Data Mining and Knowledge Discovery, 12, 79–96, 2006

DOI: 10.1007/s10618-005-0020-8

Structural Hidden Markov Models Using a Relation of Equivalence: Application to Automotive Designs

D. BOUCHAFFRA dbouchaffra@ieee.org
J. TAN
Department of Computer Science & Engineering, Oakland University, 131 Dodge Hall, Rochester, MI, 48309, USA

Revised March 3, 2005; Accepted September 21, 2005

Published online: 3 February 2006

Abstract. Standard hidden Markov models (HMM's) have been studied extensively in the last two decades. It is well known that these models assume state conditional independence of the observations. Therefore, they are inadequate for classification of complex and highly structured patterns. Nowadays, the need for new statistical models that are capable to cope with structural time series data is increasing. We propose in this paper a novel paradigm that we named "structural hidden Markov model" (SHMM). It extends traditional HMM's by partitioning the set of observation sequences into classes of equivalences. **These observation sequences are related in the sense they all contribute to produce a particular local structure.** We describe four basic problems that are assigned to a structural hidden Markov model: (1) probability evaluation, (2) statistical decoding, (3) local structure decoding, and (4) parameter estimation. We have applied SHMM in order to mine customers' preferences for automotive designs. The results reported in this application show that SHMM's outperform the traditional hidden Markov model with a 9% of increase in accuracy.

Keywords: Hidden Markov models, relation of equivalence, local structures, statistical decoding, structural decoding.

1. Introduction

Hidden Markov models (HMM's) is a widely used approach that models time-series problems from a statistical view. The relevance of HMM's was first demonstrated in speech processing and recognition in the late 1980's (Rabiner & Juang, 1993; Gales, 2000; Sanches, 2000). Neighbor areas such as signal processing (Fan & Xia, 2001), and handwriting and text recognition (Bouchaffra, Govindaraju, & Srihari, 1999) have also benefited almost at the same time from these stochastic models. Half a decade later, HMM's spread to many other areas such as image processing and computer vision (Li, Najmi, & Gray, 2000), biosciences (Asai & Handa, 1993), control (Hernandez-Hernandez, Marcus, & Fard, 1999), and others. Promising results have been obtained from the use of HMM's in several applications in the aforementioned areas. However, the number of problems where HMM's can be applied is insignificant compared to all the problems we can encounter. In other words, the use of HMM's is rare within the whole scientific community. *The main reason behind this limitation comes from the fact that HMM's have a clear conceptual framework and the ability to learn statistically, but they are unable to account for long-range dependencies which unfold structural information (Duda, Hart, & Stork, 2001; Ripley, 1996)*. Because the symbols

of an input sequence are assumed to be state conditionally independent, therefore, HMM's make no use of structure, either topological or conceptual (Gemignani, 1990).

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

To face this challenge, a few number of approaches that attempt to marry statistics with syntax have been proposed in the context of HMM's. Cai and Liu's approach integrates the statistical and structural information for unconstrained handwritten numeral recognition. Their method uses macro-states to model pattern structures (Cai & Liu, 1999). However, besides the fact that this method uses statistical and structural information in two different steps, their methodology is application-driven and therefore is *very specific*. Zhu and Garcia-Frias proposed two novel generative methods which make use of probabilistic context-free grammars and HMM's respectively to model the end-to-end error profile of radio channels (Zhu & Frias, 2004). However, they did not provide a tool to merge standard HMM's with probabilistic context free grammars into a single probabilistic framework. An other promising approach that can contribute in building structural hidden Markov models is due to Geman's work in vision. He introduced compositionality as an ability to construct hierarchical representations of scenes, whereby constituents are viewed in an *infinite variety* of relational compositions. Amongst all possible composition rules that embed syntactical information, statistical criteria such as MDL (Minimum Description Length) and Gibbs distribution are being used in order to select the optimal interpretation (Geman, et al., 2004). This approach merges statistics with syntax but unfortunately due to the greedy compositionality process, it is intractable.

Because of the gap between statistics and syntax, we propose in this paper a methodology that extends standard HMM's to account for structural information. This new paradigm that we called *structural hidden Markov model* (SHMM) decomposes the whole pattern into "meaningful" entities and assigns them syntactic information. This assignment is performed via a relation of equivalence defined in the set of observation sequences. The classes of equivalence obtained are called the *local structures* of the whole pattern. The concept of SHMM enables to capture the whole pattern by revealing its local structures one by one. Because our world is full of objects with complex structures, we believe that the concept of SHMM will leapfrog the state-of-the-arts by attempting to unfold these structures.

The organization of this paper is as follows: Section 2 describes the traditional hidden Markov model. Section 3 introduces the novel concept of structural hidden Markov model. In this very section, we cover the optimal segmentation problem, the notion of local structure and the SHMM paradigm. A selected application and experiments are presented in Section 4. Finally, the conclusion and the future work are laid in Section 5.

2. The traditional hidden markov model

The concept of hidden Markov model (HMM) has been introduced in the sixties by Baum and his colleagues (Baum & Petrie, 1966). However, the widespread application of the theory of HMM's to speech processing has occurred in the seventies (Rabiner &

Juang, 1993). To better understand the contribution of our approach, we will provide in this paper a summarized description of an HMM.

