Structural hidden Markov models: An application to handwritten numeral recognition

D. Bouchaffra* and J. Tan
Department of Computer Science, 131 Dodge Hall, Oakland University, Rochester, MI 48309, USA
E-mails: dbouchaffra@ieee.org; jtan@oakland.edu

Received 31 January 2005
Revised 11 April 2005
Accepted 28 May 2005

Abstract. We introduce in this paper a generalization of the widely used hidden Markov models (HMM's), which we name "structural hidden Markov models" (SHMM). Our approach is motivated by the need of modeling complex structures which are encountered in many natural sequences pertaining to areas such as computational molecular biology, speech/handwriting recognition and content-based information retrieval. We consider observations as strings that produce the structures derived by an unsupervised learning process. *These observations are related in the sense they all contribute to produce a particular structure.* Four basic problems are assigned to a structural hidden Markov model: (1) probability evaluation, (2) state decoding, (3) structural decoding, and (4) parameter re-estimation. We have applied our methodology to recognize handwritten numerals. The results reported in this application show that the *structural* hidden Markov model outperforms the traditional hidden Markov model with a 23.9% error-rate reduction.

Keywords: Hidden Markov models, probabilistic principal component analysis, structural information, stochastic process, handwritten numeral recognition

1. Introduction

Hidden Markov models (HMM's) is a widely used approach that models stochastic processes and sequences in several applications. The relevance of HMM's was first demonstrated in speech processing and recognition in the late 1980's [1–3]. Neighbor areas such as signal processing [4], and handwriting and text recognition [5] have also benefited almost at the same time from these stochastic models. Half a decade later, HMM's spread to many other areas such as image processing and computer vision [6], biosciences [7], control [9], and others. Promising results have been obtained from the use of HMM's in several applications in the aforementioned areas. However, the number of problems where HMM's can be applied is insignificant compared to all the problems a researcher can encounter. In other words, the use of HMM's is rare within the whole spectrum of the scientific community. *The main reason behind*

[*] Corresponding author: Professor Djamel Bouchaffra, Oakland University School of Engineering and Computer Science, 131 Dodge Hall, Rochester MI 48309, USA. Voice: +1 248 370 2242; URL: www.oakland.ed/~bouchaff.

this limitation comes from the fact that HMM's have a clear conceptual framework and the ability to learn statistically, but they are unable to account for structural information of the sequence [10,11]. Because the symbols of an input sequence are assumed to be state conditionally independent, therefore, the hidden Markov models make no use of structure, either topological or conceptual [12].

This lack of structure inherent to standard HMM's has drastically limited the recognition and classification tasks of complex patterns. The reason is that a pattern contains some *relational information* from which it is difficult to derive an appropriate feature vector. *Therefore, the analytical approaches which process the patterns only on a quantitative basis but ignore the inter-relationships between the components of the patterns quite often fails.*

To face this challenge, a few number of approaches that attempt to overcome this lack of syntax have been proposed in the context of HMM's. Fine et al. [13] introduced the hierarchical hidden Markov model (HHMM) that is designed to model domains with hierarchical structures. HHMM's provide several hierarchical classifiers that are *independent* in order to infer a global conclusion. Unfortunately, this inference is too much complicated and takes $O(T^3)$ where T is the length of the sequence, making it impractical for many domains. Cai and Liu's approach integrates the statistical and structural information for unconstrained handwritten numeral recognition. Their method uses macro-states to model pattern structures [14]. They have incorporated statistical and structural information in two different steps. However, in their structural modeling, the information is not extracted from the sequence of input data itself, but obtained via the state position index in the sequence after the pattern is built. Furthermore, their methodology is application-driven and therefore is *very specific*. Zhu and Garcia-Frias proposed two novel generative methods which make use of stochastic context-free grammars [17,22,23] and HMM's respectively to model the end-to-end error profile of radio channels [15]. However, in their approach, the structure is not learned automatically within a single probabilistic framework. Another promising approach that can contribute in building structural hidden Markov models is due to Geman's work in vision. He introduced compositionality as an ability to construct hierarchical representations of scenes, whereby constituents are viewed in an *infinite variety* of relational compositions. Amongst all possible composition rules that embed syntactical information, statistical criteria such as MDL (Minimum Description Length) and Gibbs distribution are being used in order to select the optimal interpretation [16]. This approach is a preliminary attempt to merge statistics with syntax but unfortunately, due to the greedy compositionality process, it is intractable.

We propose in this paper a novel methodology that seeks to analyze and recognize structural components within a whole organized pattern. We named this new paradigm *structural hidden Markov model* (SHMM). *Our approach assumes that the sequence that describes the entire pattern is explained by a single hidden Markov model. This hidden Markov model is extended to contain structural information that are embedded within the pattern. We evaluate the contribution of each component to the entire pattern.* These components are merged together to describe the structure of this pattern. Therefore, the concept of SHMM is different from the HHMM concept since it does not consider different independent HHMM's at different level of the hierarchy.

