measure distances between two cyclic sequences is equivalent to measure distances between every possible cyclic shift of both sequences.



A = **`aaaahggeffhaheeeeedbbbabceeefecb`**

Fig. 1. A shape of a character “two” is coded as a cyclic sequence of directions along the contour

So the question is now: how can we train HMMs for cyclic sequences? There is a time order, but we do not know where the sequences begin. An immediate idea is to use a cyclic topology as in [5], but this is not the best solution, as we will see. To overcome this problem, in the following sections, we will propose a new methodology to properly work with HMMs in order to classify cyclic sequences.

The paper is organised as follows: Section 2 and Section 3 give an overview of Hidden Markov Models and Cyclic sequences. The proposed approach is discussed in Section 4 and Section 5. Experimental results are presented in Section 6. Finally, the paper ends with conclusions in Section 7.

2 Hidden Markov Models

A Hidden Markov Model (HMM) [9] contains a set of states, each one with an associated emission probability distribution. At any instant t, an observable event is produced from a particular state and it only depends on the state. The transition from one state to another is a random event only depending on the departing state. Without loss of generality, in the following we will only consider discrete HMMs, i.e., the set of observable events is finite.

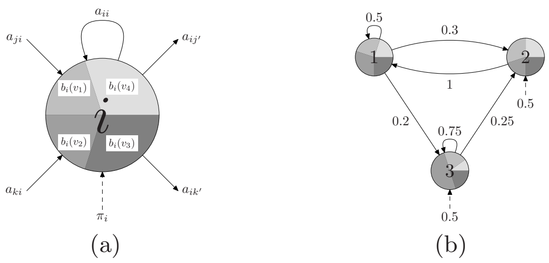


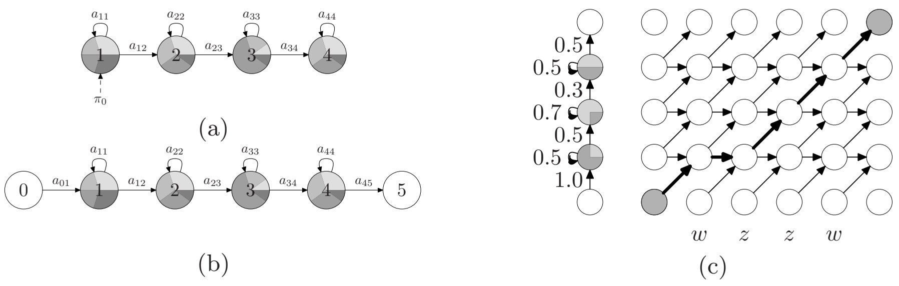
Fig. 2. (a) An HMM state that can emit any of four symbols according to the probability distribution depicted as a pie chart. (b) A complete HMM.

Given an alphabet $\Sigma = \{v_1, v_2, \ldots, v_w\}$, an HMM with n states is a triplet (A, B, π) where (1) $A = \{a_{ij}\}$, for $1 \leq i, j \leq n$, is the state transition probability matrix (a_{ij} is the probability of being in state i at time t and being in state j at time $t+1$); (2) $B = \{b_{ik}\}$, for $1 \leq i \leq n$ and $1 \leq k \leq w$, is the observation probability matrix (b_{ik} is the probability of observing v_k while being in state i); and (3) $\pi = \{\pi_i\}$, for $1 \leq i \leq n$ is an initial state probability distribution (π_i is the probability of being in state i when $t = 1$). The following conditions must be satisfied: for all i, $\sum_{1 \leq j \leq n} a_{ij} = 1$ and $\sum_{1 \leq k \leq w} b_{ik} = 1$; and $\sum_{1 \leq i \leq n} \pi_i = 1$. Figure 2 (a) depicts a state and Figure 2 (b) shows a complete HMM (transitions with null probability are not shown).

Apart from this definition, there is another one that has been popularised by the toolkit HTK [10]. This definition has two non-emitting states, the initial and the final one. The initial state, that we will identify with the number 0, has no input transitions (it eliminates the need for an explicit initial state distribution π since a_{0i} can be interpreted as π_i) and the final state, that we will identify with $n+1$, has no output transitions. These special non-emitting states simplify some computations and eases HMM composition. In the following, we will use this alternative definition.

There are efficient iterative algorithms for training the parameters of an HMM [11,12]. Unfortunately, there are no effective methods for estimating the number of states and the topology of the model. These are usually chosen heuristically depending on the application features. For example, when the sequence of symbols can be segmented, all the symbols in a segment are emitted by the same state, and consecutive segments are associated to consecutive states, a so-called Linear HMM (LHMM), i.e., a left-to-right topology like the ones shown in Figure 3 (a) and (b), can be used.