Definition 2.1. *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.*

An example of HMM is the following: Assume you are in a boat where you cannot see outside (you cannot see where the fishermen are fishing and what are they catching). The fishermen are just telling you the result of their hidden experiment (or their single catch within a particular area in the sea). An area in the sea is characterized by its depth, its water temperature, its salt density, etc. Thus a sequence of hidden experiments is performed, with the observation sequence which results in a sequence of fishes such as: "salmon, trout, cat fish, ..., sea bass denoted as: $o_1, o_2, \ldots, o_T$, where each $o_i \in \Sigma$. Given this scenario, the problem of interest is how do we build an HMM (a model) that explains the observed sequence of fish? The first problem one faces is deciding what the hidden states in the model correspond to, and then deciding how many states should be in the model. One possible choice would be to assume that there are N areas $a_1, a_2, \ldots, a_N$ in the sea that produced the fishes. Each area in the sea is a hidden state, characterized by a probability distribution, and transitions between hidden states are characterized by a state transition matrix.

2.1. Elements of an HMM

The above examples provides a good idea of what an HMM is and how it can be applied to some simple scenarios. We now introduce the elements of an HMM, and explain how the model generates observation sequences. An HMM is characterized by:

- **N**, the number of hidden states q_i in the model. The states are often called hidden but can be given a physical significance. In our example, the hidden states are the different areas described by the following features (depth, water salt density, water temperature, etc). These areas govern the observation distribution of the fish sequence. In other words, these areas explain why we have caught a particular type of fish rather than another type. These hidden states are connected (ergodic chain), which means that any area can be reached from any other area.
- **M**, the number of distinct observation (or possible fishes) per hidden state, i.e., the size of the discrete alphabet. The observation symbols $o_i \in \Sigma$ correspond to the output of the system being modeled.
- The initial state distribution $\pi = \{\pi_i\}$, where: $\pi_i = P(q_1 = S_i)$, $1 \leq i \leq N$, and $\Sigma_i \pi_i = 1$.
- The state transition probability distribution $\mathcal{A} = \{a_{ij}\}$, where: $a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i)$, $1 \leq i, j \leq N$, and $\Sigma_j a_{ij} = 1$.
- The observation symbol probability distribution in state j, $\mathcal{B} = \{b_j(k)\}$, where: $b_j(k) = P(o_k \text{ at time } t \mid q_t = S_j)$, $1 \leq k \leq M$ and $1 \leq j \leq N$, and $\Sigma_k b_j(k) = 1$.

In order to show the parameters involved in an HMM and distinguish between several HMM's, an HMM λ is usually represented as $\lambda = [\pi, \mathcal{A}, \mathcal{B}]$.

2.2. *The three basic problems of an HMM*

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

1. **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.
2. **Decoding:** Suppose we have an HMM λ as well as a sequence of observation O. Determine the most likely sequence of hidden states $q_1, q_2, \ldots q_T$ that generated the sequence of observation O.
3. **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 set of training observation sequences, determine these parameters.

In order to understand the next sections which represent the main contributions in this paper, let's focus on the evaluation problem. Let $O = (o_1 o_2 \ldots o_T)$ be the observation sequence of length T and $q = (q_1 q_2 \ldots q_T)$ be the state sequence where q_1 is the initial state. The evaluation problem is mathematically expressed as follows: Given a model λ, and the observation sequence O, evaluate the match between λ and the observation sequence O by computing $P(O \mid \lambda)$:

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

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

and using state conditional independence, we obtain:

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

The evaluation problem is based on the state conditional independence of the observation symbols. *However, there are several scenarios where the conditional independence assumption doesn't hold*. For example, while standard HMM's perform well in recognizing amino acids and consequent construction of proteins from the first level structure of DNA sequences (Krogh, et al. 1994; Eddy, 1998), 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 (Asai & Handa, 1993). Therefore, there is a need to balance the loss incurred by this state conditional independence assumption inherent to HMM's.

3. The concept of structural HMM

In this section, we introduce a mathematical description of the SHMM concept that goes beyond the traditional hidden Markov model since it emphasizes the structure (or syntax) of the visible sequence of observation. *Our idea is that a complex pattern*

$O = o_1, o_2, \ldots o_T$ can be viewed as a sequence of constituents O_i made of strings of symbols $o_i \in \Sigma$ interrelated in some way.[1] Therefore, each observation sequence O is not only one sequence in which all observations are conditionally independent, but a sequence that is divided into a series of s strings $O_i = (o_{i_1} o_{i_2} \ldots o_{i_{r_i}})$ $(1 \leq i \leq s)$. The question that remains to be addressed is how to segment an entire complex pattern into s meaningful pieces so that "local structures" assigned to these pieces can unfold ?

3.1. Optimal segmentation of the entire pattern

Our goal in this section 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 (or an entire pattern) $O = (o_1 o_2 \ldots o_T)$, 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 $1 = b_1 < b_2 < \ldots < b_s < b_{s+1} = T + 1$, 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 (or the application at hand), different error measures can be investigated. In order to remain general in our approach, we propose in this paper the following error measure:

$$\Xi(O_i) = \sum_{o_i \in O_i} d^2(o_i, \bar{o}_i), \tag{3}$$

where $\bar{o}_i$ is the *representative* symbol of the segment O_i and d is a metric that depends on the nature of data involved in the application. Appropriate metrics will be used in the case the data are real valued, ordinal or purely nominal. If the data are real valued and defined in an Euclidean space, therefore the representative symbol is the *mean* and the error measure in this case is simply the variance.

