Pattern Recognition 43 (2010) 2590–2607

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Embedding HMMs-based models in a Euclidean space: the topological hidden Markov models

Djamel Bouchaffra *

Department of Mathematics and Computer Science, Grambling State University, LA 71245, USA

ARTICLE INFO

Article history:
Received 6 May 2008
Received in revised form
18 January 2010
Accepted 27 January 2010

Keywords:
Structural hidden Markov models
Structural decoding
Topological decoding
Object contour representation
Protein fold recognition
5×2-fold cross validation paired t-test of hypothesis
Chain code representation
Handwritten numeral recognition

ABSTRACT

Current extensions of hidden Markov models such as structural, hierarchical, coupled, and others have the power to classify complex and highly organized patterns. However, one of their major limitations is the inability to cope with *topology*: When applied to a visible observation (VO) sequence, the traditional HMM-based techniques have difficulty predicting the n-dimensional shape formed by the symbols of the VO sequence. To fulfill this need, we propose a novel paradigm named "topological hidden Markov models" (THMMs) that classifies VO sequences by embedding the nodes of an HMM state transition graph in a Euclidean space. This is achieved by modeling the noise embedded in the shape generated by the VO sequence. We cover the first and second level topological HMMs. We describe five basic problems that are assigned to a second level topological hidden Markov model: (1) sequence probability evaluation, (2) statistical decoding, (3) structural decoding, (4) topological decoding, and (5) learning. To show the significance of this research, we have applied the concept of THMMs to: (i) predict the ASCII class assigned to a handwritten numeral, and (ii) map protein primary structures to their 3D folds. The results show that the second level THMMs outperform the SHMMs and the multi-class SVM classifiers significantly.

1. Introduction

The concept of hidden Markov models (HMMs) has been introduced in the sixties by Baum and his colleagues [1]. It is a widely used approach that models time series problems from a statistical view. The real milestone of HMMs occurred when they were applied to speech processing and recognition in the late 1980s [2,3]. Related areas such as signal processing [4,5], and handwriting and text recognition [6–8] have also exploited the resources of these stochastic models. Half a decade later, HMMs spread to many other areas such as image processing, computer vision [9], biosciences [10], and control [11]. Promising results have been obtained from the use of HMMs in several applications in the aforementioned areas. However, the number of problems that HMMs can model is insignificant compared to all problems one may encounter. In other words, the use of HMMs by practitioners remains scarce. The main reason behind this limitation is explained by the fact that HMMs are unable to: (i) account for long-range dependencies which unfold structural[1] information and (ii) capture topological features [12] such as the shape[2] formed by the visible observation (VO) sequence. Because the traditional HMMs modeling is based on the hidden state conditional independence assumption of the visible observations, therefore, HMMs make no use of structure. Furthermore, the fact that the HMM state transition graph is not embedded in a Euclidean space, therefore HMMs make no use of topology. *This lack of structure and topology inherent to standard HMMs has drastically limited the shape recognition task of complex objects.*

To overcome the lack of structure inherent to the traditional HMMs, a few number of approaches have been proposed in the literature. The hierarchical hidden Markov models (HHMMs) introduced in [13] are capable to model complex multi-scale structure which appears in many natural sequences. However, the original HHMM's algorithm is rather complicated since it takes $O(T^3)$ time, where T is the length of the sequence, making it impractical for many domains. To decrease the complexity of the HHMM's algorithm, Murphy and Paskin showed that an HHMM is a special kind of dynamic Bayesian network (DBN), and thereby derive a much simpler algorithm whose complexity is $O(T)$ [14]. This connection between HHMMs and DBNs enabled the complexity of the basic HHMM's algorithm to be reduced further.

The structural hidden Markov models (SHMMs) introduced in [15] offer a methodology that automatically identifies the different constituents of a VO sequence. These constituents known as "local structures" are computed via an equivalence relation defined in the space of the VO subsequences. Other

* Tel.: +1 586 744 3184; fax: +1 318 274 6388.
E-mail address: dbouchaffra@ieee.org

[1] From "structure" which is the way in which parts are arranged, or put together to form a whole.

[2] A shape is any subset $S \subset \Re^n$ with a boundary ∂S, restricted to subsets homeomorphic to a ball.

0031-3203/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2010.01.022

graphical models such as "coupled HMMs" (CHMMs) [16], factorial HMMs (FHMMs) [17], "event-coupled HMMs" (ECHMMs) [18] and "input–output HMMs" (IOHMMs) [19] that illustrate different architectures have also been proposed in the machine learning community to enhance the capabilities of the standard HMMs. Nevertheless, this generalization of the hidden Markov models to capture local structures *did not address the shape modeling problem of the VO sequence. As far as we are aware, the embedding of topological features* (*e.g.*, *shapes*) *of these local structures within HMMs has not been addressed in the literature*.

An other different approach that contributes in building structures is due to Geman's work in vision. He introduced the "compositionality" operation 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 contain syntactical information, statistical criteria such as MDL (minimum description length) and Gibbs distribution have been used to select the optimal interpretation [20]. However, even if this approach unfolds the optimal scene in a tractable manner, it does not reveal the underlying shape of the objects of the scene.

We propose in this paper *a machine learning paradigm that extends the traditional HMMs by embedding the nodes of the state transition graph in a Euclidean space* [21]. This action allows the recognition of objects that exhibit shapes. This new paradigm entitled *topological hidden Markov models* (THMMs) extends the traditional concept of HMMs by: (i) unfolding the constituents of the entire VO sequence and (ii) capturing their shapes. The first level THMMs extracts the global shape formed by the VO sequence. However, the second level THMMs decomposes the entire VO sequence into segments before capturing their local shapes.

There are several applications where THMMs can be applied: A first one would be in speech recognition where the pitch contour (rise and fall of the voice pitch) assigned to some speech units (phonemes, syllables) groupings will be extracted to provide complementary information about the uttered phrase. We believe that the fusion of a locale and a global analysis of the speech signal will be able to enhance the speech recognition task. A second application would be to classify celestial objects based on morphological features. It is well known that the ages of galaxies are explained in part by the shape formed by their constituents (large scale aggregates of stars, gas and dust). The galaxy classification task will certainly leapfrog our understanding about the origin of the universe. A third application consists of predicting a protein 3D fold known as tertiary structure given its primary structure (linear sequence of amino acids). Finally, THMMs can be helpful in remote sensing images such as pollution control, crop inventory that involves monitoring and management over a wide agricultural area or seismic wave analysis for earthquake prediction.

The organization of this paper is as follows: Section 2 clarifies the notion of VO sequence through several examples from different applications. Section 3 depicts the topological mapping between the VO sequence and the shape it depicts. Section 4 provides a brief description of the traditional HMMs. The structural hidden Markov model formalism is the object of Section 5. The novel concept of topological hidden Markov models is introduced in Section 6. We cover the first level THMMs, the optimal segmentation problem, and the second level THMMs. Two applications are presented in Section 7. Finally, the conclusion is laid in Section 8.

2. The visible observation sequence

The notion of *visible observation* sequence has been used in many different contexts in the pattern recognition and machine learning community. However, a rigorous definition and the scope of this notion have been often overlooked; they have rarely been addressed thoroughly by researchers. We define a VO sequence as a flow of symbols ordered by time. However, we define a *unit of information* (UNIF) as a shape formed by a group of symbols. If the entire VO sequence has a shape, therefore its shape represents a UNIF that we call *object*. However, if the VO sequence is made of subsequences that possess shapes, therefore each shape is by itself a UNIF. In this case, the sequence of UNIF's obtained represents an entire object. The representation of the UNIF shape is projected into a Euclidean space. A UNIF can unfold only through a *meaningful* organization of the VO sequence. In other words, not all VO sequences constitute a UNIF but only those which disclose structural constituents of the observed object. We introduce some applications from different areas that are intended to clarify the notions of VO and UNIF. A first application would consist of classifying the structure of minerals based on the topology of the bonds that link the atoms in the crystal. For example, the butane gas linear formula "CHHHCHHCHHCHHH" represents a VO sequence; the two symbols "C" and "H" located at different positions span the entire observation sequence. However, the same formula can be written in a more informative way as a sequence of UNIFs: "CH_3–CH_2–CH_2–CH_3". In this formulation, the shapes of the structural parts of the butane which are "CH_3" and "CH_2" are emphasized. A UNIF in the butane gas molecule is the shape associated to either the subsequence "CH_3" or "CH_2". *The UNIF's are certain rearrangements of their constituents that produce shapes* (*refer to* Fig. 1).

A second application is in the area of handwriting recognition: it consists of mapping handwritten word sequences into their ASCII representations. A handwritten word sequence (or script) such as: "The quick brown fox jumps over a sleazy dog" is viewed as a sequence of pixels. However, after several data processing phases including word segmentation, the VO sequence unfolds. Each isolated character can be categorized as one of the five classes "Ascender" (A), "Descender" (D), "Median" (M), "Both Ascender–Descender" (B), and "Space" (S). These classes used in the document analysis area are usually predetermined via an unsupervised clustering algorithm. Since the first handwritten character of this script that corresponds to the letter "T" is rising up (or moving upward), therefore it is depicted as "A". The second handwritten character assigned to the letter "h" is also perceived as "A", whereas the third character assigned to "e" is depicted as "M" since it remains in the median line of the handwritten script. Following the same procedure, we can finally represent the script: "The quick brown fox jumps over a sleazy dog" as the VO sequence "AAMSDMMMASAMMMMSAMMSDMMDMSMMMMSAAMSMAMMMDSAMD". This VO sequence transcription is not unique; it is simply one possible manner of globally discerning handwritten phrases. However, *it is worth to underscore that a particular "instantiation" of a group of symbols made of A, M, D and their connection produces a handwritten word with a shape. Because a word has a potential to convey a meaning, it represents a UNIF* (*refer to* Fig. 2). A shape of a handwritten word can be extracted

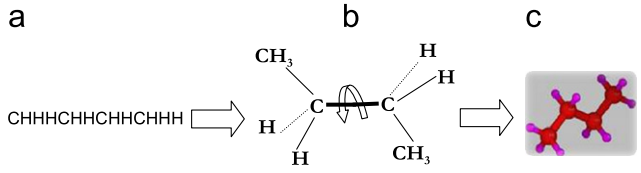


Fig. 1. The butane molecule where: (a) represents its VO sequence O = CHHHCHHCHHCHHH. Each symbol is either a carbon or a hydrogen atom. Part (b) depicts the VO sequence instantiation into UNIF's and (c) outlines UNIF shapes captured by their external contours.

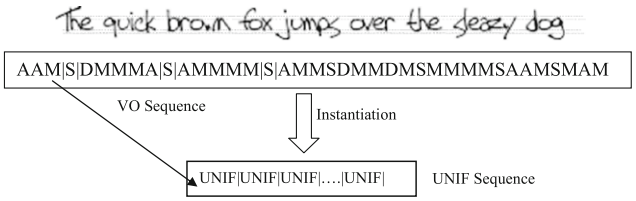


Fig. 2. The VO sequence assigned to a handwritten phrase. A certain instantiation of symbols within each segment in the VO sequence constitutes the UNIF's which are the handwritten words.

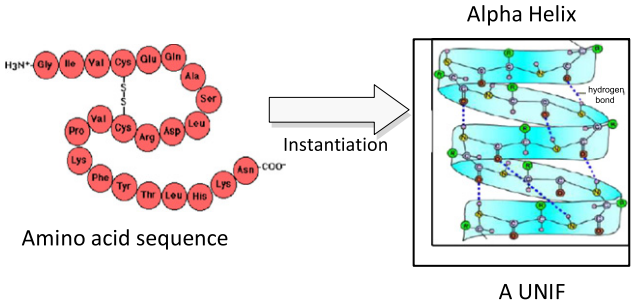


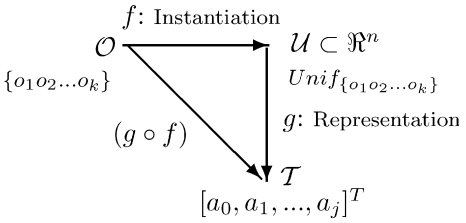
Fig. 3. The VO sequence of amino acids (protein primary structure) is mapped to its UNIF (Alpha Helix: protein secondary structure) via a particular instantiation (through stretching) of its primary structure. "Hydrophobicity" is thought to be one of the primary forces driving the folding of secondary structures.

using for example a horizontal or a vertical histogram of the scanned image of the handwritten word. This histogram maps the row (or column) number (x-axis) to the number of pixels encountered during a horizontal or a vertical scanning of the bitmap image of the handwritten word (y-axis).