Our methodolgy is motivated by the fact that a complex system can be naturally viewed as a composition of distinct parts of an organized pattern. *The use of a single HMM that produces the symbols* (*leaves of a SHMM*) *is justified by the fact that these data are of the same type.* For example, a protein structure in the three dimensional space is a composition of structures such as α sheet, β helix, etc. [18], and a digit number can be viewed as a combination of strokes.

The organization of this paper is as follows: Section 2 introduces the concept of a structural hidden Markov model. An application of this concept to handwritten numeral recognition is laid out in Section 3. Finally, the conclusion and the future work are the object of Section 4.

2. The concept of structural HMM

In this section, we introduce a mathematical description of the SHMM concept. In the traditional HMM's, the visible observations are assumed to be *state conditionally independent*. Let $O = (o_1 o_2 \ldots o_T)$ be the observation sequence of length T and $q = (q_1 q_2 \ldots q_T)$ be the state sequence where q_1 is the initial state. Given a model λ, we can write:

$$P(O \mid \lambda) = \sum_{all\ q} P(O, q \mid \lambda) \tag{1}$$

$$P(O, q \mid \lambda) = P(O \mid q, \lambda) \times P(q \mid \lambda), \tag{2}$$

and using state conditional independence, we obtain:

$$P(O \mid q, \lambda) = \prod_{t=1}^{T} P(o_t \mid q_t, \lambda).$$

However, there are several scenarios where the conditional independence assumption doesn't hold. For example, while standard HMM's perform well in recognizing amino acids and consequent construction of proteins from the first level structure of DNA sequences [18,19], they are inadequate for predicting the secondary structure of a protein. The reason for the inadequacy comes from the fact that the same order of amino acid sequences have different folding modes in natural circumstances [7]. Therefore, there is a need to balance the loss incurred by this state conditional independence assumption.

Our idea is that a complex pattern O can be viewed as a sequence of constituents O_i made of strings of symbols interrelated in some way.[1] Therefore, each observation sequence O is not only one sequence in which all observations are conditionally independent, but a sequence that is divided into a series of m strings $O_i = (o_{i_1} o_{i_2} \ldots o_{i_{r_i}})(1 \leqslant i \leqslant m)$. The symbols of a string are related in the sense that they define a local structure S_j of the whole complex pattern. This relationship between symbols helps circumvent the long range dependency problem inherent to traditional HMM's. In fact, one of the major problem of HMM's is that they have great difficulty in learning to capture long range dependencies in a sequence [8]. In this paper, the structure are determined using an unsupervised learning algorithm. The symbols interconnection produces a structure S_j. For example, a cloud of points representing a sequence of observations O_i forms a round shape S_j with a certain probability $P(S_j \mid O_i)$. Similarly a sequence of phonemes produces a word with a certain probability depending on the context. The higher the complexity of a pattern, the higher the number of structures needed to describe this pattern locally. Furthermore, the statistical information is expressed through the probability distribution of the structural information sequence that describes the whole pattern. Figure 1 depicts examples of two structural patterns.

Therefore, if $O = (O_1, O_2, \ldots, O_m) = (o_{1_1} o_{1_2} \ldots o_{1_{r_1}}, o_{2_1} o_{2_2} \ldots o_{2_{r_2}}, \ldots, o_{m_1}, o_{m_2}, \ldots, o_{m_{r_m}})$, (where r_1 is the number of observations in subsequence O_1 and r_2 is the number of observations in subsequence O_2, etc) and $S = (S_1, S_2, \ldots, S_m)$, then the probability of a complex pattern O given a model λ can be written as:

$$P(O \mid \lambda) = \sum_{S} P(O, S \mid \lambda). \tag{3}$$

[1] Any complex pattern can be expressed as a sequence of symbols when modifying the resolution level.

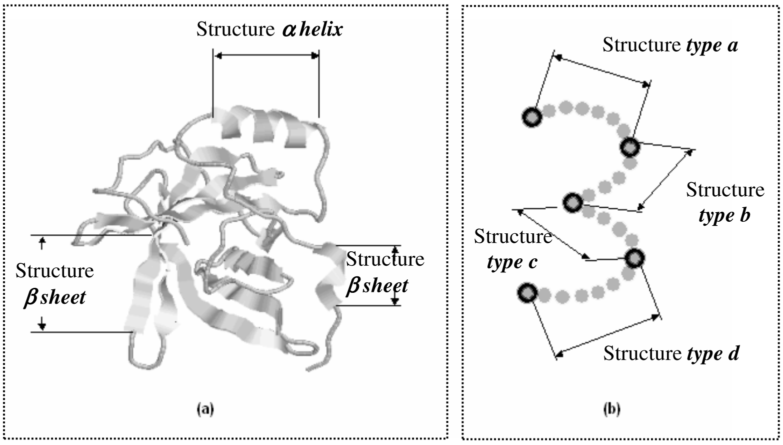


Fig. 1. Two examples of structural pattern: (a) Protein 3D structure, (b) Numeral.