Since there are several possible segmentations $\Delta \in Seg_s(O)$, therefore the global error measure is defined as:

$$\Xi(O, \Delta) = \sum_{O_i \in \Delta} \sum_{o_i \in O_i} d^2(o_i, \bar{o}_i). \tag{4}$$

In conclusion, the optimal segmentation task consists of finding the optimal segmentation $\Delta^* \in Seg_s(O)$ such that:

$$\Delta^* = \arg \min_{\Delta \in Seg_s(O)} \left[\sum_{O_i \in \Delta} \sum_{o_i \in O_i} d^2(o_i, \bar{o}_i) \right]. \tag{5}$$

Dynamic programming approaches can be used to solve this problem in a tractable and efficient manner (Bellman, 1961; Churchill, 1989). However, the optimal solution may not be unique. There could be more than one segmentation Δ that minimize the

error measure $\Xi(O, \Delta)$. Our strategy consists of selecting the one that has the smallest number of segments (smallest value of s).

3.2. The notion of local structures

Once the optimal segmentation of the entire pattern O is determined, we can define the notion of *local structure* that is inherent to SHMM's. The symbols of a string O_i are related in the sense that they define a *local structure* $C_j \in \mathcal{S}$ of the whole complex pattern. *This structural information is captured through a relation of equivalence*. Each class of equivalence gathers strings that are similar in some meaningful sense. The organization of the symbols O_i contributes to the production of a local structure C_j. For example, a cloud of points representing a sequence of observations O_i forms a round shape C_j with a certain probability $P(C_j \mid O_i)$. This round shape is viewed as a class of equivalence that contains for example all circular or elliptical shapes. Similarly, a sequence of phonemes produces a particular word with a certain probability depending on the context. This word class contains all possible pronunciation of this particular word. Therefore, we define the local structures as:

Definition 3.1. *Consider the set $\mathcal{S}$ of all observation sequences O_i, we define a binary relation "$\sim$" on $\mathcal{S}$ as follows:*

$$\begin{cases} O_i \sim O_j \Longleftrightarrow d(O_i, O_j) \leq \epsilon, \\ \forall O_k \in \mathcal{S}, d(O_i, O_k) \leq \epsilon \Longrightarrow d(O_j, O_k) \leq \epsilon, \end{cases} \tag{6}$$

where ε is a threshold value.

The type of metric (or topology) d used will depend on the nature of the data governed by the application at hand. In the case of discrete observation sequences, the traditional edit distance (minimum operation cost to transform one string into another) with uniform costs assigned to these operations can be used. It can easily be proven that the relation "$\sim$" is *reflexive, symmetric and transitive,* therefore, it is an equivalence relation.

Definition 3.2. *By partitioning the set $\mathcal{S}$ into classes of equivalences. Each class C denoted by [C] is called a local structure and belongs to the set of equivalence classes $\mathcal{S}$.*

Other classification methods based on rough set theory can be applied. The similarity measure between two patterns is described by a distance function of all constituents attributes that are tolerant (Kim & Bang, 2000; Zhou & Chen, 2002).

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, statistics and syntax are merged together in one single framework. Figure 1 depicts two examples of patterns that highlight the role of structural information.

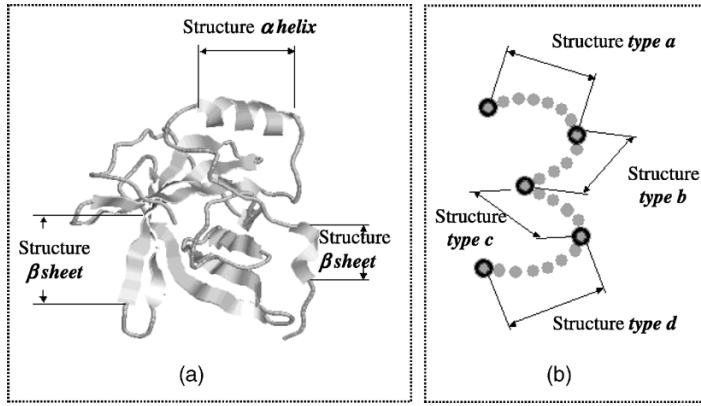

Figure 1. The presence of structural information in two different applications (a) 3D protein-folds assignment and (b) handwritten digits recognition.

3.3. The structural hidden markov model

We present in this section the mathematical expression of the structural hidden Markov model. We also give a definition of this model and the parameters involved.

If $O = (O_1, O_2, \cdots, O_s) = (o_{1_1} o_{1_2} \cdots o_{1_{r_1}}, o_{2_1} o_{2_2} \cdots o_{2_{r_2}}, \cdots, o_{s_1}, o_{s_2}, \cdots, o_{s_{r_s}})$, (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 $\sum_{i=1}^{i=s} r_i = T$) and $C = (C_1, C_2, \ldots, C_s)$, then the probability of a complex pattern O given a model λ can be written as:

$$P(O \mid \lambda) = \sum_{C} P(O, C \mid \lambda). \tag{7}$$

Therefore, we need to evaluate $P(O, C \mid \lambda)$. Since the model λ is implicitly present during the evaluation of this joint probability, therefore it is omitted. Thus we have:

$$P(O, C) = P(C, O) = P(C \mid O) \times P(O) \tag{8a}$$

$$= P(C_1 C_2 \ldots C_S \mid O_1 O_2 \ldots O_S) \times P(O) \tag{8b}$$

$$= P(C_S \ldots C_2 C_1 \mid O_S \ldots O_2 O_1) \times P(O) \tag{8c}$$

$$= P(C_S \mid C_{S-1} \ldots C_2 C_1 O_S \ldots O_1) \times P(C_{S-1} \ldots C_2 C_1 \mid O_S \ldots O_1) \times P(O). \tag{8d}$$