A third application would consist of predicting the protein 3D fold (or conformation) given its primary structure. In this application, the sequence of amino acids "GLY, ILE, VAL, CYS, GLU, GLN, ALA…" known as the primary structure is the VO sequence $O = o_1, o_2,\ldots,o_T$ of the protein 3D fold (refer to Fig. 3). The UNIF's are the protein secondary structures: *Alpha-Helix* (there are other forms of helices, but they are less stable and therefore rare), *the Beta-Sheet*, *the Beta-Turn*, and others. They represent 3D forms of local segments of proteins. However, they do not describe specific atomic positions in 3D space, which are considered to be tertiary structure. One possible instantiation of the amino acid sequence produces a protein secondary structure sequence. In the case of other applications such as speech recognition, the VO sequence can be assigned to the sequence of phonemes that composes the uttered phrase. Since a word is made of phonemes tied together in a certain organized manner, a possible UNIF sequence in this case corresponds to the pitch contour sequence assigned to the word sequence uttered. A sequence of UNIF's reveals an organized entity: some words are used and interrelated following some linguistic rules to convey meaning in language. Nevertheless, it is noteworthy that there are many other possible ways to define a VO sequence and a UNIF sequence.

3. The topological mapping: projection onto a Euclidean space

This section represents the key of the THMM's approach since it brings forward the *topological mapping* between a VO sequence and the shape it forms. External contour points assigned to UNIF's capture the shape of objects such as a 3D mineral structure, a handwritten word sequence, or a protein 3D fold. The thrust in this task is to investigate how the observation symbols are seamlessly tied together and transformed to form a meaningful structure of an object. *This exploration whose main goal is to bridge the gap between continuous and discrete structures is fundamental to the pattern recognition and machine learning community*. In the protein fold application, 3D coordinates points of amino acid atoms in the protein are made available in dedicated repositories, and therefore protein shape extraction becomes feasible by computing the protein external contour. However, in order to consider the shape information during the prediction (or classification) task of any visible observation sequence O, one has to determine a mapping between a segment of the VO sequence and the contour of its UNIF. We assume that the VO sequence selected possesses a "meaningful" structure. We first map through a function f the VO sequence to its UNIF: this mapping is called a "Sequence Instantiation" since a UNIF can be viewed as an instantiation of a VO sequence. We then, map through a function g the UNIF to its shape using a contour representation technique. A *Fourier* or a *Wavelet* coefficient vector $[a_0, a_1,\ldots,a_j]^T$ describing the external contour is computed in this phase. This mapping is called a "Shape Representation". The composite function $(g\circ f)$ relates the VO sequence $O = o_1\, o_2 \ldots o_T$ to its shape vector defined in a Euclidean space. This mapping allows the traditional HMM to be ingrained in a Euclidean space. Therefore, metrics and topology can be exploited within the HMM's framework. This composite mapping is depicted as follows:



4. Standard hidden Markov models

To better understand the contribution of the THMMs, we found it useful to first provide a summarized description of the traditional HMMs. For further information on HMMs, refer to [2].

Definition 4.1. A hidden Markov model is a doubly embedded stochastic process with an underlying stochastic process that is not observable (it is hidden), but can only be observed through another set of stochastic processes that produce the sequence of observations.

4.1. Elements of an HMM

We now introduce the elements of an HMM, and explain how the model generates observation sequences. An HMM is characterized by the following parameters:

- **N**, the number of hidden states q_i in the model.
- **R**, the number of distinct observation O_i per hidden state, i.e., the size of the discrete alphabet.
- The initial state distribution $\pi = \{\pi_i\}$, where $\pi_i = P(q_0 = e_i)$, $1 \le i \le N$, and $\sum_i \pi_i = 1$.
- The state transition probability distribution $\mathcal{A} = \{a_{ij}\}$, where $a_{ij} = P(q_{t+1} = e_j | q_t = e_i), 1 \le i,\ j \le N$ and $\sum_j a_{ij} = 1$.
- The emission probability distribution, $\mathcal{B} = \{b_j(k)\}$, where $b_j(k) = P(o_k \text{ at time t} | q_t = e_j)$, $1 \le k \le R$ and $1 \le j \le N$, and $\sum_k b_j(k) = 1$.

An HMM is usually represented as $\lambda = [\pi, \mathcal{A}, \mathcal{B}]$.

4.2. The three basic problems of an HMM

There are three basic problems that are assigned to an HMM, they are:

(i) *Evaluation*: Given the observation sequence $O = o_1, o_2, \ldots, o_T$ and a model $\lambda = [\pi, \mathcal{A}, \mathcal{B}]$, determine the probability that this observation sequence was generated by the model λ.
(ii) *Decoding*: Suppose we have an HMM λ and a VO sequence O, determine the most likely sequence of hidden states $q_1, q_2, \ldots, q_T$ that generated O.
(iii) *Learning*: Suppose we are given a coarse structure of a model (the number of hidden states and the number of observations symbols) but not the probabilities a_{ij} nor b_{jk}. Given a limited set of training observation sequences, determine these parameters. In other words, the goal is to search for the model λ that is most likely to have produced these observation sequences.

We first focus on the evaluation problem: Let $O=(o_1\ o_2 \ldots o_T)$ be the VO sequence of length T and $q=(q_1\ q_2\ \ldots q_T)$ the state sequence with q_0 as an initial state. The evaluation problem is expressed as follows: Given a model λ, and the observation sequence O, evaluate the match between λ and the observation sequence O by computing $P(O|\lambda)$:

$$P(O|\lambda) = \sum_q P(O, q|\lambda) = \sum_q P(O|q, \lambda) \times P(q|\lambda). \tag{1}$$

Using the state conditional independence assumption of the visible observation sequence O, that is: $P(o_1, o_2, \ldots, o_T|q) = \prod_{t=1}^{T} P(o_t|q_t)$, and assuming a first-order Markov chain, we derive the following:

$$P(O|\lambda) \approx \sum_q \prod_{t=1}^{T} P(o_t|q_t) \times P(q_t|q_{t-1}). \tag{2}$$

The evaluation problem is based on the state conditional independence assumption of the VO sequence symbols. *However, there are several scenarios where a long range dependency between visible observations is needed.* Besides, this dependency would be much more informative if it were not only temporal but topological as well. In other words, we would be more advanced if we knew what shape these related observations are forming. *Unfortunately, the notion of UNIF calls for topological features such as shape is absent in the traditional HMM's formalism*. It has been proven that standard HMMs perform well in recognizing amino acids and consequent construction of proteins from the first level structure of DNA sequences [22], however, they are inadequate for predicting a tertiary structure of a protein. The reason for this inadequacy comes from the fact that *the same order of amino acid sequences might have different protein folding modes in natural circumstances [10]. In other words, it is only the shape information that enables the discrimination between these different folding modes*.

5. Structural hidden Markov models

In this section, we present a brief mathematical description of the structural hidden Markov models introduced in [15]. This formalism goes beyond the traditional hidden Markov model since it emphasizes the structure of the visible observation sequences and their temporal positions.

In traditional HMMs, the visible observations are assumed to be *state conditionally independent*. However, there are several scenarios where the conditional independence assumption does not hold. For example, while standard HMMs perform well in recognizing amino acids and consequent construction of proteins from the first level structure of DNA sequences [22], 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 [10]. Therefore, there is a need to balance the loss incurred by this state conditional independence assumption.

The 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. 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 $O_i = (o_{i_1} o_{i_2} \ldots o_{i_{r_i}})$ $(1 \le i \le m)$. The symbols of a string are related in the sense that they define a local structure S_j of the whole complex pattern. This structural information is captured through a relation of equivalence between the symbols o_i. Each structure S_j is a class of equivalence that gather all similar group of symbols o_i. For example, a sequence of phonemes O_i produces a word S_j with a certain probability $P(S_j|O_i)$ 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. 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

$$P(O|\lambda) = \sum_S P(O, S|\lambda). \tag{3}$$

We first need to evaluate $P(O, S|\lambda)$, we can write

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

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

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

conditional independence of the O_i's with respect to the structure sequence is assumed. A structure S_i depends only on the observation sequence O_i and the structure probability distribution is a Markov chain of order 1. Therefore, Eq. (6) can be written as

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

In order to show how the symbols o_i are inter-related to form a particular structure, Bayes' rule has been applied in Eq. (7):

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

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

Definition 5.1. A structural hidden Markov model is a quintuple $\lambda = (\pi, \mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$, where:

- π is the initial state probability vector,
- $\mathcal{A}$ is the state transition probability matrix,
- $\mathcal{B}$ is the state conditional probability matrix of the visible observations,

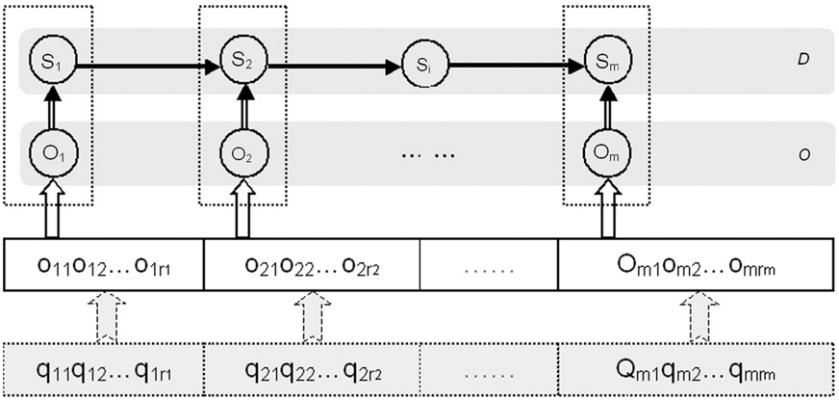


Fig. 4. A graphical representation of a structural hidden Markov model.

- $\mathcal{C}$ is the posterior probability matrix of a structure given a sequence of observations,
- $\mathcal{D}$ is the structure transition probability matrix.

A structural hidden Markov model 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.
- π, the initial state distribution, $\pi=\{\pi_i\}$, where $\pi_i=P(q_1=i \text{ at } t=0)$ and $1\le i\le N$, $\sum_i\pi_i=1$.
- $\mathcal{A}$, the state transition probability distribution matrix, $A=\{a_{ij}(t)\}$, where $a_{ij}(t)=P(q_{t+1}=j|q_t=i)$, $\sum_{j,t}a_{ij}(t)=1\ \forall i$, where $1\le i,\ j\le N$ $t=1,\ldots,T$.
- $\mathcal{B}$, the state conditional probability matrix of the observations, $B=\{b_j^t(k)=P(o_k|q_j \text{ at time } t)$, $\sum_{k,t}b_j^t(k)=1$, $1\le k\le M$ and $1\le j\le N$.
- **F**, the number of distinct structures.
- $\mathcal{C}$ is the posterior probability matrix of a structure given its corresponding observation sequence, $\mathcal{S}=P(S_j|O_i)=s_i(j)$. For each particular input string O_i, the natural constraint $\sum_j s_i(j)=1$ is obeyed. The different structures are obtained from a data set using an equivalence relation.
- $\mathcal{D}$, the structure transition probability matrix. $\mathcal{D}=\{d_{ij}\}$, where $d_{ij}=P(S_{t+1}=j|S_t=i)$, $\sum_j d_{ij}=1$, $1\le i,j\le F$.

Fig. 4 depicts a representation of a structural hidden Markov model.

6. Topological hidden Markov models

Because the topological concept of *shape and its representation* is absent in the structural HMMs, therefore the thrust in the THMMs formalism is to classify VO sequences made of symbols that when grouped together and deformed in a certain manner may exhibit shapes. It is noteworthy that not all sequences of symbols encountered in nature possess this pattern of disclosing shapes. Furthermore, the shapes assigned to UNIF's are captured by their external contours. A contour can be viewed as a discrete signal that consists of low-frequency and high-frequency contents. The low-frequency content is the most important part of the signal, since it provides the signal with its identity: This part is known as *the pure signal*. However, the high-frequency signal conveys flavor or nuance: This part is usually *associated with noise*. For example, the Fourier transform $c(k)$ of a function $f(t)$ is computed for only a limited number of k values which cover lower and higher frequency terms. Similarly, the wavelet analysis uses two technical terms which are: *approximations A* (low resolution view of the image: low-frequency components) and *details D* (details of the image at different scales and orientations: high-frequency components). Approximations refer to the high-scale factor, these components of the signal are matched with the stretched wavelets. However, details represents low-scale factor, these components of the signal are matched with the compressed wavelets. *The thrust behind the concept of THMMs is to express the probability distribution assigned to the shape of the pure signal as a function of the Gaussian distribution assigned to the shape of the signal noise*. Therefore, the tasks in the THMMs consist of: (i) representing the shape formed by the observation sequence through any state of the art shape analysis technique and (ii) modeling the noise assigned to the shape via a Gaussian distribution.