For an HMM $\lambda = (A, B)$ and a sequence of observed symbols, $x = x_1 x_2 \ldots x_m$, there are three basic problems that must be solved to be useful in applications: (1) the evaluation problem, i.e., the probability of x, given λ; (2) the decoding problem, i.e., obtaining the sequence of states that most likely produced x; and (3) the learning problem, i.e., estimating λ to maximise the probability of generating x. There are well-known, efficient algorithms for the two first problems. The Viterbi algorithm [13] solves the decoding problem by evaluating $\phi_{n+1}(m+1)$,



a_{11} a_{22} a_{33} a_{44}
1 a_{12} 2 a_{23} 3 a_{34} 4
π_0
(a)
0 a_{01} 1 a_{12} 2 a_{23} 3 a_{34} 4 a_{45} 5
(b)
0.5 0.5 0.3 0.7 0.5 0.5 1.0
w z z w
(c)

Fig. 3. (a) A Linear HMM. (b) A Linear HMM using the HTK definition. (c) Trellis for a Linear HMM and a sequence of length 4. The optimal alignment is shown with thicker arrows.

where

$$\phi_j(t) = \begin{cases} 1, & \text{if } t = 0 \text{ and } j = 0; \\ 0, & \text{if } t = 0 \text{ and } j \neq 0; \\ \max_{1 \le i \le N}(\phi_i(t-1) \cdot a_{ij}) \cdot b_j(x_t), & \text{if } 1 \le t \le m \text{ and } 1 \le j \le n; \\ \max_{1 \le i \le N}(\phi_i(m) \cdot a_{i,n+1}), & \text{if } t = m+1 \text{ and } j = n+1. \end{cases}$$

The Forward algorithm solves the evaluation problem by computing a similar recursive expression with summations instead of maximizations. Both recursive equations can be solved iteratively by Dynamic Programming in $O(n^2m)$ time ($O(nm)$ for LHMMs). The iterative version of the Viterbi algorithm computes an intermediate value at each node of the trellis graph (see Figure 3 (c)). Each node (j, t) corresponds to a state (j) and a time instant (t) and stores $\phi_j(t)$. The value at $(n + 1, m + 1)$ is the final result. The Viterbi algorithm solves the decoding problem by recovering the optimal alignment (sequence of states) in the trellis (see Figure 3 (c)).

There is no algorithm that optimally solves the training problem. The Baum-Welch [11] and the segmental K-means [12] procedures are used to iteratively improve the parameters estimation until a local maximum is found. In practice, both methods offer a comparable performance regarding classification rates.

3 Hidden Markov Models for Cyclic Sequences

A cyclic sequence can be seen as the set of sequences obtained by cyclically shifting a conventional sequence:

Definition 1 ([14]). *Let $x = x_1 \dots x_m$ be a sequence from an alphabet Σ. The cyclic shift $\sigma(x)$ of a sequence x is defined as $\sigma(x_1 \dots x_m) = x_2 \dots x_m x_1$. Let σ^s denote the composition of s cyclic shifts and let σ^0 denote the identity. Two sequences x and x' are cyclically equivalent if $x = \sigma^s(x')$, for some s. The equivalence class of x is $[x] = \{\sigma^s(x) : 0 \le s < m\}$ and it is called a* cyclic

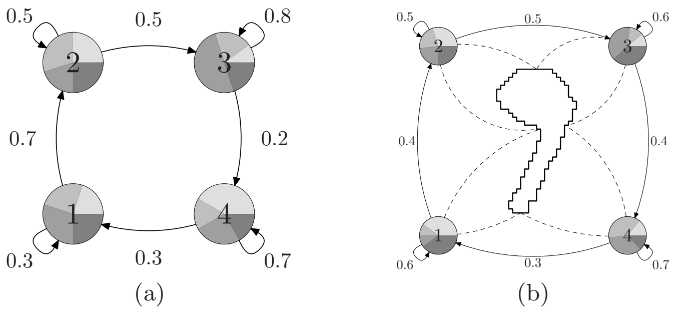


Fig. 4. (a) Cyclic HMM as proposed in [5]. (b) The contour of a shape is segmented and each segment is associated to a state of the HMM. Ideally, each state is responsible for a single segment.

sequence. *Any of its members is a representative of the cyclic sequence. For instance, the set* $\{wzz, zzw, zwz\}$ *is a cyclic sequence and* wzz *(or any other sequence in the set) can be taken as its representative.*