Therefore, we need to evaluate $P(O, S \mid \lambda)$:

$$P(O, S \mid \lambda) = P(O \mid S, \lambda) \times P(S \mid \lambda) \tag{4}$$

$$= P(O_1, O_2, \ldots, O_m \mid S_1, S_2, \ldots, S_m, \lambda) \times P(S_1, S_2, \ldots, S_m \mid \lambda) \tag{5}$$

$$\approx \prod_{i=1}^{m} \left[P(O_i \mid S_1, S_2, \ldots, S_m, \lambda) \times P(S_i \mid S_{i-1}, \ldots, S_m, \lambda) \right]. \tag{6}$$

We have assumed conditional independence of the $O_i^{\prime}s$ with respect to the structure sequence. We also assume that a structure S_i depends only on the observation sequence O_i and the structure probability distribution is a Markov chain of order 1. The reason behind this Markovian assumption comes from cognitive science. In fact, it is well-established that when we perform an object recognition task, our brain relies partly on local interactions between sub-patterns describing these objects [20]. Local interactions can also be expressed statistically by the means of Markovian fields using Gibbs distributions [11,21]. However, as pointed out in the introduction, our approach considers exclusively sequential processes that remains within the context of HMM's. Therefore it is legitimate to estimate Eq. (6) as:

$$\prod_{i=1}^{m} [P(O_i \mid S_i, \lambda) \times P(S_i \mid S_{i-1}, \lambda)]. \tag{7}$$

In order to show how the symbols o_i are inter-related to form a particular structure, we use Bayes' rule in Eq. (7), and obtain:

$$P(O, S \mid \lambda) \approx \prod_{i=1}^{m} \frac{[P(S_i \mid O_i, \lambda) \times P(S_i \mid S_{i-1}, \lambda) \times P(O_i \mid \lambda)]}{P(S_i \mid \lambda)}. \tag{8}$$

The organization of the symbols o_i is introduced mainly through the term $P(S_i \mid O_i)$ since the transition probability $P(S_i \mid S_{i-1})$ does not involve the inter-relationship of the symbols o_i. Besides, the term $P(O_i \mid \lambda)$ of Eq. (8) is viewed as a traditional HMM that involves symbols within O_i. Thus we can define a Structural HMM as follows:

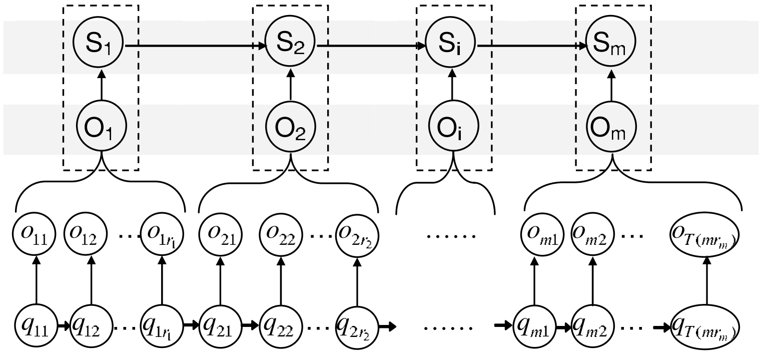


Fig. 2. A graphical representation of a structural hidden Markov model. One single HMM produces the symbols o_i of each subsequence O_i.

Definition 2.1. A structural hidden Markov model is a quintuple $\lambda = (\pi$, A, B, S, D), where:

- π is the initial state probability vector,
- A is the state transition probability matrix,
- B is the state conditional probability matrix of the visible observations,
- S is the posterior probability matrix of a structure given a sequence of observations,
- D is the structure transition probability matrix.

A SHMM is characterized by the following elements:

- **N**, the number of hidden states in the model. We label the individual states as 1, 2, …, N, and denote the state at time t as q_t.
- **M**, the number of distinct observations o_i.
- $\boldsymbol{\pi}$, the initial state distribution, $\pi = \{\pi_i\}$, where $\pi_i = P(q_1 = iatt = 0)$ and $1 \leqslant i \leqslant N$, $\sum_i \pi_i = 1$.
- A, the state transition probability distribution matrix, $A = \{a_{ij}\}$, where: $a_{ij} = P(q_{t+1} = j \mid q_t = i)$, $\sum_j a_{ij} = 1\ \forall i$, where $1 \leqslant i, j \leqslant N$ and $t = 1, \ldots, T$.
- B, the state conditional probability matrix of the observations, B= $\{b_j(k) = P(o_k \mid q_j)$, $\sum_k b_j(k) = 1$, where $1 \leqslant k \leqslant M$ and $1 \leqslant j \leqslant N$.
- **F**, the number of distinct structures.
- S is the posterior probability matrix of a structure given its corresponding observation sequence, S $= P(S_j \mid O_i) = s_i(j)$. For each particular input string O_i, we have: $\sum_j s_i(j) = 1$. Structures are obtained from a data set using an unsupervised learning algorithm.
- D, the structure transition probability matrix.

$$\mathcal{D} = \{d_{ij}\},\ \text{where}\ d_{ij} = P(S_{t+1} = j \mid S_t = i), \sum_j d_{ij} = 1, 1 \leqslant i, j \leqslant F.$$

Figure 2 depicts a representation of a structural hidden Markov model.