6.1. UNIF shape representation

Let $O=o_1,\ o_2,\ldots,o_T$ be a VO sequence of length T made of symbols o_i. Let $X(t)=\{x(t)\}_{t=1}^{t=m}$ be the closed contour representation of length m that captures the shape of its UNIF. Each n-dimension point of this contour is designated by $x(t)=[x_1(t),\ldots,x_n(t)]^T$. For the sake of simplicity, we focus in this paper on three-dimension objects ($n=3$). Object Shape representation can be performed in the spatial domain or in the transform domain. Our goal is to extract the noisy part of a signal during the shape analysis of the object. If we adopt the 3D Fourier descriptor (FD) method to efficiently discriminate the external contour of an object, therefore the contour $X(t)$ (regarded as a 2π periodic function) is approximated using an infinite sum of *sine* and *cosine* functions. In a 3D space, if $\omega=2\pi\times t/T$ (T is the total contour length), then using the Lin and Hwangs' direct scheme FD representation [23], we can estimate (using a hat notation) each point $x(t)$ of the external contour $X(t)$ as

$$\hat{x}(t)=\begin{bmatrix}\hat{x}_1(t)\\ \hat{x}_2(t)\\ \hat{x}_3(t)\end{bmatrix}=\begin{bmatrix}a_0\\ c_0\\ e_0\end{bmatrix}+\sum_{k=1}^{k=N}\begin{bmatrix}a_k & b_k\\ c_k & d_k\\ e_k & f_k\end{bmatrix}\times\begin{bmatrix}\cos(k\omega)\\ \sin(k\omega)\end{bmatrix}+\sum_{k\ge N+1}\begin{bmatrix}a_k & b_k\\ c_k & d_k\\ e_k & f_k\end{bmatrix}\begin{bmatrix}\cos(k\omega)\\ \sin(k\omega)\end{bmatrix},$$

where a_k, b_k, c_k, d_k, e_k, and f_k are the Fourier coefficients corresponding to the k th harmonics. Practically we are often satisfied with a finite number N of these functions. The inherent presence of noise in the raw data o_t warrants the use of FDs. In the

scenario where N is large, a random noise is added during the external contour reconstruction. We assume that the noisy part in Eq. (9) starts from $k=N+1$ and any other term is part of the pure signal.

Similarly, if we adopt the 3D wavelets transform (constructed as separable products of 1D wavelets by successively applying a 1D analyzing wavelet in three spatial directions x_1, x_2, and x_3, therefore we can still approximate a 3D signal using the approximation terms A and the details term D. These two component parts of the signal can be separately extracted using a filter bank. The original signal is the fusion of the A and D terms; they both contribute to the reconstruction process by revealing complementary characteristics of the signal. Mathematically stated: The inverse transform of a function $f(x)\in L^2$ with respect to some analyzing wavelet Ψ_{jk} (j: scale, k: position) is defined as $f(x)=\sum_j\sum_k c_{j,k}\Psi_{j,k}(x)$, where $c_{j,k}=\int_{-\infty}^{+\infty} f(x)\Psi_{j,k}(x)\,dx$ are coefficients known as discrete wavelet transform (DWT) of f(x) [24]. A discrete parametrized closed curve that represents the shape of a 3D object of interest is the vector: $\hat{x}(t)=[\hat{x}_1(t),\hat{x}_2(t),\hat{x}_3(t)]^T$. If the wavelet transform is applied independently to each of the $\hat{x}_1(t)$, $\hat{x}_2(t)$ and $\hat{x}_3(t)$ functions, we can describe the 3D curve in terms of a decomposition of $\hat{x}(t)$. However, the noise in the image is contained mostly in the details term D of each coordinate. This random noise which is part of the whole image signal in the transform domain is modeled probabilistically via a Gaussian distribution function. *This very noise is the source of the fusion betweendiscrete structure and topology.* In conclusion, whatever image processing technique we intend to use, we can coarsely approximate the original signal $x(t)$ by decomposing it into a sum of a pure signal and a noisy signal. We can write

$$x(t)\approx\hat{x}(t)=\zeta(t)\oplus N(t)\quad(t=1,\ldots,m), \tag{9}$$

where $\zeta(t)$ is the 3D pure signal vector (based on Fourier descriptors, or wavelet transform coefficients) assigned to low frequency components and $N(t)$ is a 3D Gaussian noise vector assigned to high frequency components, with mean vector μ_t and covariance matrix Σ_t.

6.2. First level THMMs: mathematical formulation

We introduce a mathematical expression of *the first level topological hidden Markov models* and the different problems assigned to it. We also provide a definition of this model and the parameters involved. We assume that the external contour of the shape formed by the UNIF assigned to the VO sequence is computed using any state of the art shape analysis technique. We also assume that the same symbol of a VO sequence *can be located at different n-dimension coordinates in the UNIF shape*. For example, the same amino acid can be located at different position coordinates in a 3D protein fold. In this scenario, the evaluation problem is stated as: Given a model λ, the VO sequence O, an approximation of its UNIF external contour sequence $\hat{X}(t)=\{\hat{x}(t)\}_{t=1}^{t=m}$; evaluate the match between λ and this VO sequence O by computing $P[O|\lambda]$. If q stands for the hidden state sequence assigned to O, then

$$P(O|\lambda)=\sum_q P[O,\hat{X}(t),q|\lambda]. \tag{10}$$

Using the conditional probability rule, we have

$$P[O,\hat{X}(t),q|\lambda]=P[\hat{X}(t)|O,q,\lambda]\times P(O|q,\lambda)\times P(q|\lambda). \tag{11}$$

The product $P(O|q,\lambda)\times P(q|\lambda)$ expresses a traditional hidden Markov model. Finally, the term that remains to be computed is $P[\hat{X}(t)|O,q,\lambda]$. By replacing $\hat{x}(t)$ with its $(\zeta(t)\oplus N(t))$ decomposition, we obtain

$$P[\hat{X}(t)=\{\hat{x}(t)\}_{t=1}^{t=m}|O,q]=P[\{N(t)=\hat{x}(t)\ominus\zeta(t)\}_{t=1}^{t=m}|O,q]. \tag{12}$$

However, it is reasonable to assume that the random noise embedded in the contour 3D points is independent of the hidden state sequence q, but depends only on the visible symbols representing the raw data O, therefore

$$P[\{N(t)=\hat{x}(t)\ominus\zeta(t)\}_{t=1}^{t=m}|O,q]=P[\{N(t)=\hat{x}(t)\ominus\zeta(t)\}_{t=1}^{t=m}|O], \tag{13}$$

where $N(t)$ is a multivariate Gaussian distribution. Finally, the first level THMMs evaluation problem can be written as

$$P(O|\lambda)\approx\sum_q P[\{N(t)=\hat{x}(t)\ominus\zeta(t)\}_{t=1}^{t=m}|O]\times P(O|q,\lambda)\times P(q|\lambda). \tag{14}$$

Since, each noise point on a contour depends only on its observation symbol data o_k, therefore Eq. (14) can be extended to

$$P(O|\lambda)\approx\sum_q\left[\prod_{t=1}^{m} P[N(t)=\hat{x}(t)\ominus\zeta(t)|o_{\hat{x}(t)}]\times P(q_0)\times P(o_t|q_t)\times P(q_t|q_{t-1})\right], \tag{15}$$

where $o_{\hat{x}(t)}$ are the k symbols of the VO sequence such that $f(o)\supset\hat{x}(t)$:

$$P[N(t)=\hat{x}(t)\ominus\zeta(t)|o_{\hat{x}(t)}]=\frac{1}{(2\pi)^{3/2}|\Sigma_t|^{1/2}}\times\exp\left[-\frac{1}{2}[N(t)-\mu(t)]^T\Sigma_t^{-1}[N(t)-\mu(t)]\right].$$

Both the noise $N(t)=\hat{x}(t)\ominus\zeta(t)$, and the mean $\mu(t)$ are 3D vectors. The mean vector and the covariance matrix are respectively ML-estimated as: $\hat{\mu}=(1/k)\sum_{t=1}^{t=k}N(t)$, and $\hat{\Sigma}=(1/(k-1))\sum_{t=1}^{t=k}[N(t)-\hat{\mu}][N(t)-\hat{\mu}]^T$, respectively. We now define the first level THMMs:

Definition 6.1. A first level THMM is a quadruple $\lambda=[\pi,\mathcal{A},\mathcal{B},\mathcal{T}]$, where:

- $\pi=\{\pi_i\}=P(q_0=e_i)$ is the initial hidden state probability vector,
- $\mathcal{A}=\{a_{ij}\}=P(q_{t+1}=j|q_t=i)$ is the hidden state transition probability matrix,
- $\mathcal{B}=\{b_j(k)\}=P(o_k\ at\ time\ t|q_t=e_j)$ is the emission probability matrix,
- $\mathcal{T}=P[N(t)=\hat{x}(t)\ominus\zeta(t)|o_{\hat{x}(t)}]$ is the probability distribution function assigned to the noise produced by the k contour points $\hat{x}(t)$ that belong to $f(o)$.

Conceptually, we view the generation mode of the THMM's formalism as follows: Each symbol O_i of a VO sequence O is emitted from a hidden state $q_j\in\{1,2,\ldots,n\}$ at each time unit. A sequence of symbols is therefore created and distorted (through the mapping f) to form a particular UNIF whose shape is produced through the THMMs. The THMM's formalism tells what is the shape formed by the VO sequence. The mapping g is simply a shape representation projected onto a Euclidean space. Fig. 5 depicts the state transition graph of a first level THMMs.

6.2.1. The problems assigned to a first level THMM

Four problems are assigned to a first level THMM, they are:

- *Probability evaluation*: Given a model λ and a VO sequence O with its corresponding UNIF external contour points sequence $X(t)=\{x(t)\}_{t=1}^{t=m}$, the goal is to evaluate how well does λ match O.
- *Statistical decoding*: In this problem, we attempt to find the "best" hidden state sequence $q^*=\langle q_1^*,q_2^*,\ldots,q_T^*\rangle$ such that:

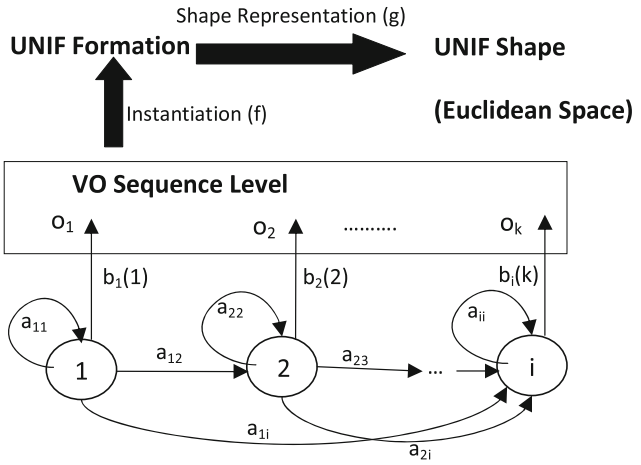


Fig. 5. The state transition graph of a first level topological hidden Markov model: The digits 1, 2, 3,…,*i* represent hidden states. The nodes o_i are emitted symbols that form a UNIF whose shape representation is projected onto a Euclidean space.

$q^* = \text{argmax}_q[P(O,q|\lambda)]$ that best "explains" the visible observation sequence O. This problem is similar to problem 2 of a traditional HMM. Its solution is implemented using Viterbi algorithm that it is based on the evaluation of the maximum probability of all sequences ending in state i at time t, $\delta_t(i)$ and the extraction of a partial best path as conducted in dynamic programming. The Viterbi algorithm is fully described in [2].

- *Topological decoding*: In this problem, the task consists of determining the "correct" shape of the UNIF assigned to the VO sequence O via the noise on its external contour.
- *Learning*: In this problem, we try to determine the model parameters $\lambda = [\pi, \mathcal{A}, \mathcal{B}, \mathcal{T}]$ that maximize $P(O|\lambda)$.

6.3. Second level topological hidden Markov models

We introduce in this section a mathematical description of *the second level topological hidden Markov models* that extends the first level THMMs.