We assume that C_i depends only on O_i and C_{i-1} (as illustrated in Figure 2), and the structure probability distribution is a Markov chain of order 1. The reason behind this Markovian assumption comes from cognitive science. In fact, it is well-established that when we perform an object recognition task, our brain relies partly on local interactions between sub-patterns describing these objects (Fodor & Pylyshyn, 1998). Local interactions can also be expressed statistically by the means of Markovian fields using Gibbs distributions (Bartolucci & Besag, 2002; Ripley, 1996). However, as pointed out in

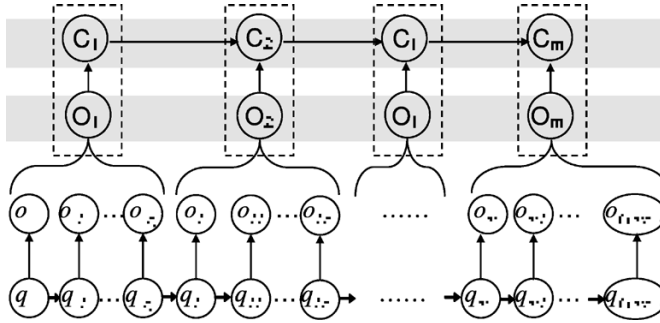

Figure 2. A graphical representation of a structural hidden Markov model.

the introduction, our approach considers exclusively sequential processes that remains within the context of HMM's. Finally, we can recursively approximate Eq. (8d) as:

$$P(O, C) \approx \prod_{i=1}^{S} P(C_i \mid O_i, C_{i-1}) \times P(O). \tag{9}$$

We now evaluate $P(C_i \mid O_i, C_{i-1})$ as follows:

$$P(C_i \mid O_i, C_{i-1}) = \frac{P(O_i C_{i-1} \mid C_i) P(C_i)}{P(O_i C_{i-1})}$$

$$= \frac{P(O_i \mid C_{i-1} C_i) P(C_{i-1} \mid C_i) P(C_i)}{P(O_i \mid C_{i-1}) P(C_{i-1})}.$$

Since O_i does not depends on C_{i-1}, we have:

$$P(C_i \mid O_i, C_{i-1}) = \frac{P(O_i \mid C_i) P(C_{i-1} \mid C_i) P(C_i)}{P(O_i) P(C_{i-1})}$$

$$= \frac{P(C_i \mid O_i) P(O_i) P(C_i \mid C_{i-1}) P(C_{i-1}) P(C_i)}{P(C_i) P(C_i) P(O_i) P(C_{i-1})}$$

$$= \frac{P(C_i \mid O_i) P(C_i \mid C_{i-1})}{P(C_i)}. \tag{10}$$

From Eqs. (9) and (10), we have:

$$P(O, C) \approx \prod_{i=1}^{S} \frac{P(C_i \mid O_i) P(C_i \mid C_{i-1})}{P(C_i)} \times P(O) \tag{11}$$

The organization (or syntax) of the symbols $o_i \in \Sigma$ is introduced mainly through the term $P(C_i \mid O_i)$ since the transition probability $P(C_i \mid C_{i-1})$ does not involve the

inter-relationship of the symbols O_i. Besides, the term $P(O)$ of equation 11 is viewed as a traditional HMM. Therefore, we can define a Structural HMM as follows:

Definition 3.3. *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,*
- $\mathcal{C}$ *is the posterior probability matrix of a structure given a sequence of observations,*
- $\mathcal{D}$ *is the structure transition probability matrix.*

A SHMM is characterized by the following elements:

- **N**, the number of hidden states in the model. We label the individual states as $1, 2, \ldots, N$, and denote the state at time t as q_t.
- **M**, the number of distinct observations O_i
- π, the initial state distribution, where $\pi_i = P(q_1 = i)$ and $1 \leq i \leq N$, $\Sigma_i \pi_i = 1$.
- $\mathcal{A}$, the state transition probability distribution matrix, $\mathcal{A} = \{a_{ij}\}$, where $a_{ij} = P(q_{t+1} = j \mid q_t = i)$ and $1 \leq i, j \leq N$, $\Sigma_j a_{ij} = 1$.
- $\mathcal{B}$, the state conditional probability matrix of the observations, $\mathcal{B} = \{b_j(k)\}$, in which $b_j(k) = P(o_k \mid q_j)$, $1 \leq k \leq M$ and $1 \leq j \leq N$, $\Sigma_k b_j(k) = 1$.
- **F**, the number of distinct local structures.
- $\mathcal{C}$ is the posterior probability matrix of a structure given its corresponding observation sequence, $\mathcal{C} = P(C_j \mid O_i) = c_i(j)$. For each particular input string O_i, we have: $\Sigma_j c_i(j) = 1$.
- $\mathcal{D}$, the structure transition probability matrix. $\mathcal{D} = \{d_{ij}\}$, where $d_{ij} = P(C_{t+1} = j \mid C_t = i)$, $\sum_j d_{ij} = 1$, $1 \leq i, j \leq F$.

Figure 2 depicts a representation of a structural hidden Markov model. We now define the problems that are involved in an SHMM.

3.4. Problems assigned to a structural HMM

There are four problems that are assigned to a SHMM:

(i) Probability evaluation,
(ii) Statistical decoding,
(iii) Structural decoding, and
(iv) Parameter estimation (or training).