2.1. Problems assigned to a SHMM

There are four problems that are assigned to a SHMM:

(i) Probability evaluation, (ii) State decoding, (iii) Structural decoding, and (iv) Parameter re-estimation.

- **Probability evaluation:** Given a model λ and an observation sequence $O = (O_1, \cdots, O_m)$, we evaluate how well does the model λ match O. This problem has been discussed in Section 2.
- **State decoding:** In this problem, we attempt to determine the state sequence that "best" explain the input sequence of observations. This problem is similar to problem 2 of the traditional HMM and can be solved using Viterbi algorithm as well.
- **Structural decoding:** This is the most important problem since we attempt to determine the "optimal structure of the model".
- **Parameter re-estimation:** In this problem, we try to optimize the model parameters $\lambda = (\pi$, A, B, S, D).

2.1.1. *Probability evaluation*

The evaluation problem in SHMM consists of evaluating the probability for the model $\lambda = (\pi$, A, B, S, D) to produce the sequence O. From Eq. (8), this probability can be expressed as:

$$\begin{aligned} P(O \mid \lambda) &= \sum_{S} P(O, S \mid \lambda) \\ &\approx \sum_{S} \left\{ \prod_{i=1}^{m} \left[\frac{s_i(i) \times d_{i-1i}}{P(S_i)} \times \sum_{q} \pi^i_{q1} b^i_{q1}(o_1) a^i_{q1q2} b^i_{q2}(o_2) \ldots a^i_{q(r_i-1)q_{r_i}} b^i_{q_{r_i}}(o_{r_i}) \right] \right\}, \end{aligned} \quad (9)$$

where the superscript i indicates that the a^i, b^i, and π^i come from the local structure S_i. However, since the SHMM concepts assumes that the entire sequence O is produced by a single HMM that has been extended to contain the matrices S and $\mathcal{D}$, therefore we have only one matrix A and one matrix B for all the components O_i.

2.1.2. *State decoding*

The state decoding problem consists of determining the optimal state sequence $q^* = arg \max_{q}(P(O_i, q \mid \lambda))$ that best "explains" the sequence of symbols within O_i. It can be computed using Viterbi algorithm as in traditional HMM's.

2.1.3. *Structural decoding*

The structural decoding problem consists of determining the optimal structure sequence $S^* =< S^*_1, S^*_2, \ldots, S^*_t >$ such that: $S^* = \max_{\mathcal{S}} P(O, S \mid \lambda)$.

We define:

$$\delta_t(i) = \max_{\mathcal{S}} P(O_1, O_2, \ldots, O_t, S_1, S_2, \ldots, S_t = i \mid \lambda)$$

that is, $\delta_t(i)$ is the highest probability along a single path, at time t, which accounts for the first t strings and ends in structure i. Then, by induction we have:

$$\delta_{t+1}(j) = \left[\max_{i} \delta_t(i) d_{ij} \right] s_{t+1}(j) \frac{P(O_{t+1})}{P(S_j)}. \quad (10)$$

Similarly, this latter expression can be computed using *Viterbi* algorithm. However, we estimate δ in each step *through the structure transition probability matrix.* This optimal sequence of structures describes the structural pattern piecewise.

2.1.4. Parameter Re-estimation

Many algorithms have been proposed to re-estimate the parameters for traditional HMM's. For example, Petar and his colleagues [24] used "Monte-Carlo Markov Chain" sampling scheme. In the structural HMM paradigm, we have used a "Forward-backward maximization" algorithm to re-estimate the parameters contained in the model λ. We used a bottom-up strategy that consists of re-estimating $\{\pi_i\}$, $\{a_{ij}\}$, $\{b_j(k)\}$ in a first phase and then re-estimating $\{s_j(k)\}$ and $\{d_{ij}\}$ in a second phase. Let's define:

- $\xi_r(u,v)$ as the probability of being at structure u at time r and structure v at time $(r+1)$ given the model λ and the observation sequence O. We can write:

$$\xi_r(u,v) = P(q_r = u, q_{r+1} = v|\lambda, O) = \frac{P(q_r = u, q_{r+1} = v, O|\lambda)}{P(O|\lambda)}. \tag{11}$$

Using Bayes formula, we can write:

$$\xi_r(u,v) = \frac{P(O_1 O_2 ... O_r, q_r = u \mid \lambda) d_{uv} P_v(O_{r+1}) P(O_{r+2} O_{r+3} \dots O_T \mid q_r = v, \lambda)}{P(O_1 O_2 \dots O_T \mid \lambda)}. \tag{12}$$