Psychophysical studies [25] show that we can recognize objects using fragments of outline contour alone. In this context, a VO sequence $O=o_1,o_2,\ldots, o_T$ is viewed as made of constituents $O_1,O_2,\ldots,O_s$. Each O_i is a string of symbols $o_i \in \Sigma$ interrelated in some way. In other words, each VO sequence O is not only one sequence in which all symbols are conditionally independent, but also a sequence that is divided into a series of s strings $O_i = o_{i_1}, o_{i_2} \ldots o_{i_{l_i}}$ $(1 \le i \le s)$. The task within the second level THMMs is threefold: (i) segment an entire VO sequence into s "meaningful" pieces, (ii) determine the shape of each UNIF assigned to a segment O_i by embedding it in a Euclidean space, and (iii) compute the joint probability of the entire VO sequence O with its UNIF sequence.

6.3.1. Optimal segmentation of the entire VO sequence

The goal is to determine a methodology that enables segmenting a T-element sequence into s "meaningful" segments (or strings) using a predefined criterion. This problem is known as *the* (s,s) *segmentation problem*. Let $Seg_s(O)$ be the set of all segmentations of O into s segments. Therefore, the (s,s) segmentation problem can be stated as follows: Assume we are given a sequence $O = o_1\, o_2 \ldots o_T$, where $o_i \in \Sigma$, how can we determine the best segmentation $\Delta^* \in Seg_s(O)$ amongst all possible segmentations of O into s segments? A segmentation $\Delta \in Seg_s(O)$ is defined by $s+1$ segment boundaries $1 = b_1 < b_2 < \cdots < b_s < b_{s+1} = T+1$, generating segments $O_1, O_2,\ldots,O_s$ where: $O_i = o_{b_i}, \ldots, o_{b_{i+1}-1}$. The best segmentation Δ^* is the one that creates *homogeneous* segments O_i with respect to some error measure. Depending on the nature of the data, different error measures can be investigated. We propose the following error measure: $E(O_i) = \sum_{o_i \in O_i} d^2(o_i, \overline{o}_i)$, where $\overline{o}_i$ is the *most representative* symbol of the segment O_i and d is a distance. If the data are real valued and defined in a Euclidean space, therefore the most representative symbol is the *mean* and the error measure in this case is simply the variance. Since there are several possible segmentations $\Delta \in Seg_s(O)$, thus the global error measure is defined as

$$E(O,\Delta) = \sum_{O_i \in \Delta} \sum_{o_i \in O_i} d^2(o_i, \overline{o}_i). \tag{16}$$

Finally, the optimal segmentation task consists of finding the segmentation $\Delta^* \in Seg_s(O)$ that minimizes $E(O,\Delta)$. Dynamic programming approaches is used to solve this problem in a tractable and efficient manner [26]. However, the optimal solution may not be unique. There could be more than one segmentation Δ that minimize the error measure $E(O,\Delta)$. Our strategy consists of selecting the one that has the smallest number of segments s.

6.3.2. UNIF formation through unsupervised clustering

So far, we have defined a UNIF as a shape that can unfold after a stretch (or a deformation) of a VO subsequence. However, we have not shown how this process can be achieved. The objective of this section is to unravel the formation of the UNIF entity. *The UNIF's are built through an unsupervised clustering algorithm applied to a set of vectors representing shapes*. Each cluster gathers the shapes (formed by the constituents o_i's) that are similar in some sense. The organization of the symbols o_i contributes to the production of the UNIF U_j. For example, a cloud of points O_i representing a VO sequence forms a circle or an ellipse U_j with a certain probability $P(U_j|O_i)$. This circular (or elliptical) shape is viewed as a cluster that gathers all round shapes with respect to some metric distance and a fixed threshold. Therefore, we define the notion of UNIF's as follows:

Definition 6.2. By partitioning the set $\mathcal{S}$ into a set of clusters. Each cluster U is a UNIF that describes piecewise the global shape formed by the entire VO sequence O.

The higher the complexity of the shape formed by the VO sequence, the higher the number of UNIF's needed to describe it. Fig. 6 depicts examples of two objects that are decomposed into several UNIF's (or structures).

6.3.3. Mathematical formulation of the second level THMMs

We present the mathematical expression of the second level THMMs. We also give a definition of this model and the

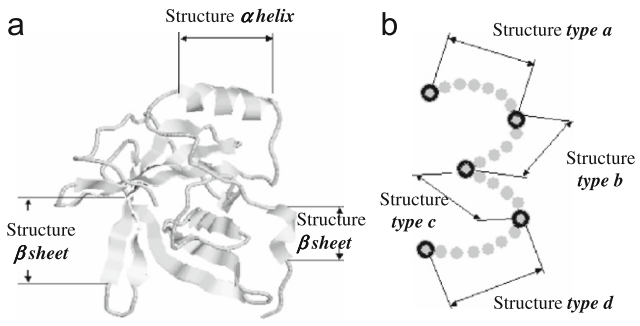


Fig. 6. The organization of constituents and their shapes in two different objects (a) 3D protein fold with α helix and β sheet as UNIF's and (b) a handwritten character representing the digit "three" with structure of type a, b, c and d as UNIF's.

parameters involved. Let $O = O_1, O_2, \ldots, O_s = o_{1_1} o_{1_2} \ldots o_{1_{r_1}}$, $o_{2_1} o_{2_2} \ldots o_{2_{r_2}}, \ldots, o_{s_1}$, $o_{s_2}, \ldots, o_{s_{r_s}}$, (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., such that $\sum_{i=1}^{i=s} r_i = T$). Let $U = U_1, U_2, \ldots, U_s$ be the UNIF sequence assigned to the subsequences O_i's, and $X(t) = X_1(t), X_2(t), \ldots, X_s(t)$ be the sequence of all external contours assigned to the UNIF sequence. The length of $X(t)$ is equal to $m = m_1 + m_2 + \cdots + m_s$, where m_j is the length of the subcontour $X_j(t)$. Each O_i is mapped to its contour $X_j(t)$ using the mapping I defined in Section 3. Let $\hat{X}_i(t)$ be the series of 3D points of each $X_i(t)$, and $\hat{X}(t) = \{\hat{X}_i(t)\}_{i=1}^{i=s}$ be the series of the 3D points of the entire contour $X(t)$. The probability of the observation sequence O with its external contour $X(t)$ (defined piecewise) given a model λ can be written as

$$P(O|\lambda) = \sum_U P[O, \hat{X}(t), U|\lambda]. \tag{17}$$

Since the model λ is implicitly present during the evaluation of this joint probability, therefore it is omitted. We first need to evaluate $P[O, \hat{X}(t), U]$. It is reasonable to assume that the series $\hat{X}(t)$ depends only on the observation sequence O. Thus, using Bayes' formula first and then conditional independence of the $\{\hat{X}_i(t)\}_{i=1}^{i=s}$, we can write

$$P[O, \hat{X}(t), U] \approx \prod_{i=1}^{i=s} P[\hat{X}_i(t)|O_i] \times P(O, U). \tag{18}$$

We evaluate each term separately. We first start by computing the first term of Eq. (18), which is: $P[\hat{X}_i(t)|O_i]$. Since the vector $N_i(t) = \hat{x}_i(t) \ominus \zeta_i(t)$, then

$$\begin{aligned} P[\hat{X}_i(t)|O_i] &= \prod_{t=1}^{t=r_i} P[N_i(t) = \hat{x}_i(t) \ominus \zeta_i(t)|O_i] \\ &= \prod_{t=1}^{t=r_i} P[N_i(t) = \hat{x}_i(t) \ominus \zeta_i(t)|o_{\hat{x}_i(t)}] \\ &= \prod_{t=1}^{t=r_i} \frac{1}{(2\pi)^{3/2}|\Sigma_{i,t}|^{1/2}} \\ &\quad \times \exp\left[-\frac{1}{2}[N_i(t) - \mu_i(t)]^T \Sigma_{i,t}^{-1} [N_i(t) - \mu_i(t)]\right] \equiv \Phi_i. \end{aligned}$$

The second term of Eq. (18) is computed as follows. For the sake of simplicity, we assume that O_i depends only on U_i, and the UNIF probability distribution is a Markov chain of order 1 (illustrated by Fig. 7). Finally, we can recursively approximate the second term of Eq. (18):

$$P(O_1, \ldots, O_s, U_1, \ldots, U_s) \approx \prod_{i=1}^{i=s} P(O_i|U_i) \times P(U_i|U_{i-1}), \tag{19}$$

where $P(U_1|U_0) \equiv P(U_1)$ since the form U_0 does not exist. If

$$\frac{P(U_i|O_i) \times P(O_i) \times P(U_i|U_{i-1})}{P(U_i)} \equiv \Psi_i, \tag{20}$$

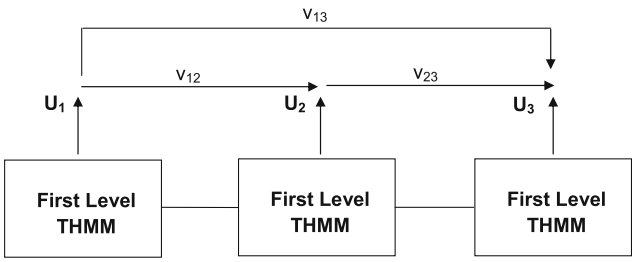


Fig. 7. A state transition graph of a second level THMM. The global shape of the entire VO sequence is captured piecewise through the UNIF's U_i extracted from each first level THMM.

therefore, by regrouping the expressions of all the terms involved in Eq. (18), we obtain the final expression of the second level THMMs:

$$P(O|\lambda) \approx \sum_{U_1 U_2 \ldots U_s} \prod_{i=1}^{i=s} [\Phi_i] \times [\Psi_i]. \tag{21}$$

The uncertainty about the shapes of the external contour formed by the observation sequence $O = (O_1, O_2, \ldots, O_s)$ is captured by the Gaussian noise probability distribution. The UNIF U_i assigned to O_i is introduced via the term $P(U_i|O_i)$. Besides, the term $P(O_i)$ of Eq. (21) is viewed as a first level THMMs. Therefore, we can state:

Definition 6.3. A second level THMM is a sextuple $\lambda = [\pi, \mathcal{A}, \mathcal{B}, \mathcal{U}, \mathcal{D}, \mathcal{T}]$, where:

- π, the initial hidden state distribution within a constituent O_i, where $\pi_i = P(q_0 = i)$ and $1 \le i \le N$, $\sum_i \pi_i = 1$.
- $\mathcal{A}$, the hidden state transition probability distribution matrix within a constituent O_i, $\mathcal{A} = \{a_{ij}\}$, where $a_{ij} = P(q_{t+1} = j|q_t = i)$ and $1 \le i, j \le N$, $\sum_j a_{ij} = 1$.
- $\mathcal{B}$, the emission probability matrix within a constituent O_i, $\mathcal{B} = \{b_j(k)\}$, in which $b_j(k) = P(o_k|q_j)$, $1 \le k \le R$ and $1 \le j \le N$, $\sum_k b_j(k) = 1$.
- $\mathcal{U}$ is the posterior probability matrix of a UNIF U_i given its constituent O_i, $\mathcal{U} = P(U_j|O_i) = u_i(j)$, subject to: $\sum_j u_i(j) = 1$.
- $\mathcal{D}$, the UNIF transition probability matrix, where: $\mathcal{D} = \{d_{ij}\} = P(U_{t+1} = j|U_t = i)$, $\sum_j d_{ij} = 1$, $1 \le i, j \le F$.
- $\mathcal{T}$, is the noise probability distribution contained in the representation of the shape $X_i(t)$ formed by the subsequence O_i, it is written as: $P[N_i(t) = \hat{x}_i(t) \ominus \zeta_i(t)|O_i] = 1/(2\pi)^{3/2}|\Sigma_{i,t}|^{1/2} \times \exp[-(1/2)[N_i(t) - \mu_i(t)]^T \Sigma_{i,t}^{-1} [N_i(t) - \mu_i(t)]]$.
- **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.
- **R**, the number of points in an external contour $X_i(t)$.
- **F**, the number of distinct UNIF's.

Fig. 7 depicts the state transition graph of a second level THMMs.