- *Evaluation:* Given a model λ and an $O = (O_1, \cdots, O_s)$, an observation sequence, we evaluate how well does the model λ match O. This problem has been discussed in Section 3.
- *Statistical decoding:* In this problem, we attempt to find the best state sequence. This problem is similar to problem 2 of the traditional HMM and can be solved using Viterbi algorithm as well.

- *Local structure decoding:* This is the most important problem since we attempt to determine the "optimal local structures of the model". An example of an optimal sequence of structures is: <round, curved, straight, . . . , slanted, . . . , >. This sequence of structures helps describing physical or abstract objects. For example, autonomous robots based on this learning machine can be trained to recognize the components of a human face described as a sequence of shapes such as: <round (human head), vertical line in the middle of the face (nose), round (eyes), curved (mouth), . . . , >. Similarly, a customer's opinion of an automobile is composed of his/her opinion of the front, the side and the rear of this automobile. These partial opinions describe the whole external view of the automobile and impact significantly the customer's purchase decision.
- *Parameter estimation (Training):* In this problem, we try to optimize the model parameters $\lambda = [\pi, \mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}]$ to maximize $P(O \mid \lambda)$.

3.4.1. Probability evaluation. The evaluation problem in SHMM consists of evaluating the probability for the model $\lambda = [\pi, \mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}]$ to produce the sequence O. From Equation 11, this probability can be expressed as:

$$P(O \mid \lambda) = \sum_C P(O, C \mid \lambda) = \sum_C \left\{ \prod_{i=1}^{s} \frac{c_i(i) \times d_{i-1,i}}{P(C_i)} \right\}$$

$$\times \sum_q \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \ldots a_{q_{(T-1)} q_T} b_{q_T}(o_T). \tag{12}$$

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

3.4.3. Local structure decoding. The structural decoding problem consists of determining the optimal structure sequence $C^* =< C_1^*, C_2^*, \ldots, C_t^* >$ such that:

$$C^* = \arg\max_C P(O, C \mid \lambda).$$

We define:

$$\delta_t(i) = \max_C [P(O_1, O_2, \ldots, O_t, C_1, C_2, \ldots, C_t = i \mid \lambda)]$$

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

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) d_{ij} \right] c_{t+1}(j) \frac{P(O_{t+1})}{P(C_j)}. \tag{13}$$

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

3.4.4. Parameter estimation (training). In order to estimate $P(C_j \mid O_i)$, we used a non parametric classifier such as the k-nearest-neighbor method (Dude et al., 2001). After building the local structures, we denote K as the number of segments in its corresponding class of equivalence C_k. Then for any given class C_j, we estimated $c_i(j) = P(C_j|O_i)$ as:

$$c_i(j) \approx \frac{\text{the number of } O_i'\text{s nearest } K \text{ neighbors that are in } C_j}{K} \times 100\%. \qquad (14)$$

This *k-nearest-neighbor* posterior probability estimation technique obeys the exhaustivity and exclusivity constraint: $\sum_j c_i(j) = 1$. This estimation enables to built the entire matrix $\mathcal{C}$. Since the contour is represented by a sequence of local structures, then we used the Baum-Welch optimization technique to estimate the matrix $\mathcal{D}$. The other parameters, $\pi = \{\pi_i\}$, $\mathcal{A} = \{a_{ij}\}$, $\mathcal{B} = \{b_j(k)\}$, were estimated like in traditional HMM's (Rabiner & Juang, 1993). In the case of an unseen segment O_u that might be encountered during a testing phase, the probability $P(C_j \mid O_u)$ will be estimated by $P(C_j \mid O_i)\ \forall j$, where O_i is the "closest" segment to O_u in the training set.

4. Selected application: mining customer's preferences for automotive designs

We have applied the concept of Structural Hidden Markov Model in order to predict customers' preferences for automotive designs. This data mining aids automotive design engineers in predicting customers' perceptions on particular cars before they are put into making. This problem is very relevant to automotive companies since it provides them with a prior feedback about a particular automobile design.

4.1. Application description

When a customer wants to buy a car, she/he first observes the car's exterior features, and then she/he examines the car's interior features. The exterior features are very important in the sense that they influence customer's views of a car greatly. This is also the opinion of automotive companies which place a great emphasis on exterior appearance design. For example, Chrysler always makes real-size models of a car's exterior design, shows them to a lot of customers, and collects survey data of their opinions and feelings. This feedback is then sent to the design department. According to the information contained in the feedback, the exterior design engineers refine their designs. This process of "mining customers' opinions" and "refining designs" is repeated several times. At the last stage, the design engineers finalize the designs and put them into making. Traditionally, design engineers analyze the survey data manually. However, the analyzing process is time-consuming and tiresome. The purpose of this application is to build a computational method that helps engineers to improve their designs, speed up their job by releasing them from the tedious

Table 1. Respondent's feelings and their frequency during the survey.

Word	Frequency	Word	Frequency
Ordinary	11%	Ugly	1.5%
Beautiful	26%	Boxy	5%
Attractive	12%	Eww	0.5%
Amazing	5%	Boring	3.5%
Appealing	11.5%	Disgusting	1%
Nice	3.5%	Awesome	3%
Sleek	1%	Sporty	2%
Average	8.5%	Showy	0.5%
Lovely	4%	Other	0.5%

manual information processing, and eventually make cars that match the need of customers.