Then we define the following probabilities:

- $\alpha_r(u) = P(O_1 O_2 \dots O_r, q_r = u \mid \lambda)$
- $\beta_r(u) = P(O_{r+1} O_{r+2} \dots O_T \mid q_r = u, \lambda)$
- $P_v(O_{r+1}) = P(q_{r+1} = v \mid O_{r+1}) \times \dfrac{P(O_{r+1})}{P(q_{r+1} = v)}$,

therefore:

$$\xi_r(u,v) = \frac{\alpha_r(u) d_{uv} s_{r+1}(v) P(O_{r+1}) \beta_{r+1}(v)}{P(O_1 O_2 \dots O_T \mid \lambda) P(q_{r+1} = v)}. \tag{13}$$

We need to compute the following:

- $P(O_{r+1}) = P(o^1_{r+1} \dots o^k_{r+1} \mid \lambda) = \sum_{all\ q} P(O_{r+1} \mid q, \lambda) P(q \mid \lambda) = \sum_{q_1 \cdots q_T} \pi_{q_1} b_{q_1}(o_1) a_{q1q2} \dots b_{q_k}(o_k)$
- $P(q_{r+1} = v) = \sum_j P(q_{r+1} = v \mid q_r = j)$
- $P(O_1 O_2 \dots O_T \mid \lambda)$. This term requires π, A, B, S, D. The parameters π, A, and B can be re-estimated as in traditional HMM. In order to re-estimate S and D, we define:

$$\gamma_r(u) = \sum_{v=1}^{N} \xi_r(u,v). \tag{14}$$

Then we compute the improved estimates of $s_v(r)$ and d_{uv} as:

$$\hat{d}_{uv} = \frac{\sum\limits_{r=1}^{T-1} \xi_r(u,v)}{\sum\limits_{r=1}^{T-1} \gamma_r(u)}, \tag{15}$$

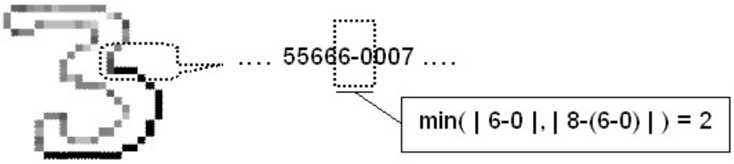


Fig. 3. The strokes are separated where the chain code directions change significantly.

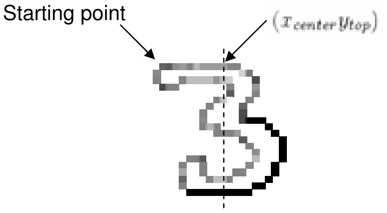


Fig. 4. The criterion to select the chain code starting point.

$$\hat{s}_v(r) = \frac{\sum\limits_{r=1, O_r=v_r}^{T} \gamma_r(v)}{\sum\limits_{r=1}^{T} \gamma_r(v)}. \tag{16}$$

From Eq. (16), we derive:

$$\hat{s}_r(v) = \hat{s}_v(r) \times \frac{\hat{P}(q_r = v)}{\hat{P}(O_r)}. \tag{17}$$

We calculate improved $\xi_r(u,v)$, $\gamma_r(u)$, $\hat{d}_{uv}$ and $\hat{s}_r(v)$ repeatedly until some convergence criterion is achieved.

3. Application: Handwritten numeral recognition

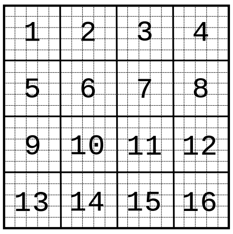
We have applied the concept of SHMM in handwritten numeral recognition. Usually, a handwritten numerals recognition system includes three parts: image processing, feature extraction, and classification. The image processing phase was skipped since we have used the MNIST database which is a subset of a larger NIST database. All the digits in the MNIST repository have been size-normalized to fit in a 20*20 pixel box, and centered in a 28*28 pixel image. The training set contains 60,000 digits and the testing set 10,000 digits.

3.1. Feature extraction

It is well known that the performance of a handwritten numeral recognition system depends largely on the feature extraction phase. In our application, we used the standard 8-directions chain code to represent the closed outer contours of the digits [25]. The extracted features are sequences of integers from 0 to 7. However, the chain code method has its own weaknesses. It is not capable of: (i) performing a good corner detection; (ii) detecting sharp boundary changes, and thus not capable of capturing structural

Table 1
The structure numbers and their models

Digits	1	2	3	4	5	6	7	8	9	0
Number of structures	11	16	17	13	17	15	14	19	14	16



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Fig. 5. Hidden states and their corresponding cells.