Since cyclic sequences have no initial/final point, Linear HMMs seem inappropriate to model them. In [5], the authors proposed a circular HMM topology to model cyclic sequences. Figure 4 (a) shows this topology (the initial and final non-emitting states are not shown for the sake of clarity). This topology can be seen as a modification of the left-to-right one where the "last" emitting state is connected to the "first" emitting state. The proposed structure eliminates the need to define a starting point: the cyclic sequence can be segmented to associate consecutive states to consecutive segments in the cyclic sequences, but no assumption is made on which is the first state/first segment (see Figure 4 (b)); therefore, there is an analogy with Linear HMMs. However, there is a problem that breaks this analogy: the model is ergodic (all states can be reached from any state) and the cyclic sequence symbols can "wrap" the model, i.e., the optimal sequence of states can contain non-consecutive, repeated states and, therefore, a single state can be responsible for the emission of several non-consecutive segments in the cyclic sequence. Thus, the problem that we have is the following:

Problem 1. To properly model cyclic sequences, HMMs should take into account that any symbol of the sequence can be emitted by the first emitting state and when such a symbol has been chosen as emitted by this state, its previous symbol must be emitted by the last state.

4 Cyclic Linear HMMs

We can use Linear HMMs in a similar way than cyclic sequences. A cyclic Linear HMM (CLHMM) can be seen as the set of LHMMs obtained by cyclically shifting a conventional LHMM:

Definition 2. *Let $\lambda = (A, B)$ be an LHMM. Given A, let $\sigma(A)$ be the following transformation:*

$$A = \begin{bmatrix} 1 & 0 & \dots & & & & 0 \\ 0 & a_{11} & a_{12} & 0 & \dots & & 0 \\ 0 & 0 & a_{22} & a_{23} & 0 & \dots & 0 \\ & & & \ddots & \ddots & & \\ 0 & \dots & 0 & \ddots & \ddots & 0 & 0 \\ 0 & \dots & & & a_{nn} & a_{nn+1} & 0 \\ 0 & \dots & & & & 0 & 0 \end{bmatrix}, \sigma(A) = \begin{bmatrix} 1 & 0 & \dots & & & & 0 \\ 0 & a_{22} & a_{23} & 0 & \dots & & 0 \\ & & \ddots & \ddots & & & \\ 0 & 0 & \ddots & \ddots & 0 & \dots & 0 \\ 0 & \dots & 0 & a_{nn} & a_{nn+1} & 0 & 0 \\ 0 & \dots & & & a_{11} & a_{12} & 0 \\ 0 & \dots & & & & 0 & 0 \end{bmatrix}.$$

Let $\sigma(B)$ be $\sigma(b_1 \dots b_n) = b_2 \dots b_n b_1$ (where b_i are rows from matrix B). The composition of r cyclic shifts of λ is defined as $\sigma^r(\lambda) = (\sigma^r(A), \sigma^r(B))$. Two LHMMs λ and λ' are cyclically equivalent if $\lambda = \sigma^r(\lambda')$, for some r. The equivalence class of λ is $[\lambda] = \{\sigma^r(\lambda) : 0 \leq r < n\}$ and it is called a Cyclic LHMM. *Any of its members is a representative of the Cyclic LHMM (see Figure 5 (a)).*

Then, to solve Problem 1:

Definition 3. *The Viterbi score for a cyclic sequence $[x_1x_2 \dots x_m]$ and a CLHMM $[\lambda]$ is defined as $P([x]|[\lambda]) = \max_{0\leq r<n} \left(\max_{0\leq s<m} P(\sigma^s(x)|\sigma^r(\lambda)) \right)$ and this score has associated an optimal alignment that is called optimal cyclic alignment.*

This is computationally expensive, but the following lemma shows that in order to compute the Viterbi score for a cyclic sequence and a CLHMM, one can simply choose a representative of the CLHMM and compute the Viterbi score between this LHMM and the cyclic sequence.