6.3.4. Problems assigned to a second level THMM

There are five problems that arise in the context of a second level THMM:

- *Probability evaluation*: Given a model $\lambda = [\pi, \mathcal{A}, \mathcal{B}, \mathcal{U}, \mathcal{V}, \mathcal{T}]$ and a sequence of observations $O = (O_1, \ldots, O_s)$, we evaluate how well does the model λ match O. This problem has been discussed in Section 6.3.3. It can be implemented using the forward procedure as in the traditional HMMs.
- *Statistical decoding*: The statistical decoding problem consists of determining the optimal hidden state sequence $q^* = \text{argmax}_q[P(O_i, q|\lambda)]$ that best "explains" a constituent O_i. This process is repeated for each constituent of the entire sequence O. This task is implemented using Viterbi algorithm.
- *Structural decoding*: The structural decoding problem consists of determining the optimal UNIF sequence $U^* = \langle U_1^*, U_2^*, \ldots, U_s^* \rangle$ such that

$$U^* = \underset{U}{\text{argmax}} P(O, U|\lambda). \tag{22}$$

We define

$$\delta_t(i) = \max_{\mathcal{U}}[P(O_1, \ldots, O_t, U_1, \ldots, U_t = i|\lambda)], \tag{23}$$

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 form i. Then, using induction we obtain

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) v_{ij}\right] u_{t+1}(j) \frac{P(O_{t+1})}{P(U_j)}. \quad (24)$$

Similarly, this latter expression can be computed using *Viterbi* algorithm. However, we estimate δ in each step *through the UNIF transition probability matrix*. For example, a sequence such as: ⟨round, curved, straight, zigzag ... convex⟩ can be derived to describe the global shape formed by the VO sequence.

- *Topological decoding*: In this problem, the task consists of determining the "correct" shapes of the UNIF's assigned to the VO subsequences O_i via the noise embedded in their external contours $X_i(t)$. In this step, the mean vector $\mu_i(t)$ and the covariance matrix $\Sigma_{i,t}$ assigned to each O_i are ML estimated as in the first level THMMs. The UNIF sequence ⟨round, curved, straight, zigzag ... convex⟩ is decoded in terms of its contour vector sequence.
- *Learning*: The goal in this section is to compute the model parameters $\lambda = [\pi, \mathcal{A}, \mathcal{B}, \mathcal{U}, \mathcal{V}, \mathcal{T}]$ that maximize the likelihood $P(O|\lambda)$. In order to estimate the posterior probability $P(U_j|O_i)$, we used the *k-nearest-neighbors* method [27]. For any given UNIF U_j, we estimated $u_i(j) = P(U_j|O_i)$ as $u_i(j) \approx k_j/k$, where k_j is the number of contour representation vectors that belong to the UNIF U_j (cluster) amongst all possible k-nearest neighbors of the external contour representation vector assigned to $X_i(t)$. This estimation requires the definition of "neighbor". If one adopts the Fourier descriptor technique to represent the shape external contour, therefore, the similarity distance between two 3D shapes is computed using the L_2 norm between their Fourier descriptor vectors. In the case of an unseen sequence O_u that might be encountered during a testing phase, the probability $P(U_j|O_u)$ will be estimated by $P(U_j|O_i)$ $\forall j$. The VO sequence O_i is such that the contour $X_i(t)$ of its UNIF is the "closest" to the contour $X_u(t)$ assigned to O_u in the training set. The *k-nearest-neighbors* posterior probability estimation technique obeys the exhaustivity and exclusivity constraint: $\sum_j u_i(j) = 1$. This estimation enables to built the entire matrix $\mathcal{U}$. Since the contour of the entire VO sequence is represented by a sequence of UNIF's, therefore we can use the Baum–Welch optimization technique to estimate the matrix $\mathcal{V}$. The other parameters, $\pi = \{\pi_i\}$, $\mathcal{A} = \{a_{ij}\}$, $\mathcal{B} = \{b_j(k)\}$, were estimated as in HMMs [2].

It is worth to underscore that the UNIF sequence describes the structural pattern topologically, piece by piece. Because of the shape consideration, it becomes possible to differentiate between low energy state levels of two protein secondary structures such as "CompressedHelix" and "ElongatedHelix". This difference is fundamental in proteomics since the folding mode is related to the energy state level. The following algorithm describes the different steps involved in a second level THMM.

- **Training**:

(i) Collect a training set containing VO sequences of arbitrary sizes
(ii) Break up each VO sequence into segments as explained in Section 6.3.1
(iii) Determine the UNIF sequence assigned to these segments through Instantiation
(iv) Compute the shape of these UNIF's using their external contour vectors
(v) Cluster these vectors into k clusters labeled U_i (i=1,...,k).
(vi) Extract the noise component in each shape representation vector using a filter bank.
(vii) Compute the optimal model $\lambda^* = [\pi^*, \mathcal{A}^*, \mathcal{B}^*, \mathcal{U}^*, \mathcal{V}^*, \mathcal{T}^*]$ for each class ω_i (i=1,...,c).

- **Testing**:

(i) Break up each VO sequence into segments and determine the UNIF's assigned to these segments and their contours
For each sequence O of the test set **Do**
Begin
Compute $P(O|\lambda_i)$ (i=1,...,c),
Select the best model and assign its class ω_i to the test sequence O
End
(ii) Compute the accuracy of the second level THMMs using a re-estimation method.

7. Selected applications

In order to demonstrate the overall significance of the THMM's paradigm, we have selected two different applications: (i) handwritten numeral recognition, and (ii) protein fold recognition. We have compared the THMM's approach with some state of the art classifiers; the experimental results obtained in these two applications are reported below.

7.1. Application 1: handwritten numeral recognition

In this section, we show how the classification task of handwritten numerals is conducted using a THMM's approach. 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 digits in the MNIST repository have been size-normalized to fit in a $20 * 20$ pixel box, and centered on a $28 * 28$ pixel image. The training set contains 60,000 digits and the testing set 10,000 digits.

7.1.1. Data collection and UNIF formation

It is well known that the performance of a handwritten numeral recognition system depends largely on the feature extraction phase. In this application, *a sequence of 2D coordinate points of the external closed contour of the entire numeral corresponds to the VO sequence*. We used the standard 8-directions chain code to represent the closed contours of the digits [28]. The extracted 8-directions features are sequences of integers from 0 to 7. The chain code representation entails a deformation of the contour 2D points sequence which corresponds to the mapping f (from the VO sequence to the UNIF sequence) introduced in Section 3. It is well known that 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 information. To solve this problem, we extracted a sequence of *local structures* that composes the entire digit contour. Each local structure is a sequence of integers from 0 to 7 that represents a *UNIF*. We used an unsupervised clustering algorithm to extract the local structures automatically from the chain code sequence of the numeral. *This phase corresponds to the structural decoding of the handwritten numeral.* The first step of the structure extraction is to determine all numeral *strokes*. A stroke is separated from another one 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. 8.

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 the "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 having extracted all strokes, we clustered them using the probabilistic principal component analysis (PPCA) technique [29] and assigned each cluster (or UNIF) a label. Finally, each cluster is considered as a local structure and each stroke has different probabilities belonging to different clusters. We calculated a "weighted mean" vector ξ_i from the strokes for each local structure U_i using the following equation:

$$\xi_i = \frac{\sum_{V_k \in U_i} P(U_i|V_k) \times V_k}{\sum_{V_k \in U_i} P(U_i|V_k)}, \tag{25}$$

where V_k is the feature vector of each stroke contained in the local structure U_i. The weighted mean vector ξ_i represents a *signature* of each local structure U_i in terms of its shape: It corresponds to the mapping g introduced in Section 3. This signature enables to perform a comparison between several local structures. As outlined in Section 6.3.4, *the topological decoding reveals the shape of each local structure of the handwritten numeral*. This information is embedded in the feature vector ξ_i which is computed through a Gaussian probability distribution assigned to the noise in the chain code representation of each stroke contained in this local structure. We have used the ML estimation to compute the mean vector $\mu_i(t)$ and the covariance matrix $\Sigma_{i,t}$ of the Gaussian probability distribution. A numeral is thus expressed as a sequence of local structures with their shapes. Table 1 shows the number of local structures found for all 10 digits.

Fig. 8. The strokes are separated when a significant change of the chain code directions occurs.

Because THMMs require 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. Fig. 9 depicts an example of the starting point.

7.1.2. Training the THMMs

The training of a topological hidden Markov model is an iterative process that seeks to maximize the probability that each THMM accounts for the training sample sequences. Since all 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. 10. Thus we have obtained 16 states for each of the 10 THMMs from "0" to "9", respectively. We have used the Forward–backward algorithm to estimate the matrix $\mathcal{A}$, $\mathcal{B}$, $\mathcal{U}$, and $\mathcal{V}$. While the Baum–Welch algorithm itself is well-defined, initialization of the THMMs is much tricky. To initialize $\mathcal{A}$, 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 $\mathcal{B}$ is much simpler. We just assign all $b_j(k)$'s the value $\frac{1}{8}$. The matrices $\mathcal{U}$ and $\mathcal{V}$ were initially empty, which means they do not have rows and columns before starting the training. As the training proceeds, rows and columns will emerge in the matrices.

7.1.3. Classification results

The testing phase is conducted on the MNIST 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

Table 1
The local structure numbers and their models.

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

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

Table 1
The local structure numbers and their models.

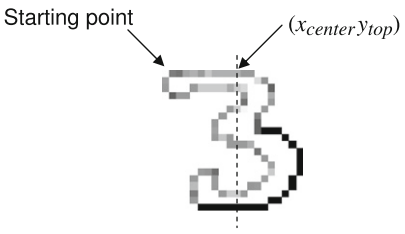


Fig. 9. The criterion for selecting the chain code starting point.

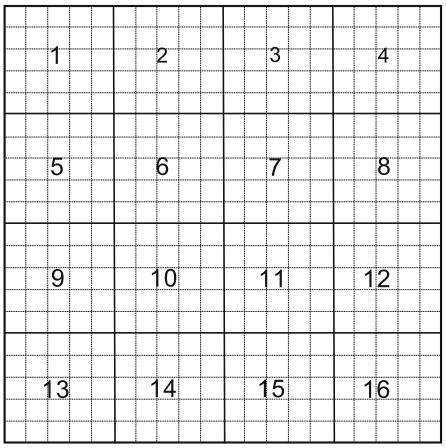


Fig. 10. Hidden states and their corresponding cells.

Table 2
Comparison of performance between THMMs and SHMMs.

Model	**Accuracy** (%)	**Reject-rate** (%)	**Error reduction** (%)
THMMs	98.5	1.24	(THMMs vs. SHMMs)
SHMMs	96.4	1.98	58.3

different probabilities assigned to structures, we used all possible structures as input to the 10 models. All 10 THMMs 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 size of the testing data set. The error-rate reduction is obtained by

$$\frac{(\text{error rate of SHMMs})-(\text{error rate of THMMs})}{(\text{error rate of SHMMs})}100\%. \quad (26)$$

We have compared the THMM's approach with the SHMMs classification technique. The SHMM's approach considers only local structures of the handwritten numeral without information about their shapes (topological decoding). In other words, SHMMs are not enough powerful to discriminate between handwritten numerals since the shape information is vital in this application [15]. The training of both classifiers was coded using MATLAB. Table 2 shows the performance comparison between THMMs and SHMMs. The error rate E which is 1.5% and the reject rate R of 1.24% confer a small $10E+R$[3] value of 16.24% to the THMM's classifier: It represents a significant improvement.

7.2. Application 2: protein fold recognition

In this section, we show how the second level THMMs can be applied to solve one of the most challenging problems in molecular biology known as: the *protein 3D-fold recognition*. This problem is stated as follows: *Given a primary structure of a protein* (*a sequence of amino acids which is the VO sequence*), *predict its 3D fold class* (*amongst 27 selected classes*).

7.2.1. Motivation

The primary structure of a protein is its linear sequence (or linear polymers) of amino acids and the location of any disulfide bridges. There are 20 amino acids which are: (A, ALA); (C, CYS); (D, ASP); (E, GLU); (F, PHE); (G, GLY); (H, HIS); (I, ILE); (K, LYS); (M, MET); (N, ASN); (P, PRO); (Q, GLN); (R, ARG); (S, SER); (T, THR); (V, VAL); (W, TRP); (L, LEU); (Y, TYR). In biochemistry and structural biology, secondary structure is the general 3D form of local segments of biopolymers such as proteins and nucleic acids (DNA/RNA). It does not, however, describe specific atomic positions in 3D space, which are considered to be tertiary structure. Each protein can be considered as a tertiary structure—a sequence of secondary structures folded in a certain way in the 3D space [30]. Fig. 11 illustrates the relationships between the primary, secondary and tertiary structures of a protein. This folding process of a protein is a global overview of the protein's energy surface [31]. It is a thermodynamically driven process—Proteins fold by reaching their thermodynamically most stable structure. However, many local and non-local interactions take part in the process, and therefore the search space of possible structures becomes enormous. As the protein databank grows larger, the proteins classification process and its folding prediction become slower and more difficult. Computational analysis of biological data obtained in genome sequencing is essential for the understanding of cellular functions and the discovery of new drugs and therapies. Since the functions of proteins may come from their 3D structures, *the method of measuring structural similarity* (*or mapping one structure to another*) *between two proteins can infer their functional proximity*. Sequence–sequence and sequence–structure comparison play a critical role in predicting a possible function for new sequences. Sequence alignment is accurate in detecting relationships between proteins. However, this method is not efficient when two proteins are structurally similar, but have no significant sequence similarity. Protein fold recognition is an important approach to structure discovery that does not rely on sequence similarity. *It consists of mapping an amino acid sequence of unknown structure to one of a library of target 3D structures* (*or folds*). Unraveling the protein 3D structure is one of the many goals needed to decode the human genome or the genome of any given pathogen.