information. To solve this problem, we assigned a sequence of structures to represent the digit contours. We used an unsupervised learning algorithm to extract the structures automatically from its chain code sequence. The first step of the structure extraction is to determine all *strokes*. The strokes are separated when a significant change of a contour direction of a chain code occurs. In other words, if the difference (which is the minimum value of clockwise and counterclockwise change) between two successive chain code directions is no less than a preset threshold, then a new stroke is created. An example showing how to separate two successive strokes is illustrated in Fig. 3. The chain code of the contour segment in the round corner rectangle box is "....556660007....". We can see that the biggest chain code difference is between the "6" and "0" in the middle. This difference is $\min(|6-0|, |8-(6-0)|) = 2$ which is no less than our preset threshold value "2". Thus, we consider "....55666" as one stroke, and "0007...." as another stroke. After we have extracted all strokes, we clustered them using the probabilistic principal component analysis technique (PPCA) [26] and assigned each cluster a label. Finally, each cluster is considered as a structure and each stroke has different probabilities belonging to different clusters. We calculated a "weighted mean" value from the strokes for each structure S_i using the following equation:

$$\mu_i = \frac{\sum\limits_{V_k \in S_i} P(S_i \mid V_k) \times V_k}{\sum\limits_{V_k \in S_i} P(S_i|V_k)} \tag{18}$$

where V_k is the feature vector of each stroke O_i in structure S_i coming from the classification of PPCA. The weighted mean μ_i represents an approximation of each structure S_i by a representative vector. This vector representation of a structure enables to perform comparison between structures. A numeral is thus expressed as a composition of stroke structures. Table 1 shows the number of structures found for all 10 digits. Because SHMM requires the input features to be sequential, we have chosen a point on the contour as the starting point of the chain code. The criterion to select the starting point is as follows: We first choose a point $(x_{center} y_{top})$, where x_{center} is the center of the image, and y_{top} is the top "y coordinate" of the contour point along the vertical center line. Then we traverse along the contour counterclockwise until we meet the beginning point of a stroke for the first time. We consider this beginning point of the stroke as the starting point of the chain code. Figure 4 depicts an example of the starting point.

Table 2
Comparison of performance with SHMM and HMM

Model	Accuracy (%)	Error-rate (%)	Error Reduction (%) (SHMM vs HMM)
SHMM	96.5	3.5	
HMM	95.4	4.6	23.9

3.2. Training the SHMM

The training of a structural hidden Markov model is an iterative process that seeks to maximize the probability that the SHMM accounts for the training sample sequences. Since all the digits were size-normalized to fit in a 20*20 pixel box, we placed a 4 by 4 mesh on the box. The grid lines of the mesh are evenly set, so that there are 16 cells in total and each cell covers 5*5 $=$ 25 pixels. The 16 cells correspond to the hidden states as illustrated in Fig. 5. Thus we have obtained 16 states for each of the 10 SHMM's from "0" to "9" respectively. As outlined in Section 2.1.4, We have used the Forward-backward algorithm to estimate the matrix A, B, S, and D. While the Baum-Welch algorithm itself is well-defined, initialization of the SHMM is much tricky. To initialize A, we set each $a_{i,j}$ to 0 if cell $i \neq j$ and they are not neighbors, otherwise we set it to the value of 1 divided by (*the number of neighbors of cell* i) $+$ 1. For example, we set $a_{1,1} = a_{1,2} = a_{1,5} = a_{1,6} = \frac{1}{4}$ and $a_{1,j(j \neq 1,2,5,6)} = 0$. The initialization of B is much simpler. We just assign all $b_j(k)$'s the value $\frac{1}{8}$. S and D were initialized as empty matrices "empty" means there is no row and column initially. As the training process is going, we'll enrich the matrices by inserting rows and columns. The training process of the SHMM in this application is described using the following algorithm:

```
Begin:
 Initialize ($\pi$, A, B, S, D).
 $C_S = \emptyset$. ($C_S$ is a set of structures (clusters) represented by feature vectors)
 $C_O = \emptyset$. ($C_O$ is a set of strokes (subsequences) represented by feature vectors)
 Set a threshold value $\rho$. Set a convergence criterion $\theta$. $z \leftarrow 0$.
 For Each input image $x$ in the training set
   Perform image processing and extract its outer contour "chain code" sequence $O$.
   Segment $O$ to a sequence of subsequences as $\{O_i\}$ and extract their feature vectors $V_i$'s. (find all strokes)
   $n_o \leftarrow$ number of $O_i \notin C_O$. Insert all $O_i \notin C_O$ in $C_O$.
   Cluster $\{O_i\}$ in $C_O$ using PPCA and assign $P(S_i \mid O_i)$ where $\sum_{allS_i} P(S_i \mid O_i) = 1$.
   $n_s \leftarrow 0$.
   For Each structure $S_i$
      $\mu \leftarrow \frac{\sum_{V_k \in S_i} P(S_i \mid V_k) V_k}{\sum_{V_k \in S_i} P(S_i \mid V_k)}$. (compute the mean strokes feature vector that represents $S_i$)
      If $C_S = \emptyset$, Then (if $C_S$ is not empty, insert a new structure)
         $C_S \leftarrow C_S \cup \{S_i\}$, $n_s \leftarrow n_s + 1$.
      Else
         $S^* \leftarrow arg \min_{S' \in C_S} \|\mu' - \mu\|_2$ and $\sigma \leftarrow \|\mu^* - \mu\|_2$.
           (find the closest structure to the current one, where $\mu^*$ characterizes $S^*$)
         If $\sigma < \rho$, Then (structure already exists)
            Combine $S_i$ with structure $S^*$ to form $S''$ and compute $\mu''$ for $S''$.
              (the combination is done by combining the two feature vector clusters that of two structures
                $S_i$ and $S^*$, and thus creating a new structure $S''$.)
            $C_S \leftarrow (C_S \setminus \{S'\}) \cup \{S''\}$. (remove $S'$ from $C_S$ and insert $S''$).
        **Else**
            $C_S \leftarrow C_S \cup \{S_i\}, n_s \leftarrow n_s + 1$.
        **EndIf**
    **EndIf**
**EndFor.**
Add $n_o$ rows and $n_s$ columns to S with randomly initialized numbers.
Add $n_s$ rows and $n_s$ columns to D with randomly initialized numbers.
**Repeat**
    $z \leftarrow z + 1$.
    Compute $\hat{\pi}^z$, $\hat{a}^z$, $\hat{b}^z$, $\hat{s}^z$ and $\hat{d}^z$ from $\hat{\pi}^{z-1}$, $\hat{a}^{z-1}$, $\hat{b}^{z-1}$, $\hat{s}^{z-1}$ and $\hat{d}^{z-1}$ using Equations 15 to 17.
    $\hat{\pi}_i^z \leftarrow \hat{\pi}_i^{z-1}$, $a_{ij}^z \leftarrow \hat{a}_{ij}^{z-1}$, $b_j^z(k) \leftarrow \hat{b}_j^{z-1}(k)$, $s_j^z(k) \leftarrow \hat{s}_j^{z-1}(k)$, $d_{ij}^z \leftarrow \hat{d}_{ij}^{z-1}$.
**Until** $\max_{i,j,k} \left[\pi_i^z - \pi_i^{z-1}, a_{ij}^z - a_{ij}^{z-1}, b_j^z(k) - b_j^{z-1}(k), s_j^z(k) - s_j^{z-1}(k), d_{ij}^z - d_{ij}^{z-1},\right] < \theta$.
            (convergence achieved)
$\pi_i \leftarrow \pi_i^z$, $a_{ij} \leftarrow a_{ij}^z$, $b_j(k) \leftarrow b_j^z(k)$, $s_j(k) \leftarrow s_j^z(k)$, $d_{ij} \leftarrow d_{ij}^z$. (update $\pi$, A, B, S, D)
**EndFor.**
**End.**
```