Lemma 1. $P([x]|[\lambda]) = P([x]|\lambda) = \max_{0\leq s<m} P(\sigma^s(x)|\lambda)$

Proof: Consider an optimal alignment ς_1 that represents a maximum probability between λ and $\sigma^{s_1}(x)$, for some s_1, then, there is an optimal alignment ς_2 between $\sigma(\lambda)$ and $\sigma^{s_2}(x)$, for some s_2, such that ς_2 represents exactly the same emitted symbols for each state as ς_1. $\square$

Therefore, it can be computed by means of the conventional Viterbi score computed on m conventional sequences in $O(m^2n)$ time. We propose a more efficient algorithm to evaluate the Viterbi score. The method computes the optimal alignment that begins in any state, visits all the states and does not visit any state once it has left it. The algorithm is inspired in the Maes' algorithm for the Cyclic Edit Distance (CED) [14] and computes the Viterbi score in $O(mn \log m)$ time. The score is computed on an extended trellis where the original sequence appears concatenated with itself in the horizontal axis and alignments must begin and end in nodes with the same colour (corresponding to the size of the string), see Figure 5 (b). The efficiency of the algorithm is based on the "non-crossing

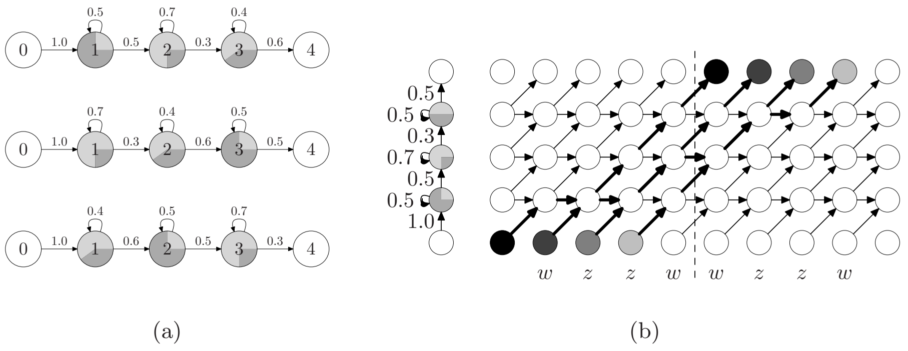


Fig. 5. (a) A CLHMM represented by its set of LHMMs. (b) Extended trellis for a Linear HMM and a cyclic sequence of length 4. The optimal alignments for each starting point are shown with thicker arrows, one of them is the optimal cyclic alignment.

paths" property [14]: Let ς_i be the optimal alignment beginning at node $(i, 0)$ and ending at node $((m + i + 1, n + 1)$ in the extended trellis and let j, k, and l be three integers such that $0 \le j < k < l \le m$; there is an optimal path starting at node $(k, 0)$ and arriving to $(k + m + 1, n + 1)$ that lies between ς_j and ς_l.

This property leads to a Divide and Conquer, recursive procedure: when ς_j and ς_l are known, $\varsigma_{(j+l)/2}$ is computed by only taking into account those nodes of the extended trellis lying between ς_j and ς_l; then, optimal alignments bounded by ς_j and $\varsigma_{(j+l)/2}$ and optimal alignments bounded by $\varsigma_{(j+l)/2}$ and ς_l can be recursively computed. The recursive procedure starts after computing ς_0 (by means of a standard Viterbi computation) and ς_m, which is ς_0 shifted m positions to the right. Each recursive call generates up to two more recursive calls and all the calls at the same recursion depth amount to $O(mn)$ time; therefore, the algorithm runs in $O(mn \log m)$ time. This adaptation of Maes' algorithm comes naturally after defining the Viterbi score in lemma 1.

In principle, we could adopt a symmetric approach defining a cyclic shift on the states of the Linear HMMs to obtain the same Viterbi score. This is appealing because $n < m$ and, therefore, "doubling" the HMM in the extended trellis instead of the sequence would lead to an $O(mn \log n)$ algorithm. This would be better than $O(mn \log m)$ since $n < m$ (and, usually, $n \ll m$). However, it cannot be directly done:

Lemma 2. $P([x]|[\lambda]) \neq \max_{0 \le r < n} P(x|\sigma^r(\lambda))$

Proof: Let us use a counterexample. Let $[x] = v_1 v_2 v_1$ be a cyclic sequence on the alphabet $\Sigma = \{v_1, v_2\}$. Let $[\lambda]$ be a CLHMM with 2 emitting states and $a_{01} = 1$, $a_{11} = 0.5$, $a_{12} = 0.5$, $a_{22} = 0.5$, $a_{23} = 0.5$, $b_{01} = 1$, and $b_{12} = 1$. The definition of the Viterbi score in lemma 1 leads to a value of 0.125 (for the sequence