7.2.2. Background

Researchers have been devising and applying new methods to solve this problem and a work of great value has already been undertaken. Lawrence Hunter applied heuristic Bayesian classification to define and enumerate structural motifs present in protein macromolecular systems [32]. White et al. applied a nonlinear optimal filtering algorithm to predict a protein's tertiary structure [33]. Dubchak and his team proposed a method for predicting protein folding class based on a global protein chain description that uses a voting scheme [34]. Maeda et al. proposed a classification method of protein folds using a structural transformation of one protein to another [35]. Ding et al. worked on multi-class protein fold recognition using support vector machines (SVMs) and neural networks (NNs) [36]. The multi-class SVM's approach used by Ding et al. will be compared to the second level THMM classifier in this paper. Furthermore, Jason et al. built a protein classification system which depends significantly on the choice of a "good" representation of the input sequences of amino acids [37]. Though their work achieved the state of the art classification performance, their methodology does not handle unknown and unlabeled data. *It is worth to underscore that the topological interaction between secondary structures has not been fully exploited in most of the research conducted in this area.* Our goal through the THMMs is to predict the protein fold by merging the amino acid sequence (sequential information) and the 3D folding of the secondary structures (shape information).

[3] $10E+R$ is a standard classifier comparison measure used in document analysis.

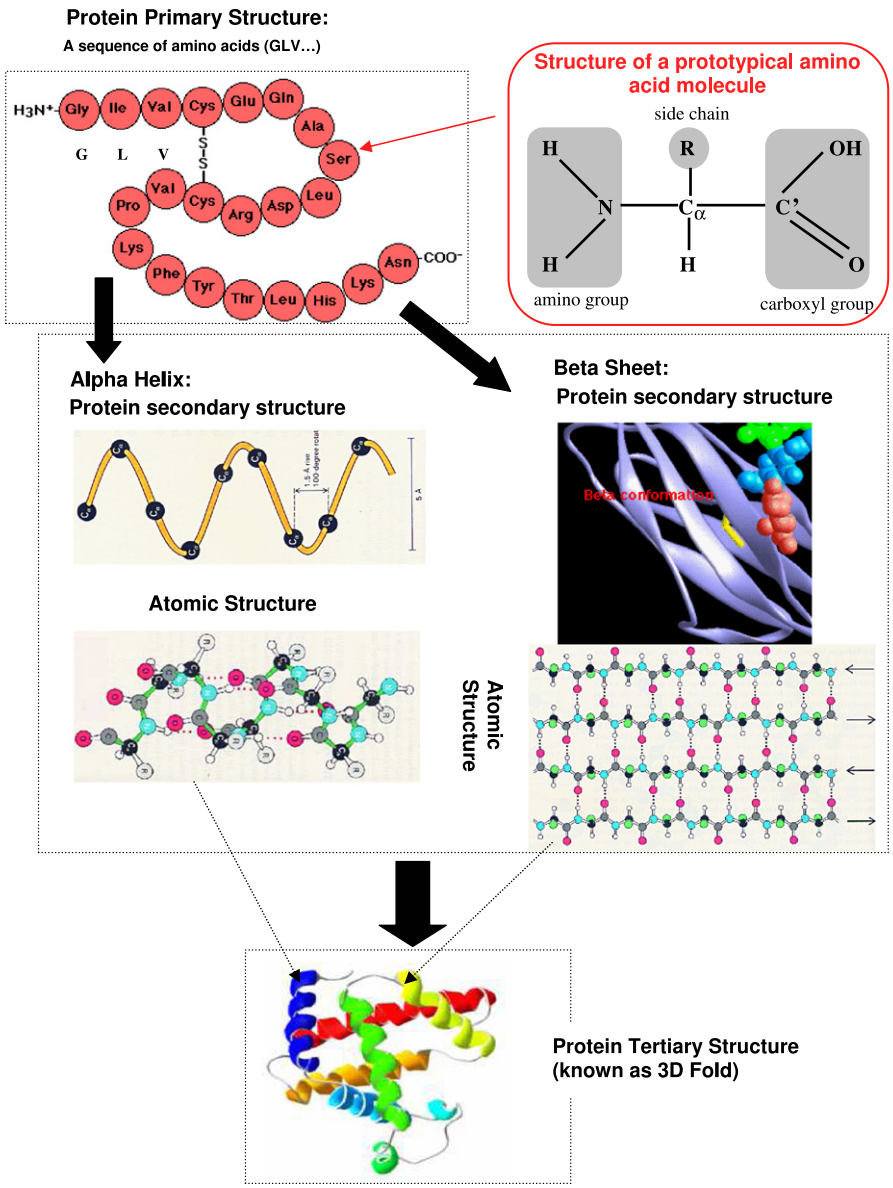


Fig. 11. The relationships between the different structures of a protein. In the amino acid molecule (top-right), the chemical groups bound to the central alpha (α) carbon are highlighted in grey. The R-group represents any of the possible 20 amino acid side chains.

7.2.3. Data collection and UNIF formation

The data set used during our experiment was extracted from the SCOP (Structural Classification Of Proteins) database. It is the PDB-40D set developed by the authors of SCOP [38]. This data set contains 990 protein primary structure sequences. In this database the entire protein sequence is already segmented into several subsequences of amino acids assigned to secondary structures. In other words, there is no need to perform any prior automatic segmentation task in this particular application. The data set we have extracted contains fixed format records of *3D Cartesian coordinates*, *occupancies* and *temperature factors* for the atoms composing the *polypeptide backbone* of the amino acid molecules of the entire protein primary structure. Fig. 12 (part[A]) and Table 3 display an example of a peptide backbone and a part of the PDB set, respectively. In this application, *each O_i is a VO subsequence of amino acids and each UNIF U_i is a protein secondary structure.* In other words, each symbol o_i of O_i is an amino acid. Fig. 13 depicts a segmented primary structure sequence O of the protein 2DKB, and its secondary structure sequence. The UNIF's assigned to the subsequences O_i are formed using the unsupervised clustering algorithm applied to their external contours $X_i(t)$ representing their shapes. In other words, the clustering of external contours vectors creates the secondary structures: *This process corresponds to the structural decoding of the THMMs.* Besides, each amino acid o_i is assigned to a set of atoms which are 3D Cartesian coordinates. These 3D coordinate points allow to capture the shape of the entire backbone (refer to Fig. 12 (part[B]). Because of the abrupt changes at any variable locations in the shape of the entire protein polypeptide backbone, we have adopted the 3D wavelet transform to optimally represent the external contour of a subsequence O_i. However, because of the lack of *shift invariance* and a poor *directional selectivity* inherent to the traditional discrete wavelet transform [39], we have used the *Dual-Tree Complex Wavelet Transform* in order to capture the contour of a subsequence O_i.

7.2.4. Training and testing

We have implemented the unsupervised clustering to partition the wavelet coefficients vectors (or contour vectors) assigned to all the different secondary structures (or UNIF's) contained in the data set. We have therefore built 16 clusters corresponding to the different folding modes of the basic four secondary structures: "Helix", "Sheet", "Turn", and "Extended" from the data set. In other words, we made a difference between an elongated "Helix" and a compressed "Helix". *This structural decoding phase is important since it is sensitive to the shape differences of the protein secondary structures*. Likewise, *the topological decoding in this application corresponds to the determination of the correct shape representing each cluster built*. Fig. 14 illustrates the major phases undertaken to create the protein secondary structures.

- (A) Training and testing without cross validation: One of our goals in the experimental phase is to consistently compare the THMMs with both the SHMMs and SVM classifiers on the same data set provided by Ding's team [36]. Therefore, we have conducted the training and testing phases based on a data set that contains 990 proteins in which 696 belong to the 27 largest folds. Ding's team extracted 605 proteins from the original data set (in which 311 proteins are part of the 27 largest folds) to train a multi-class SVM classifier based on the "one versus others" (OVO) classification method using a Gaussian kernel. In this approach, 27 fold classes have been partitioned into a two-class problem: the first class contains

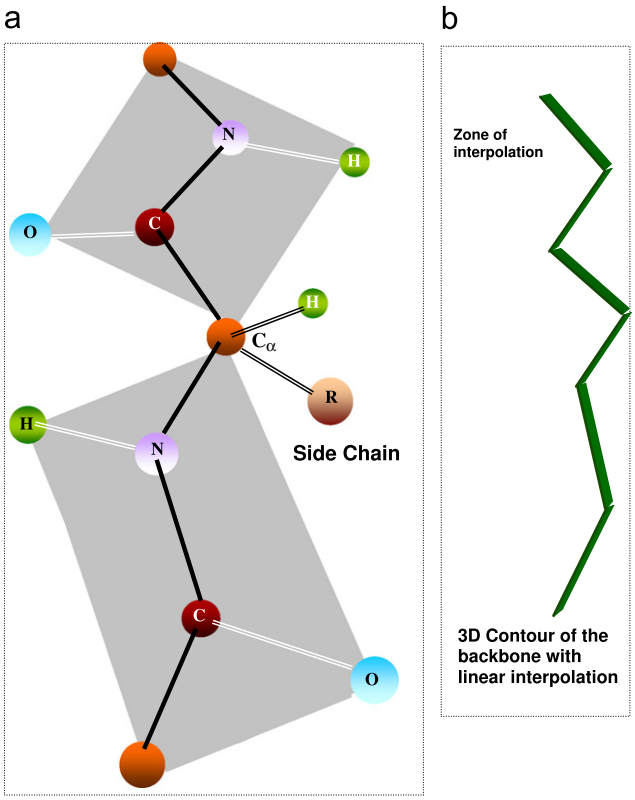


Fig. 12. Part [A] illustrates the planar peculiarity of the peptide bond. The atoms connected by the black edges represent the peptide backbone (or polypeptide backbone in the case of the entire protein primary structure). Part [B] depicts the 3D shape of a local backbone. A sequence of these shapes captures the topology of the entire protein.

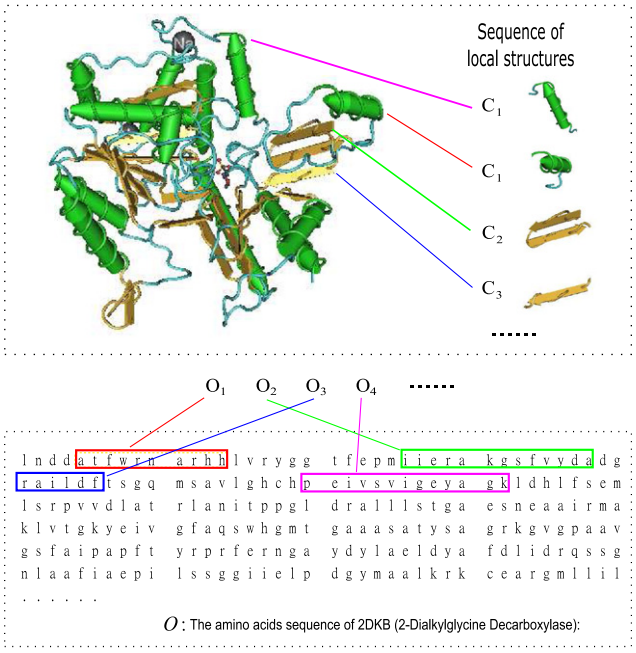


Fig. 13. The 3D structure of a protein (fold) captured by its secondary structure sequence U_j. A protein 3D fold is viewed as a combination of structural (local structures) and topological information (shapes).

Table 3
The PDB SEQRES records describe the sequence of the crystallized polymer.