3.3. *Classification results*

The testing phase is conducted on the MNIST test data set. Given a test sample image x of a numeral, we extracted the chain code string of its contour. Then we determined the strokes from which we derived all structures assigned to x. Because a stroke has different probabilities assigned to structures, we used all possible structures as input to the 10 models. All ten SHMM's were tested using x as input. We then determined the model λ^* that maximizes $P(O|\lambda_i)$ and assign its numeral class to the input x.

The accuracy was computed as the number of correctly recognized digit divided by the testing data set size. The error-rate is equal to $1 - accuracy$ and the error-rate reduction is obtained by:

$$\frac{(\text{error rate of HMM}) - (\text{error rate of SHMM})}{(\text{error rate of HMM})} 100\%. \tag{19}$$

We have compared the SHMM approach with the Hidden Markov Model (HMM) classification technique. The training of both of them was coded using MATLAB. Table 2 shows the performance comparison between SHMM and HMM. The error-reduction of SHMM vs.

HMM is 23.9% which is a significant improvement.

4. Conclusion and future work

We have presented a novel mathematical paradigm that extends the traditional HMM in order to capture and model structural information present in the data. This work is an extension and a complete version of the model introduced in [27]. SHMM's generalize HMM's and therefore provide a partial answer to three fundamental problems that arise in complex sequence modeling: (i) SHMM's are capable to model structural and statistical information within a single probabilistic learning scheme, while maintaining a computational tractability. (ii) SHMM's correlate two consecutive structures, thus reducing the state conditional independence effect in traditional HMM's. (iii) SHMM's are very well-adapted to sequences of patterns of the same type, therefore they maintain a low computational cost. The reason is explained by the fact that the structures are produced by subsequences of symbols generated by a single HMM. However, because the symbols of subsequences are emanating from *a single Markovian process*, SHMM's lack the ability to handle statistical inhomogeneities relevant in applications involving different modalities. Experimental results show that the concept of structural HMM is promising since it has outperformed the HMM concept on a standard pattern recognition task.

Our future work is twofold:

- compare the modeling power of SHMMs with that of maximum entropy models or conditional random fields,
- investigate the SHMM concept where observations are continuous,
- extend the SHMM concept to a factorial SHMM so that the structures can be explained by multiple processes (or multiple causes) rather than by a single one.