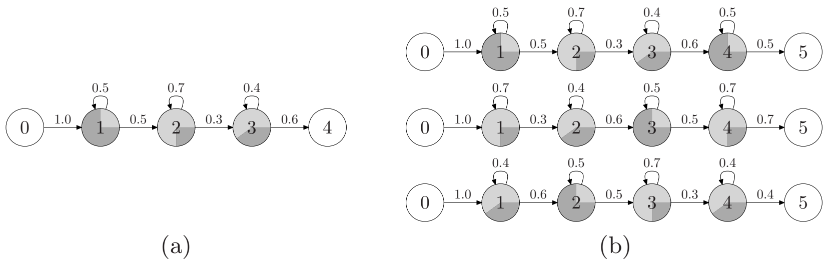


Fig. 6. (a) A CLHMM $[\lambda]$ represented by a representative (an LHMM). (b) The corresponding LHMMs for the $\iota^r(\lambda)$ operation, for $0 \leq r < n$ (where $n = 3$). From top to bottom, $\iota(\lambda)$, $\iota^2(\lambda)$ and $\iota^3(\lambda)$.

$\sigma^2(v_1v_2v_1) = v_1v_1v_2$). If we try to perform a cyclic shift in the Linear HMM, we have two possible cyclic shifts both possibilities give us 0 as the Viterbi score. □

Let $[\lambda] = (A, B)$ be a CLHMM, let $[x]$ be a cyclic sequence and let $\iota(\lambda)$ be an operation that performs a cyclic shift ($\sigma(\lambda)$) and inserts a copy of the first emitting state before the last state, but its transition to the next state has the value of its self transition (see Figure 6). Then,

Theorem 1.

$$P([x]|[\lambda]) = \max_{0 \leq r < n} \Big(\max \big(P(x|\sigma^r(\lambda)), P(x|\iota^r(\lambda)) \big) \Big).$$

Proof (sketch): Each alignment induces a segmentation on x. All the symbols in a segment are aligned with the same state of the CLHMM. There is a problem when $x_{m-p}x_{m-p+1}\dots x_m$ and $x_1x_2\dots x_q$, for some p, $q \geq 0$, belong to the same segment of x. In that case, the optimal alignment cannot be obtained by simply cyclic shifting λ, since x_m must be aligned with the state n and x_1 must be aligned with the state 1, i.e., they never fall in the same segment. The LHMM $\iota^r(\lambda)$, formed by inserting to $\sigma^r(\lambda)$ the first emitting state after the last one, permits to align $x_{m-p}x_{m-p+1}\dots x_m$ and $x_1x_2\dots x_q$ with the first state, since this state also appears at the end of $\iota^r(\lambda)$. On the other hand, there is another problem: let us suppose we have now the complete segment at the beginning of the sequence, $p+q$ symbols, then the first self transition must be executed $p+q-1$ times, but if the segment is in the situation explained above, the first self transition will be executed just $p+q-2$ times, the transition to the last non-emitting state provides this necessary extra transition. □

Fortunately, for each value of r, $P(x|\sigma^r(\lambda))$ can be obtained as a subproduct of the computation of $P(x|\iota^r(\lambda))$. The trellis underlying $P(x|\sigma^0(\lambda))$ is a subgraph of the one underlying $P(x|\iota^0(\lambda))$.

The value of $P(x|\sigma^r(\lambda))$ and $P(x|\iota^r(\lambda))$, for each r, can be obtained by computing optimal alignments in an *extended trellis* similar to the one in Figure 5 (b), but now "doubling" the LHMM. It should be taken into account that, unlike in Maes' algorithm, the optimal path starting at $(r, 0)$ can finish either at node $(r+n-1, m)$ or $(r+n, m)$ and the recursive computation can be applied just using the optimal alignments between $\sigma^r(\lambda)$ and x as a new left or right border.

5 Segmental *K*-means for Cyclic Linear HMMs

The proposed algorithm to compute the Viterbi score for a CLHMM cannot be extended to Forward-value computation because there is no optimal alignment on the trellis on which the "non-crossing paths" property holds. Since the Baum-Welch training procedure is based on the Forward (and Backward) values, we cannot use it for cyclic strings without requiring n times more time, which is too expensive. However, on this purpose, we can adapt the segmental K-means algorithm [12].

In creating the CLHMM for each class, we should guarantee that the parameters we obtain are the optimum for a given set of training cyclic sequences. Since our decision rule is the state optimised likelihood function, it requires that the estimated parameter $[\bar{\lambda}]$ be such that $P([x]|[\bar{\lambda}])$ is maximised for the training set. Starting from an initial model $[\lambda^0]$ (the superscripts indicate the iteration number), this procedure takes us from $[\lambda^k]$ to $[\lambda^{k+1}]$ such that $P([x], \varsigma^k|[\lambda^k]) \leq P([x], \varsigma^{k+1}|[\lambda^{k+1}])$, where ς^k is the optimal cyclic alignment for $[x] = [x_1 x_2 \ldots x_m]$ and $[\lambda^k]$. According to lemma 1, this is equivalent to $P([x], \varsigma^k|\lambda^k) \leq P([x], \varsigma^{k+1}|\lambda^{k+1})$, i.e., to train a representative LHMM λ is equivalent to train this CLHMM $[\lambda]$ (we can adopt here as well a symmetric approach, that is to say, we can apply the cyclic shift to λ instead of x, but we use this one to make understanding easier).