4.2. Automotive data collection and clustering

Based on what automotive companies generally do, we adopted the similar survey method as mentioned above. We collected 500 images of regular cars (no trucks or vans!) with their three exterior views (front, side and rear, i.e., 1500 images). A preprocessing phase of car images has been performed in order to let the people involved in the survey to focus on the exterior shapes of cars. This phase eliminates the influence of some features such as color, lamp shapes and tires. We removed the lamps and tires from the images, painted the body with white background color, and extracted the contours of the three views. Then we presented these contours to 300 university students. The students were asked to give their opinions on the three views of a car viewed separately. *Opinions are adjectives that express students' feelings of the car views at first sight. Every contour is assigned different opinions by different students*. 300 students would probably give as much as 300 different opinions to one contour. We adopted the "majority voting" method to obtain a unique opinion that is assigned to a contour. Thus we obtained 1500 adjectives (some of them are identical) clustered with synonymy using the online lexical reference system WordNet (Fellbaum, 1998). Each centroid of a cluster is called a *perception*. Each respondent's opinion (adjective) belongs to one and only one perception. Table 1 shows the adjectives that were collected during our survey and their frequencies of appearance in our sample.

Because it is very difficult to acquire a large automotive data set, we generated 10 artificial rules samples of size 500 using the *bootstrap* resampling technique (Efron, 1982). Then, we combined the 10 samples and obtained an artificial data set containing 10 times the data as our original data set, which means that we have 5000 strings of contours and 5000 conclusions for "front", "side" and "rear" views respectively. Finally, we divided this generated data set into 2 parts, 2/3 for training and 1/3 for testing.

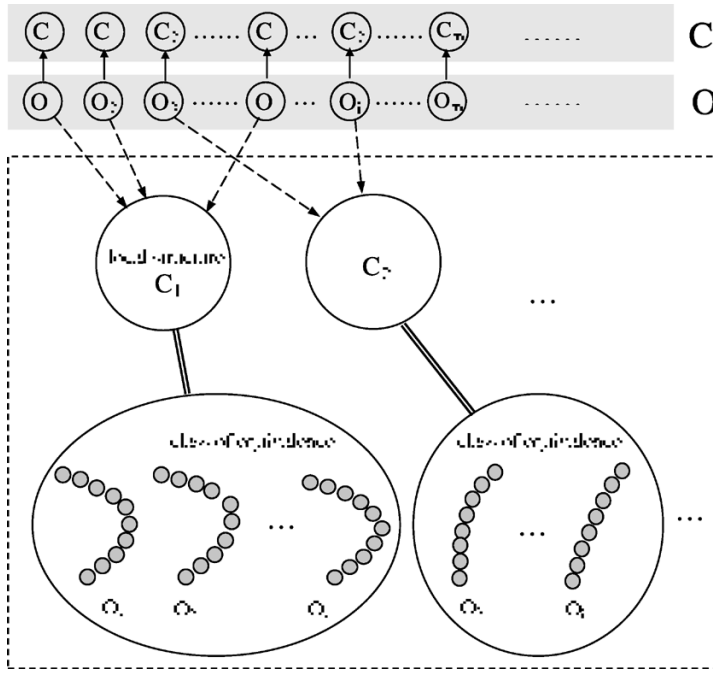

Figure 3. Set of classes of equivalence representing local structures. "Similar" shapes are put together in a same class.

4.3. Chain-code representation and local structures

The observations are the automotive contours represented by *chain-code* strings. chain-code is a well-known method for contour representation (Freeman, 1961). We used the standard implementation of the chain-code based on the eight directions (0–7). We extracted the contour of the three views of the car by following the contour in a clockwise manner and kept track of the directions as we went from one contour pixel to the next. However, the chain-code method has its own weaknesses: (i) it is not capable of performing a good corner detection or (ii) detecting sharp boundary changes, and thus not capable of capturing structural information. To solve this problem, we used a sequence of "local structures" to represent the car contours. The segmentation of each external contour was conducted using the shape convexity criterion. The sign of the second derivative is computed for each point located on the contour. Three rules were used to determine where to segment the contour:

1. If the second derivative changes its sign on a particular point of the contour, then this point is a boundary of the segment, otherwise we continue to extend the segment.
2. If the second derivative of a point is 0, then it is considered as an extension of the preceding segment.
3. An exception of rule 1 is: if the length of a segment is less than threshold "6", then we consider this segment as an extension of its preceding segment. If this segment is the first on the contour, then we concatenate it with its following segment. The threshold

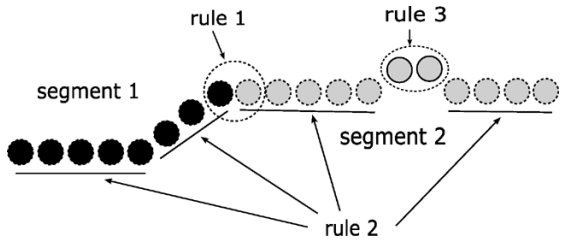

Figure 4. An example showing how two segments of a piece of a contour are formed according to the three rules we have set up.

> "6" is chosen because during our experiment, we have found out that the segments shorter than "6" are more often errors of image processing rather than strokes that characterize the car shapes.

Figure 4 shows how these three rules work. The black points form segment 1. The gray points form segment 2. The two gray points with solid boundaries are considered as part of segment 2.

This three-rules criterion is compatible with the distance introduced in Eq. (3). In other words, if we assign $+1$ to a symbol O_i that has a positive second derivative, -1 to a symbol with that has a negative second derivative and 0 to a symbol that has a zero-second derivative, we can see that the rule-based segmentation minimizes the error measure $\Xi(O, \Delta)$. However, the complexity in the optimal segmentation problem is reduced prominently in this application since we have prior information. This information is that the customer is sensitive to the shape (its convexity) of the external contour. Different views don't have the same number of segments. For example, model "BMW 330i" has 16 segments on its front view, 15 segments on its side view, and 17 segments on its rear view. The total number of segments we have obtained from the training data set is 1091. The maximum length of a segment is 209, and the minimum is 7.