References

[1] L. Rabiner and B.H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[2] M.J.F. Gales, Cluster adaptive training of hidden markov models, IEEE Transactions on Speech and Audio Processing **8**(4) (July 2000).

[3] I. Sanches, Noise-compensated hidden markov models, *IEEE Transactions on Speech and Audio Processing* **8**(5) (Sept. 2000).

[4] G. Fan and X.G. Xia, Improved hidden markov models in the wavelet-domain, *IEEE Transactions on Signal Processing* **49**(1) (Jan. 2001).

[5] D. Bouchaffra, V. Govindaraju and S.N. Srihari, Postprocessing of recognized strings using nonstationary markovian models, *IEEE Transactions: Pattern Analysis and Machine Intelligence PAMI* **21**(10) (October 1999).

[6] J. Li, A. Najmi and R.M. Gray, Image classification by a two-dimensional hidden markov model, *IEEE Transactions on Signal Processing* **48**(2) (Feb. 2000), 517.

[7] K. Asai, S. Hayamizu and H. Handa, Prediction of protein secondary structures by hidden markov models, *Computer Application in the Biosciences* (*CABIOS*) **9**(2) (1993), 141–146.

[8] Y. Bengio and P. Fransconi, Diffusion of context and credit information in Markovian models, *Journal of Artificial Intelligence Research* **3** (1995), 249–270,

[9] D. Hernandez-Hernandez, S.I. Marcus and P.J. Fard, Analysis of a risk-sensitive control problem for hidden markov chains, *IEEE Transactions on Automatic Control* **44**(5) (May 1999), 1093.

[10] R. Duda, P. Hart and D. Stork, *Pattern Classification*, (Wiley, New York), 2001.

[11] B.D. Ripley, *Pattern Recognition and Neural Networks*, (Cambridge University Press), 1996.

[12] M.C. Gemignani, *Elementary Topology*, (Second edition), Dover Publications, Inc, NY, 1990.

[13] S. Fine, Y. Singer and N. Tishby, The hierarchical hidden markov model: analysis and applications, *Machine Learning* **32**(41) (1998).

[14] J. Cai and Z.Q. Liu, Integration of structural and statistical information for unconstrained handwritten numeral recognition, *IEEE Transactions: Pattern Analysis and Machine Intelligence*, *PAMI* **21**(3) (March 1999).

[15] W. Zhu and J.G. Frias, Stochastic context-free grammars and hidden markov models for modeling of bursty channels, *IEEE Transactions on Vehicular Technology* **53**(3) (May 2004).

[16] S. Geman, E. Bienenstock, S. Geman and D. Potter, "Compositionality, MDL Priors, and Object Recognition", Internal Report, Division of Applied Mathematics, Brown University, 2004.

[17] K.S. Fu, *Syntactic Pattern Recognition and Applications*, Prentice-Hall, Englewood Cliffs, N.J., 1982.

[18] A. Krogh, M. Brown, I.S. Mian, K. Sjolander and D. Haussler, Hidden markov models in computational biology: applications to protein modeling, *J. Mol. Biol.* **235** (1994), 1501–1531.

[19] S.R. Eddy, Profile hidden markov models, *Bioinformatics* **14**(9) (1998), 755–763.

[20] J.A. Fodor and Z.W. Pylyshyn, Connectionism and cognitive architecture: A critical analysis, *Cognition* **28** (1988), 3–71.

[21] F. Bartolucci and J. Besag, A recursive algorithm for markov random fields, *Biometrika* **89**(3) (2002), 724–730.

[22] D. Jurafsky, C. Wooters, J. Segal, A. Stolcke, E. Fosler, G. Tajchman and N. Morgan, *Using a Stochastic Context-Free Grammar as a Language Model for Speech Recognition*, In Proc. ICASSP'95 (pp. 189–192), 1995.

[23] A. Stolcke, An efficient probabilistic context-free parsing algorithm that computes prefix probabilities, *Computational Linguistics* **21**(2) (1995), 165–201.

[24] M.D. Petar and C. Joon-Hwa, An MCMC sampling approach to estimation of nonstationary hidden markov models, *IEEE Transactions on Signal Processing* **50**(5) (May 2002).

[25] H. Freeman, On the encoding of arbitrary geometric configurations, *IRE Trans. Electron. Comput.* **EC-10** (June 1961), 260–268.

[26] Y. Mori, K. Honda, A. Kanda and H. Ichihashi, A unified view of probabilistic PCA and regularized linear fuzzy clustering, *Proceedings of the International Joint Conference on Neural Networks* **1** (July 2004), 20–24.

[27] D. Bouchaffra and J. Tan, *The Concept of Structural Hidden Markov Models: Application to Mining Customers' Preferences for Automotive Designs*, 17-th International Conference on Pattern Recognition (ICPR), 23–26 August, 2004, Cambridge, United Kingdom.