Thus, in this procedure to train the LHMM λ (lemma 1) a number of (training) cyclic sequences are required. Each cyclic sequence $[x] = [x_1 x_2 \ldots x_m]$ consists of m observation symbols. The algorithm then consists in the following steps[1]:

1. The process is started by performing uniform cyclic alignments as optimal ones (for all states) of any representative x for each training cyclic sequence $[x]$.
2. Compute the transition probability matrix $\hat{A}$, according to optimal cyclic alignments, for $0 \leq i \leq n$ and $0 \leq j \leq n$:

$$\hat{a}_{ij} = \frac{\text{Number of occurrences of } \{x_t \in i \text{ and } x_{(t+1) \bmod m} \in j\} \text{ for all } t}{\text{Number of occurrences of } \{x_t \in i\} \text{ for all } t}$$

3. Compute the observation probability matrix $\hat{B}$, according to optimal cyclic alignments, for $0 \leq i \leq n$ and $0 \leq k \leq w$:

[1] We only consider discrete models. These steps can be extended to the continuous ones, refer to [12] for further details.

$$\hat{b}_{ik} = \frac{\text{Number of occurrences of } \{x_t \in i \text{ and } x_t = v_k\} \text{ for all } t}{\text{Number of occurrences of } \{x_t \in i\} \text{ for all } t}$$

4. Compute $P([x]|\hat{\lambda}) = \max_{0 \leq s < m} P(\sigma^s(x)|\hat{\lambda})$ (where $\hat{\lambda} = (\hat{A}, \hat{B})$) for each training cyclic sequence $[x]$. This Viterbi score gives us an optimal cyclic alignment $\hat{\varsigma}$ for each $[x]$, $\hat{\varsigma}$ defines the sequence of states visited and which symbols are emitted for each state (and it can be computed in $O(mn \log m)$ time, Section 4).
5. If these optimal cyclic alignments are different to the previous iteration, repeat step 2 through step 5; otherwise, stop.

Following the line of thought in [12]:

Theorem 2. *The adapted segmental K-means for CLHMMs and cyclic sequences converges in Zangwill's global convergence sense [15].*

Proof (sketch): What needs to be shown is that the algorithm (that obtains $\bar{\lambda}$ from λ) is closed and that $P([x]|\lambda)$ is an ascent function for the algorithm: (i) The algorithm is closed because we assume that the function $P([x]|\lambda)$ is continuously diferentiable in λ for almost all $[x]$ in a totally finite measurable space; and (ii) Let ς^* and $\bar{\varsigma}$ be two optimal cyclic alignments such that, $\varsigma^* = \arg\max_{\varsigma} P([x], \varsigma|\lambda)$ and $\bar{\varsigma} = \arg\max_{\varsigma} P([x], \varsigma|\bar{\lambda})$, then:

$$\begin{aligned} \max_{\varsigma} P([x], \varsigma|\bar{\lambda}) &\geq P([x], \varsigma^*|\bar{\lambda}) \\ = \max_{\lambda'} P([x], \varsigma^*|\lambda') &= \max_{\lambda'} \Big(\max_{\varsigma} \big(\max_{r} P(\sigma^r(x), \varsigma|\lambda') \big) \Big) \\ &\geq \max_{\varsigma} P([x], \varsigma|\lambda). \end{aligned} \quad (1)$$

The maximization over λ' in (1) can be replaced by the adapted segmental K-means method explained above. □

Finally, taking into account that this iterative training method obtains a local maxima and for this reason, a good starting LHMM plays an important role. Step 1 obtains this initial LHMM, but as you have probably noticed it is an almost random procedure. We propose a different, automatic method for a Step 0: finding an optimal starting point for all cyclic sequences through comparison with a reference cyclic sequence via the Cyclic Dynamic Time Warping (CDTW) [2] algorithm. The starting point for all sequences is chosen in function of the optimal alignment in the extended CDTW graph. Thus, in step 1 the chosen representatives of the cyclic strings will all have a relative order and the uniform alignments will produce a better initial LHMM.