SEQRES	SEQRES	…	ATOM	ATOM	ATOM	ATOM	…
1	2		1	2	9	10	
396	396		N	CA	N	CA	
MET	LEU		MET	MET	PHE	PHE	
ASP	GLY		5	5	6	6	
GLU	LEU		41.402	40.919	39.627	39.199	
ASN	ALA		11.897	13.263	14.840	15.440	
ILE	ASP		15.262	15.600	14.228	12.964	
THR	LEU		1.00	1.00	1.00	1.00	
ALA	PHE		48.61	47.70	48.66	45.33	
ALA	ARG						
PRO	ALA						
ALA	ASP						
ASP	GLU						
PRO	ARG						
ILE	PRO						

The sequence labels in the PDB ATOM reports the record name, the atom serial number, the atom name, a residue name, a residue number, the (x, y, z) Cartesian coordinates, the isotropic thermal parameter and the occupancy.

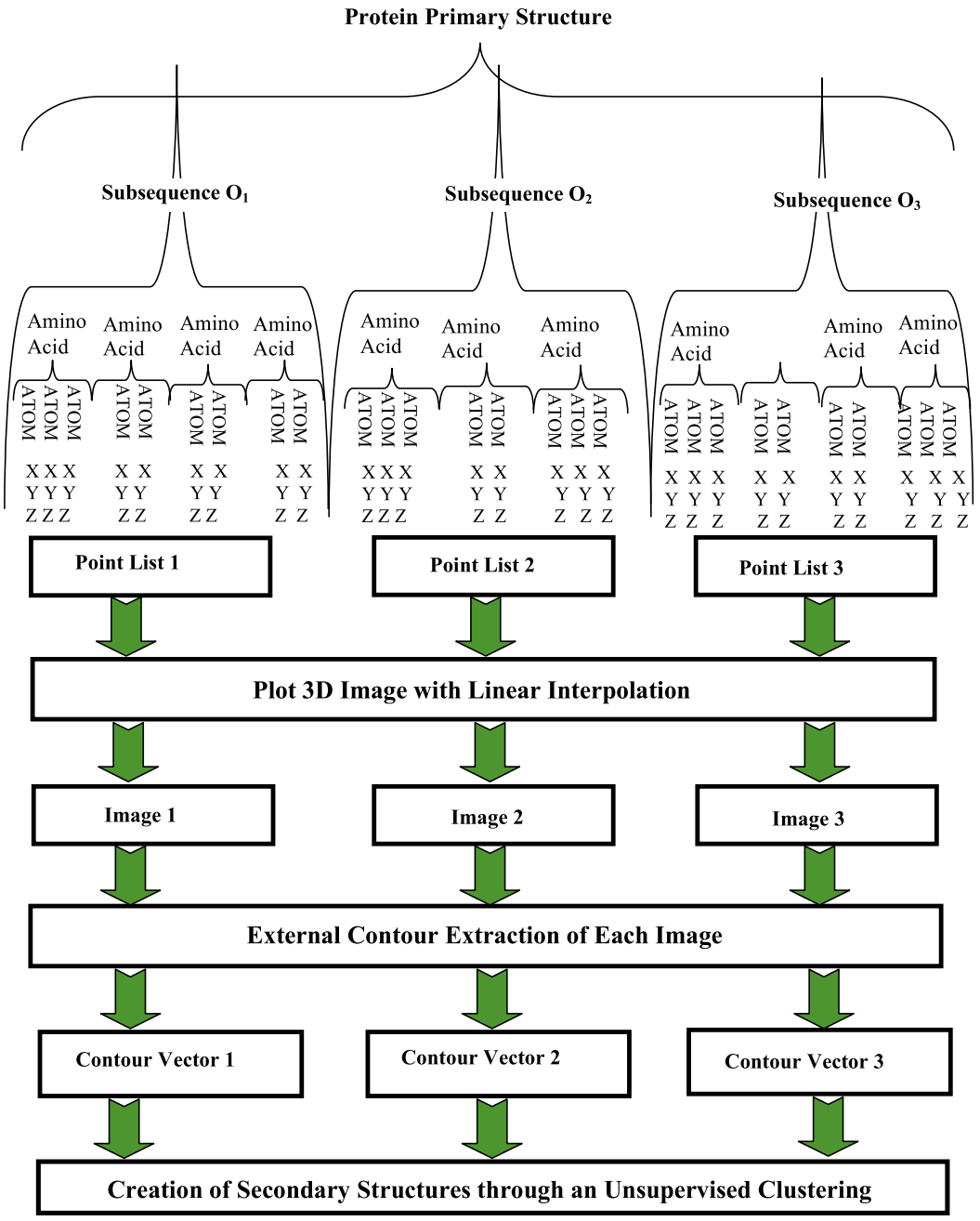


Fig. 14. The different phases undertaken to create the set of protein secondary structures. Each amino acid is composed of a certain number of atoms represented by their 3D coordinates (*X*,*Y*,*Z*).

Table 4
Subsets of 27 SCOP protein folds used for classification.

	Fold name	Fold class	# TRS	# TES
Alpha	Globin-Like	1	13	6
	Cytochrome C	3	7	9
	DNA-binding-3-helical bundle	4	12	20
	4-Helical-up-and-down bundle	7	7	8
	4-Helical cytokines	9	9	9
	EF-hand	11	7	9
Beta	Immunoglobulin-like-beta-sandwich	20	30	44
	Cupredoxins	23	9	12
	Viral-coat and capsid proteins	26	16	13
	Con A-like-lectins/ glucaneses	30	7	6
	SH3-like-barrel	31	8	8
	OB-fold	32	13	19
	Trefoil	33	8	4

Table 4 (*continued*)

	Fold name	Fold class	# TRS	# TES
	Trypsin-like serine proteases	35	9	4
	Lipocalins	39	9	7
Alpha/beta	TIm-barrel	46	29	48
	FAD-binding-motif	47	11	12
	Flavodoxin-Like	48	11	13
	NAD-binding Rossmann-fold	51	13	27
	P-loop	54	10	12
	Thioredoxin-like	57	9	8
	Ribonuclease H-like motif	59	10	14
	Hydrolases	62	11	7
	Periplasmic binding protein-Like	69	11	4
Alpha+Beta	Beta-grasp	72	7	8
	Ferredoxin-like	87	13	27
	Small inhibitors, toxins, lectins,…,	110	12	27

The terms "Fold class", "#TRS" and "#TES" stand for class label, size of the training set and size of the testing set, respectively.

objects in one "true" class, and the "others" class combines all other classes. A two class classifier is therefore trained for this two-class problem. This procedure has been repeated for each of the 27 fold classes. Because their main focus was on the 27 largest folds, therefore they have used 385 proteins out of the 696 largest folds for testing (311 proteins for training and 385 for testing in total) (refer to Table 4 for details). The testing set of proteins did not participate in the training phase. Furthermore, since there are 27 protein fold classes in the data set, therefore we have built 27 second level THMM models. Each protein primary structure sequence has been tested on all 27 THMMs. The model who generated the highest score was the selected fold class assigned to that protein sequence. The rate of success of the THMMs is represented by its accuracy (1-error rate). In order to exploit the strengths of SHMMs and SVM (OVO) simultaneously, we have combined the results of both classifiers viewed as "black-boxes". This procedure is justified by the fact that a combination of classifiers performs usually better than a single classifier [40]. *A multi-classifier system is a powerful solution to difficult pattern recognition problems involving large class sets and noisy input*. In our experiment, we have adopted "the Borda Count" (BC) strategy to determine the final classification results. We assumed both SHMMs and SVM (OVO) to have the same weight in decision making. The Borda Count for a class is the sum of the number of classes ranked below it by each classifier. It represents a measure of the *strength of agreement* of the classifiers that the input protein belongs to that class. The combination ranking is given by arranging the classes so that their BCs are in descending order. The combination classifier output is the class with the highest BC.

- (B) Training and testing using cross validation: In order to measure the power of generalization of the THMMs classifier, we have also conducted a 5-fold cross validation estimation technique on the 696 proteins. We divided the 27 largest folds set containing 696 proteins into five sets, each of which contains 139 proteins. We then selected one set for testing and the other 4 sets (556 proteins) for training. We repeated this procedure 5 times with each time selecting a different set for a validation data. We have tested the THMM classifier and the five accuracy results obtained for each fold of the 27 fold classes are then averaged to produce a single accuracy estimation for each fold class. A test is deemed "correct" if the predicted class of the input fold is exactly the true class of this fold, otherwise it is considered "incorrect".

In order to capture the correlation between classes that is absent in the SVM (OVO) classifier, we adopted the multi-class kernel-based vector machines (MKVM) approach proposed in [41]. We have implemented "the basic algorithm for learning a multiclass, kernel-based support vector machine using KKT (Karush–Kuhn–Tucker) conditions" as described in [41]. We used a Gaussian kernel and set the value of $\varepsilon = 0.001$. The β value and the standard deviation σ of the kernel function in this algorithm were both determined using the 5-fold cross validation scheme on the 696 proteins.

7.2.5. Classification results and analysis

We have compared the THMMs classifier with both the SHMMs, and the SVM (OVO) classifiers individually as well as with their combination SHMMs/SVO. In the experiments we have run, the Gaussian kernel worked far better than the linear and the polynomial kernels. The results depicted in Table 5 show the prediction accuracy for every protein fold class of the 27 fold classes using all classifiers without applying the 5-fold cross validation estimation technique. The average accuracy has also been computed for each classifier. These results demonstrate the superiority in performance of the THMMs over the other classifiers. In order to confirm that the SHMM average prediction accuracy is significantly different from the SVM, we have conducted a 5×2 CV paired t-test of hypothesis. Dietterich [42] has studied the properties of 10-fold cross validation followed by a paired t-test for determining whether there is a significant difference between the averages of the prediction accuracy of two classifiers. He found out that such a test suffers from higher than expected type I error (Prob (reject $H_0|H_0 = true$)). To remove this pitfall, he proposed a new test: the 5×2-fold cross validation. In this test, 2-fold cross-validation is executed 5 times (five replications) resulting in 10 accuracy values. The data are reshuffled and restratified after each replication. All 10 values are used for average accuracy estimation in the t-test but only values from one of the five 2-fold cross validation rounds are used to estimate the variance. The *null hypothesis* H_0 for the five 2-fold cross validation paired t-test is that the two prediction accuracy averages are equal: H_0: $\mu_{SVM} = \mu_{SHMM}$, whereas the *alternate hypothesis* H_a: $\mu_{SVM} \neq \mu_{SHMM}$.

Let $p_i^{(j)}$ represents the difference between the error rates (1-precision) of SVM and SHMM classifiers on fold j=1,2 of replication i=1,…,5. Let

$$\overline{p}_i = \frac{(p_i^1 + p_i^2)}{2}, \tag{27}$$

Table 5
Prediction accuracy using THMMs (second level), multi-class SVM (OVO), SHMMs, and the combination of SHMMs/SVM (OVO) without applying the 5-fold cross validation estimation technique.

Fold class	SVM (OVO)	SHMMs	SHMMs/SVM (OVO)	THMMs
1	87.5	83.3	87.5	96.2
3	50.9	77.8	88.9	93.2
4	43.7	35.0	50.0	58.3
7	53.5	100.0	100.0	100.0
9	69.8	50.0	77.8	83.9
11	50.0	66.7	66.7	73.3
20	48.6	56.6	59.1	65.5
23	15.3	33.3	33.3	52.3
26	46.8	34.7	61.5	71.0
30	25.0	33.3	33.3	50.0
31	41.9	50.0	75.0	82.2
32	27.4	26.0	42.1	51.3
33	50.0	75.5	50.0	78.1
35	25.0	25.0	50.0	62.2
39	39.3	50.0	71.4	79.3
46	60.5	50.0	60.4	71.0
47	56.9	58.3	66.7	74.0
48	29.5	34.7	38.4	48.0
51	31.2	30.0	48.1	56.2
54	47.2	60.0	60.0	73.1
57	25.0	75.0	50.0	79.2
59	39.3	35.7	35.7	48.4
62	78.6	85.7	85.7	92.0
69	25.0	50.0	100.0	79.0
72	25.0	50.0	75.0	81.4
87	24.5	33.3	44.4	58.0
110	69.3	33.3	51.8	64.4
Average	**45.2**	**51.6**	**61.6**	**71.16**

$$S_i^2 = (p_i^{(1)} - \overline{p}_i)^2 + (p_i^{(2)} - \overline{p}_i)^2, \tag{28}$$

denote the mean of the difference between the error rates of SVM and SHMM and the variance estimation, respectively. Therefore, the two-sided test rule states that the null hypothesis should be accepted with level of significance α if the following statistics:

$$\frac{p_1^{(1)}}{\sqrt{\sum_i (s_i^2/5)}} \sim t_5 \quad \text{(student's t distribution with 5 degrees of freedom)} \tag{29}$$

belongs to the $100 \times (1-\alpha)$ confidence interval. This can be written as

$$\frac{p_1^{(1)}}{\sqrt{\sum_i (s_i^2/5)}} \in (-t_{\alpha/2,5}, t_{\alpha/2,5}). \tag{30}$$

We have set the level of significance $\alpha = 0.05$, a lookup to the student distribution table indicates $t_{0.025,5} \approx 3.365$ and hence provides a confidence interval of $(-3.365, 3.365)$. The statistic value defined in Eq. (29) has been computed for the sample of size 27 classes and found outside the confidence interval which states that *the null hypothesis is rejected and the alternate hypothesis is therefore accepted.* In conclusion, the SHMM classifier precision is significantly different from the SVM classifier precision.