4.4. Training and testing the SHMM

Once the optimal segmentation has been computed, we partitioned the set of segments into classes of equivalences. *As outlined above, each local structure is a class of equivalence containing similar strings of chain-code of a small segment of a car contour.* An automobile is now processed using four phases (i) car contour extraction. This contour is shown to the customers, (ii) segmentation of the car contour, (iii) partitioning of the segments into local structures, and (iv) chain-code representation of each local structure. Figure 5 depicts the four-phase processing of the automobile contour.

The partitioning of the segments into local structures was conducted through the following two phases:

- Compute the edit distance between each pair of chain-code strings of segments.
- Build the classes of equivalence of segments with respect to the threshold ε.

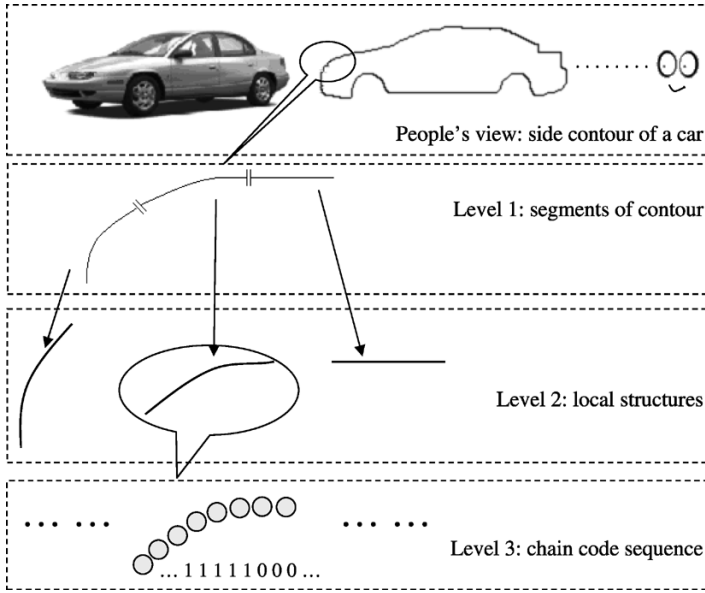

Figure 5. Four-phases processing of a car contour.

We have chosen $\varepsilon = 23$ and obtained 30 classes of equivalence (30 local structures). We used the estimation technique described in Section 3.4.4 to compute π, $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$.

Once the SHMM assigned to a car view which is represented by the entire sequence O is built, we used the testing data set to evaluate the model accuracy. To achieve this goal, we first determined the local structures assigned to the testing set. We have selected 5 perception categories in this application which are the 5 clusters of adjectives obtained using the lexical database WordNet. Therefore five models λ_i have been generated, and each model is built in order to learn one perception category. The number of samples in each category is the cardinal of each cluster described in Section 4.2. Currently the categories we have obtained and their numbers of samples are: ugly-165, ordinary-323, nice-60, attractive-487, beautiful-465. The best model λ^* that is assigned to the predicted perception for each side is computed via the following equation:

$$\lambda^* = arg \max_{\lambda_i} P(O \mid \lambda_i). \tag{15}$$

If the predicted model is λ_p and the true model obtained from survey is λ_t, then our precision is defined as:

$$Precision = \frac{\sum \delta(\lambda_p - \lambda_t)}{\mid input\ patterns \mid} \tag{16}$$

where $\delta(x - a)$ is the Kronecker symbol which is "1" if x = a, and "0" otherwise, the denominator $\mid input\ patterns \mid$ represents the total number of patterns. In our application, the numerator is the number of correctly classified contours, and the

Table 2. Comparison of the predicted perceptions with true results on car views.

	Sequence of survey perceptions			Sequence of predicted perceptions		
Car makes	Front	Side	Rear	Front	Side	Rear
Audi a430	Nice	Attractive	Attractive	Nice	Attractive	Nice
Audi a8l	Ordinary	Attractive	Nice	Attractive	Attractive	Nice
Audi s6a	Beautiful	Attractive	Attractive	Beautiful	Attractive	Nice
BMW 525i wagon	Beautiful	Attractive	Nice	Ugly	Attractive	Nice
BMW convertible	Ordinary	Ordinary	Ordinary	Ordinary	Ordinary	Nice
BMW z3 roaster	Ugly	Ugly	Ugly	Ugly	Nice	Ugly
Cadillac ext	Ordinary	Ordinary	Ordinary	Attractive	Ordinary	Ordinary
Chrysler concorde	Beautiful	Attractive	Attractive	Beautiful	Attractive	Nice
Honda civic hybrid	Attractive	Beautiful	Attractive	Attractive	Attractive	Attractive
Volvov70	Attractive	Ordinary	Beautiful	Beautiful	Ordinary	Beautiful

Table 3. Performances of HMM and SHMM classifiers using 5-fold cross-validation.

5-fold cross validation samples	HMM	SHMM
1	73%	81%
2	78%	80%
3	66%	85%
4	70%	79%
5	73%	82%
Average	72%	81.4%

denominator is the number of all the three exterior view contours in the testing data set. The denominator number is $5000 \times 3 = 15000$. Table 2 shows the perceptions obtained from survey along with the predicted perceptions of 10 car makes. If one of the three categories in the predicted sequences is different from the corresponding category in the survey perception, then this is counted as an error. However, the sequence of the survey perceptions <attractive, beautiful, attractive> is equal to the sequence of the predicted perceptions <attractive, attractive, attractive> since "beautiful" and "attractive" belong to the same WordNet synset (set of synonyms).