6 Experiments

In order to assess the behaviour of the algorithms presented, we performed comparative experiments on the MPEG-7 Core Experiment CE-Shape-1 (part B) database. This database consists of 1400 images divided in 70 shape classes of 20 images for each class (see Figure 7).

All the images were clipped, scaled into a 32×32 and 50×50 pixels matrix and binarized, and their outer contours were represented by 8-directional chain-codes (the average length is 120 for the 32×32 images and 200 for the 50×50 images). All the classification experiments (70 equiprobable classes) were cross-validated (20 partitions, 70 samples for testing and 1330 samples for training).

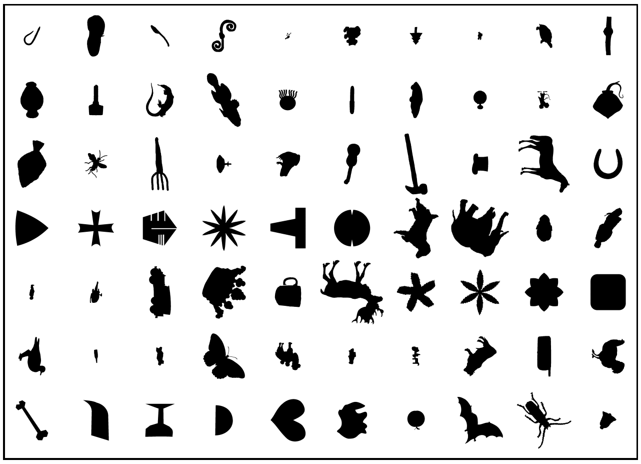
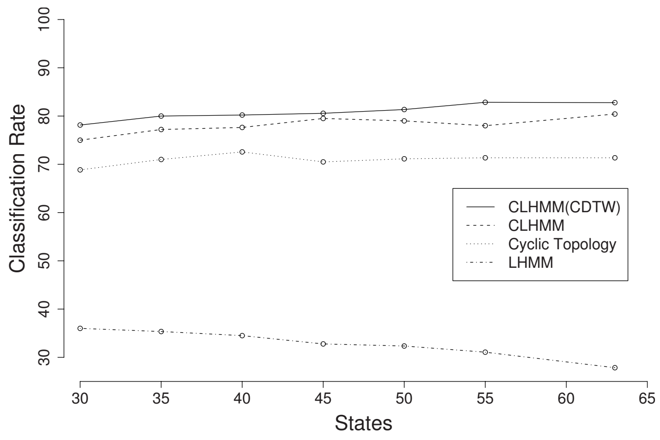


Fig. 7. Some images in the MPEG-7 CE-Shape-1 (part B) database

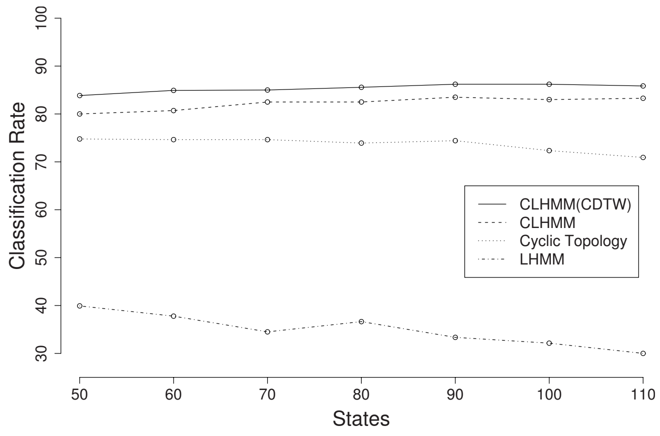
First experiments try to show that the Cyclic Linear HMMs produces better classification rates on cyclic sequences than those obtained with a conventional LHMM or with a cyclic topology [5], and also that a good initial LHMM can improve the training of the CLHMM. Then, we have four methods: (i) LHMM, a conventional LHMM with a conventional Viterbi Score and Segmental K-means[2], (ii) Cyclic Topology, the method from [5], (iii) CLHMM, our approach, and finally (iv) CLHMM(CDTW), our approach using a CDTW-based initiation.

Since we are interested in cyclic sequences, the contours were coded as conventional sequences with a random starting point. All HMMs were trained (with the HTK toolkit [10]) with randomly chosen starting points for all the sequences in the training set. Since the results can depend on the number of states, we have performed experiments varying this parameter. Figure 8 (a) shows the classification rate for the four methods with random starting points as a function of the number of states for the 32×32 images. Figure 8 (b) shows equivalent experiments for the 50×50 images. It can be seen that both methods proposed in this work improve the other ones. CLHMMs provide better results than [5] because of the

[2] Obviously, the results of this method are very poor, they are exposed to show how important is the starting point invariance.