Moreover, it is important to outline that the features implemented in [36] were based on statistical information (such as "composition", "transition", and "distribution") of amino acids. Therefore, their feature extraction phase did not take into account *the order* of the secondary structures found in the whole sequence. However, in the THMM's approach, it is the sequence of secondary structures and their shapes that capture the protein 3D fold. *In conclusion, the second level THMMs tend to consistently model genomic and proteomic data at the same time.* The experiments have revealed the following: (i) SHMMs appear to perform better than the SVM when the input has a long protein sequence composed of the same secondary structures. (ii) SVM is more appropriate to recognize shorter primary structures composed of different secondary structures. (iii) The combination of SHMMs and SVM has reduced this erratic behavior by complementing both classifiers. This combination has globally improved the accuracy. (iv) The 3D shape of the protein backbone modeled via the THMMs is a distinguishing feature that further enhances the global prediction accuracy.

Likewise, the 5-fold cross validation estimation has shown a slightly lower recognition prediction accuracy in the average in the case of the THMMs and the SVM compared to the "train and test" method (without CV). However, these differences are deemed not to be very significant. Table 6 depicts the accuracy of the second level THMMs and the SVM (MKVM) classifiers using the 5-fold cross validation (CV) estimation technique.

8. Conclusion

We have presented a machine learning paradigm that extends the traditional HMMs by (i) extracting local structures of a visible observation sequence, and (ii) embedding the state-transition graph in a Euclidean space to unravel shape information about these local structures. Our approach acts on the visible observation sequence by determining their UNIFs. Therefore, the THMM's approach is well-suited to: (i) exploit long-range dependencies, and (ii) account for metric information associated to the pattern. THMMs extend several HMMs-based paradigms that are not adequate to gain an insight into the structural world. The protein fold mapping application shows that the THMM formalism holds promise since it has significantly outperformed both the SVM, and the SHMMs classifiers. The performance obtained in this application suggests that different 3D shape representation techniques

Table 6
Prediction accuracy using the second level THMMs and the SVM (MKVM) using a 5-fold cross validation (CV) estimation technique.

Fold class	SVM (MKVM + CV)	Second Level THMMs + CV
1	87.2	95.3
3	50.2	92.0
4	43.3	59.1
7	57.3	100.0
9	70.8	82.3
11	50.3	74.2
20	53.9	63.3
23	17.1	50.6
26	47.2	73.3
30	25.2	50.5
31	42.2	83.6
32	30.5	50.8
33	48.3	77.5
35	25.9	62.8
39	39.1	78.3
46	66.2	69.6
47	55.1	72.3
48	29.1	46.2
51	32.2	54.3
54	53.3	74.5
57	25.1	80.1
59	43.5	47.2
62	75.1	90.3
69	24.1	80.1
72	27.9	80.2
87	24.5	56.1
110	70.1	64.2
Average	**44.98**	**70.69**

should be experimented and evaluated. This task is part of our future investigation. We believe that this embedding of topology will open a new area in which dynamic Bayesian networks can exploit more powerful topological features such as homeomorphism, homotopy equivalence and topological invariance.

References

[1] L. Baum, T. Petrie, Statistical inference for probabilistic functions of finite state Markov chain, Annals Mathematical Statistics 37 (1966) 1554–1563.
[2] L. Rabiner, B. Juang, Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cliffs, NJ, 1993.
[3] I. Sanches, Noise-compensated hidden Markov models, IEEE Transactions on Speech and Audio Processing 8 (5) (2000) 533–540.
[4] G. Fan, X. Xia, Improved hidden Markov models in the wavelet-domain, IEEE Transactions on Signal Processing 49 (1) (2001) 115–120.
[5] J.S. Evans, V. Krishnamurthy, Hidden Markov model state estimation with randomly delayed observations, IEEE Transactions: Signal Processing 47 (8) (1999) 2157–2166.
[6] Z. Sun, W. Jiang, J. Sun, Adaptive Online Multi-Stroke Sketch Recognition based on Hidden Markov Model, Advances in Machine Learning and Cybernetics, vol. 3930, Springer Berlin/Heidelberg, 2006, pp. 948–957.
[7] D. Bouchaffra, V. Govindaraju, S. Srihari, Postprocessing of recognized strings using nonstationary Markovian models, IEEE Transactions: Pattern Analysis and Machine Intelligence 21 (10) (1999) 990–999.
[8] C.C. Tappert, C.Y. Suen, T. Wakahara, The state of the art in online handwriting recognition, IEEE Transactions: Pattern Analysis and Machine Intelligence 12 (8) (1990) 787–808.
[9] J. Li, A. Najmi, R. Gray, Image classification by a two-dimensional hidden Markov model, IEEE Transactions on Signal Processing 48 (2) (2000) 517–533.
[10] K. Asai, S. Hayamizu, H. Handa, Prediction of protein secondary structures by hidden Markov models, Computer Application in the Biosciences (CABIOS) 9 (2) (1993) 141–146.
[11] D. Hernandez-Hernandez, S. Marcus, P. Fard, Analysis of a risk-sensitive control problem for hidden Markov chains, IEEE Transactions on Automatic Control 44 (5) (1999) 1093–1100.
[12] M. Gemignani, Elementary Topology, second ed., Dover Publications, Inc., New York, 1990.
[13] S. Fine, Y. Singer, N. Tishby, The hierarchical hidden Markov models: analysis and applications, Machine Learning 32 (1) (1998) 41–62.
[14] K. Murphy, M. Paskin, Linear time inference in hierarchical HMMs, in: Proceedings of Neural Information Processing Systems, Boston, USA, 2001, pp. 833–840.
[15] D. Bouchaffra, J. Tan, Structural hidden Markov models using a relation of equivalence: application to automotive designs, Data Mining and Knowledge Discovery Journal 12 (1) (2006) 79–96.
[16] M. Brand, N. Oliver, A. Pentland, Coupled hidden Markov models for complex action recognition, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Puerto Rico, June 17–June 19, 1997, pp. 994–1000.
[17] Z. Ghahramani, M. Jordan, Factorial hidden Markov models, Machine Learning 29 (2–3) (1997) 245–273.
[18] T. Kristjansson, B. Frey, T. Huang, Event-coupled hidden Markov models, in: IEEE International Conference on Multimedia and Exposition, vol. 1, 2000, pp. 385–388.
[19] Y. Bengio, P. Frasconi, Input–output HMMs for sequence processing, IEEE Transactions: Neural Networks 7 (5) (1996) 1231–1249.
[20] E. Bienenstock, S. Geman, D. Potter, Compositionality, MDL priors, and object recognition, in: M.C. Mozer, M.I. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, vol. 9, MIT Press, Cambridge, 1997, pp. 838–844.
[21] D. Bouchaffra, Embedding HMMs-based models in a Euclidean space: the topological hidden Markov models, in: The 19th IEEE International Conference on Pattern Recognition (ICPR) Proceedings (Oral Presentation), Convention Center, Tampa Florida, 2008, pp. 1–4.
[22] S. Eddy, Profile hidden Markov models, Bioinformatics 14 (9) (1998) 755–763.
[23] C. Lin, C. Hwang, New forms of shape invariants from elliptic fourier descriptors, Pattern Recognition 20 (5) (1987) 535–545.
[24] S.-T. Bow, Pattern Recognition and Image Preprocessing, second ed., Marcel Dekker, Inc., 2002.
[25] I. Biederman, G. Ju, Surface vs. edge-based determinants of visual recognition, Cognitive Psychology 20 (1) (1988) 38–64.
[26] R. Bellman, On the approximation of curves by line segments using dynamic programming, Communication of the ACM 4 (6) (1961) 284.
[27] R. Duda, P. Hart, D. Stork, Pattern Classification, Wiley, New York, 2001.
[28] H. Freeman, On the encoding of arbitrary geometric configurations, IRE Transactions on Electronic Computers EC-10 (1961) 260–268.
[29] Y. Mori, K. Honda, H. Ichihashi, A unified view of probabilistic pca and regularized linear fuzzy clustering, in: Proceedings of the International Joint Conference on Neural Networks, no. 1, 2004, pp. 20–24.
[30] P.E. Bourne, H. Weissig, Structural Bioinformatics, Wiley-Liss, 2003.
[31] J.N. Onuchic, Theory of protein folding: the energy landscape perspective, Annual Review of Physical Chemistry 48 (1997) 545–600.
[32] L. Hunter, D. States, Bayesian classification of protein structural elements, in: Proceedings of the Twenty-fourth Annual Hawaii International Conference on Systems Sciences, vol. 1, 1991, pp. 595–604.
[33] J. White, C. Stultz, T. Smith, Protein classification by stochastic modeling and optimal filtering of amino-acid sequences, Mathematical Bioscience 119 (1) (1994) 35–75.
[34] I. Dubchak, I. Muchnik, S. Holbrook, S. Kim, Prediction of protein folding class using global description of amino acid sequence, Proceedings of the National Academy os Science 92 (1995) 8700–8704.
[35] T. Maeda, K. Kamada, N.T. Ohkawa, H. Nakamura, A. Kidera, Feature extraction of protein folds based on secondary structure transformation, in: Proceedings of the IEEE International Joint Symposia on Intelligence and Systems, March 21–March 23, Rockville, Maryland, 1998, pp. 158–162.
[36] C. Ding, I. Dubchak, Multi-class protein fold recognition using support vector machines and neural networks, Bioinformatics 17 (4) (2001) 349–358.
[37] J. Weston, C. Leslie, E. Ie, D. Zhou, A.W. Elisseeff, Noble, semi-supervised protein classification using cluster kernels bioinformatics, Bioinformatics 21 (15) (2005) 3241–3247.
[38] L. Lo Conte, B. Ailey, T. hubbard, S. Brenner, A.G. Murzin, C. Chothia, Scop: a structural classification of proteins database, Nucleic Acids Research 28 (2000) 257–259.
[39] N.G. Kingsbury, The dual-tree complex wavelet transform: a new technique for shift-invariance and directional filters, in: Proceedings of the 8th IEEE Digital Signal Processing Workshop, No. 86, Bryce Canyon, Utah, USA, 1998.
[40] G. Fumera, F. Roli, A theoretical and experimental analysis of linear combiners for multiple classifier systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (6) (2005) 942–956.
[41] K. Crammer, Y. Singer, On the algorithmic implementation of multi-class kernel-based vector machines, Journal of Machine Learning (2001) 265–292.
[42] T.G. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Computation 10 (7) (1998) 1895–1923.

About the Author—DJAMEL BOUCHAFFRA (www.djamel-bouchaffra.info) is an Associate Professor of Computer Science at Grambling State University, LA. Dr. Bouchaffra has been selected as the recipient of several teaching excellence awards.

His field of research is in Pattern Recognition, Machine Learning, Computer Vision and Adaptive Artificial Intelligence. Professor Bouchaffra introduced both the structural and the topological hidden Markov models as two paradigms that attempt to merge discrete and continuous structures together. His areas of application include bioinformatics, biometrics, watermarking, speech, and handwriting recognition.

He has written several papers in peer-reviewed conferences and premier journals. He chaired several sessions in conferences. He is a reviewer for funding agencies such as NASA and journals such as IEEE TPAMI, TNN, TKDE. He is one of the two Guest-Editors of the special issue of the Journal of Pattern Recognition (PR): ''Feature Generation and Machine Learning for Robust Multimodal Biometrics'' of the PR volume 41/3.

Professor Bouchaffra is an editorial board member of the ''Pattern Recognition Journal'' (Elsevier), ''The Open Information Systems Journal'', (Bentham Science), an associate editor for the Journal ''Advances in Artificial Intelligence'' (Hindawi), ''the Journal of Engineering Letters'' (IAE) and the ''Scientific Journals International'' (SJI).

Dr. Bouchaffra is a senior member of the IEEE, and a member of the IEEE Computer Society.