We have compared the SHMM approach with the traditional Hidden Markov Model (HMM) classification technique. The training of both the SHMM and HMM was coded using The software package MATLAB. The accuracy computation in the case of the HMM is based on the comparison between the predicted category and the true category (from survey) for each view *separately*. However, the accuracy in the case of the SHMM is based on the comparison of the predicted and the true sequence assigned to the three views *at once*. The design engineers are interested in discovering the flaws from the three views separately rather than from the whole car. For example, in the case of HMM, if the front view perception of *Cadillac ext* is predicted as "attractive" while the true category "ordinary" then we have an error of classification. Thus, the HMM was applied

to each view of the car separately. Each view contributes to the prediction result without interfering with other views.

In order to measure the power of generalization of the SHMM's classifier, we used the *m-fold cross-validation* estimation technique. We divided the images of the 500 cars into 5 sets, each of which contains images of 100 cars. Then we selected one set for training and the other 4 sets for testing. We repeated this procedure 5 times with each time selecting a different set for training. Table 3 shows the accuracy of each round and the average accuracy.

5. Conclusion and future work

We have presented in this paper a novel modeling technique that merges syntactical and statistical information into a single probabilistic framework. Our approach relates visible observation sequences through their contribution to a same local structure built from an equivalence relation. Doing so, SHMM's bypass the state conditional independence assumption inherent to traditional hidden Markov modeling. SHMM's extend traditional HMM's by incorporating the structural dimension within the statistical design. The SHMM concept represents a preliminary attempt to unfold structural information embedded in complex patterns (Bouchaffra & Tan, 2004). The automotive application shows that SHMM concept is promising since it has significantly outperformed the traditional hidden Markov model classifier. However, this is an undergoing research, more data need to be collected, and comparisons with other classifiers are necessary in order to measure the global contribution of SHMM's. Our future work is to apply SHMM's to other areas such as molecular biology, complex chemical compounds recognition and natural language processing where structure is prominent.

Note

1. In other words, it is possible to decrease the resolution level of a complex pattern.

Acknowledgment

The authors would like to thank Chrysler automotive company for their comments during the data collection period needed for experiment. We are also grateful to all students from Oakland University who spent a part of their time to answer our questions during this survey.

References

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

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

Baum, L. and Petrie, T. 1966. Statistical inference for probabilistic functions of finite state markov chain. Ann. Math. Stat., 37:1554–1563.

Bellman, R. 1961. On the approximation of curves by line segments using dynamic programming. Communication of the ACM, (4(6)).

Bouchaffra, D., Govindaraju, V., and Srihari, S. 1999. Postprocessing of recognized strings using nonstationary Markovian models. IEEE Transactions: Pattern Analysis and Machine Intelligence, 21(10).

Bouchaffra, D. and Tan, J. 2004. The concept of structural hidden markov models: Application to mining customers' preferences for automotive designs. In 17th international conference on pattern recognition, (icpr). Cambridge, United Kingdom.

Cai, J. and Liu, Z. 1999. Integration of structural and statistical information for unconstrained handwritten numeral recognition. EEE Transactions: Pattern Analysis and Machine Intelligence, 21(3).

Churchill, G. 1989. Stochastic models for heterogeneous DNA sequences. Bulletin of Mathematical Biology, (51):79–94.

Duda, R., Hart, P., and Stork, D. 2001. Pattern classification, New York: Wiley.

Eddy, S. 1998. Profile Hidden Markov models. Bioinformatics, 14(9):755–763.

Efron, B. 1982. In the jackknife, the bootstrap and other resampling plans. SIAM.

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

Fellbaum, C. 1998. Wordnet: an electronic lexical database. Bradford Book.

Fodor, J. and Pylyshyn, Z.W. 1988. Connectionism and cognitive architecture: a critical analysis. Cognition, (28):3–71.

Freeman, H. 1961. On the encoding of arbitrary geometric configurations. IRE Trans. Electron. Comput., EC-10, 260–268.

Gales, J. 2000. Cluster adaptive training of hidden markov models. IEEE Transactions on Speech and Audio Processing, 8(4).

Geman, S., Bienenstock, E., Geman, S., and Potter, D. 2004. Compositionality, MDL priors, and object recognition (Tech. Rep.). Division of Applied Mathematics, Brown University.

Gemignani, M. 1990. Elementary topology. second edition. New York: Dover Publications, Inc.

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

Kim, D. and Bang, S. 2000. A handwritten numeral character classification using tolerant rough set. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(9).

Krogh, A., Brown, M., Mian, I., Sjolander, K., and Haussler, D. 1994. Hidden markov models in computational biology: Applications to Protein Modeling. J. Mol. Biol. 235:1501–1531.

Li, J., Najmi, A., and Gray, R. 2000 Feb. Image classification by a two-dimensional hidden markov model. IEEE Transactions on Signal Processing, 482(2):517.

Rabiner, L. and Juang, B. 1993. Fundamentals of speech recognition. Prentice Hall.

Ripley, B. 1996. Pattern recognition and neural networks. Cambridge University Press.

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

Zhou, J. and Chen, D. 2002 November. The subsystem of the fuzzed rough sets based on equivalence class. In Proceedings of the First International Conference on Machine Learning and Cybernetics. Beijing.

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