(a)



(b)

Fig. 8. (a) Classification rate for random starting point sequences as a function of the number of states for the 32×32 images. (b) Idem for the 50×50 images.

problems that a cyclic topology has (Section 3). The highest classification rate is always obtained with CLHMM(CDTW), confirming that the adapted Segmental K-means trains better from a good initial LHMM.

We also performed experiments of time in classification (for one partition on the 50×50 images) on a 2.4Ghz Pentium 4 running Linux 2.4 (the algorithm was implemented in C++). As explained in Section 4, cyclic shifting (or doubling in the extended trellis) the LHMM instead of the sequence is advisable. In Figure 9, this fact is shown empirically.

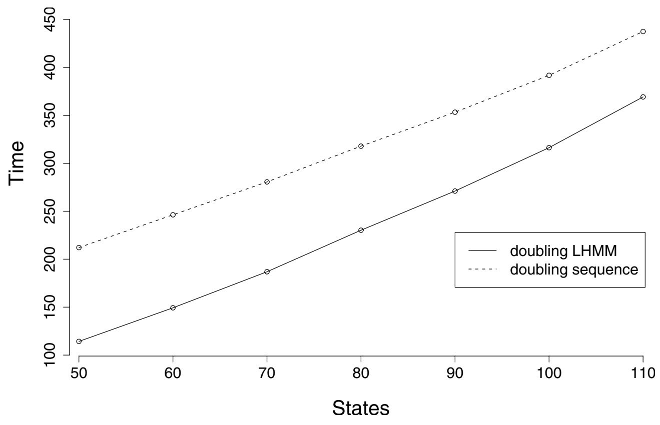


Fig. 9. Time comparison (in seconds) on the 50 × 50 images between both extended trellis, doubling the LHMM or doubling the sequence

7 Conclusions

In this work, we have presented a new methodology to use HMMs for dealing with cyclic sequences, called Cyclic Linear HMMs. We have formulated a framework to use these models for classification and training, adapting the Viterbi and the Segmental K-means algorithms. Experiments performed on a shape classification task show that our methods outperform other proposals from the literature.

References

1. Zhang, D., Lu, G.: Review of shape representation and description techniques. Pattern Recognition 37(1), 1–19 (2004)
2. Marzal, A., Palazón, V.: Dynamic time warping of cyclic strings for shape matching. In: Singh, S., Singh, M., Apte, C., Perner, P. (eds.) ICAPR 2005. LNCS, vol. 3687, pp. 644–652. Springer, Heidelberg (2005)
3. Adamek, T., O'Connor, N.E.: A multiscale representation method for nonrigid shapes with a single closed contour. IEEE Trans. Circuits Syst. Video Techn 14(5), 742–753 (2004)
4. Milios, E.E., Petrakis, E.G.M.: Shape retrieval based on dynamic programming. IEEE Transactions on Image Processing 9(1), 141–147 (2000)
5. Arica, N., Yarman-Vural, F.: A shape descriptor based on circular hidden markov model. In: International Conference on Pattern Recognition, vol. I, pp. 924–927 (2000)
6. Bicego, M., Murino, V.: Investigating hidden markov models' capabilities in 2D shape classification. IEEE Trans. Pattern Anal. Mach. Intell. 26(2), 281–286 (2004)
7. Cai, J., Liu, Z.Q.: Hidden markov models with spectral features for 2D shape recognition. IEEE Trans. Pattern Anal. Mach. Intell. 23(12), 1454–1458 (2001)
8. He, Y., Kundu, A.: 2-D shape classification using hidden markov model. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(11), 1172–1184 (1991)

9. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE 77(2) (1989)
10. Young, S., Odell, J., Ollason, D., Valtchev, V., Woodland, P.: The HTK Book. Cambridge University 1996 (1995)
11. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. Ann. Math. Stat. 41, 164–171 (1970)
12. Juang, B.H., Rabiner, L.R.: The segmental K-means algorithm for estimating parameters of hidden markov models. IEEE Transactions on Acoustics, Speech, and Signal Processing 38(9), 1639 (1990)
13. Forney, G.D.: The Viterbi algorithm. Proceedings of the IEEE 61, 268–278 (1973)
14. Maes, M.: On a cyclic string-to-string correction problem. Information Processing Letters 35, 73–78 (1990)
15. Zangwill, W.I.: Nonlinear Programming. A Unified Approach. Prentice-Hall, Englewood Cliffs (1969)