

# INTRODUCTION TO STRUCTURAL HMM AND ITS APPLICATION IN PATTERN CLASSIFICATION

**D. BOUCHAFFRA (Senior Member IEEE), J. TAN**

Department of Computer Science & Engineering

131 Dodge Hall, Oakland University

Rochester, MI, 48309, USA

Phone: (248) 370-2242

{bouchaff, jtan}@oakland.edu

## **ABSTRACT**

The classification of complex patterns remains a challenging problem in the pattern recognition community. In this paper we propose a classification paradigm that merges statistics with syntax in a seamless way. This novel approach called “Structural HMM”(SHMM) extends traditional HMM by incorporating syntax with statistics within a single probabilistic framework. We have applied SHMM in order to data mine customers’ preferences for automotive designs. The results reported shows that SHMM’s outperform traditional hidden Markov model classifiers.

## **INTRODUCTION**

Human perception of objects contains roughly two types of relationships: the *classification* relationship, by which a human *generalizes* experience, and the *componential* relationship, by which we *organize* the whole made up of many parts that seems to be an inherent quality of all things. Therefore, statistics and structure are always driving humans in a decision problem in pattern recognition (PR). Unfortunately, for most practical problems, this *purely statistical* approach is not feasible. 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 between components quite often fails.* Several approaches have been attempted in order to capture the structural relationship between observations. They are “syntactical pattern recognition” [Fu1982], “Bayesian belief networks” (BBN’s) [Heckerman1995], and traditional “hidden Markov model” paradigm [Rabiner1993]. However, they all have the weak points that they either consider syntactic or statistical relationships, but not both at the same time. The thrust in this paper is to combine statistics with syntax in a seamless way so that we will be capable to optimally represent complex patterns. We propose a novel paradigm that we called *structural hidden Markov model* (SHMM) that merges statistical and syntactical data together in a single probabilistic framework. We extract structural information from sequences of observations viewed as input strings to be derived by a grammar. The structural information are conclusions (or variables of a grammar) that accept the input strings (sequences of observations). Because our world is full of complex patterns, we believe that the concept of SHMM opens a new door for understanding and unfolding these structures. In this research statistics and syntax as complementary, they do not compete in the pattern representation phase.

## **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.

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

$$P(O | \lambda) = \sum_{all\ q} P(O, q | \lambda), \text{ and } P(O, q | \lambda) = P(O | q, \lambda) \times P(q | \lambda).$$

Using state conditional independence, we obtain:  $P(O | q, \lambda) = \prod_{t=1}^T P(o_t | q_t, \lambda)$ .

However, there are several scenarios where the conditional independence assumption doesn't hold. For example, while standard HMM's perform well in recognizing amino acids and consequent construction of proteins from the first level structure of DNA sequences [Krogh1994], they are inadequate for predicting the secondary structure of a protein. Therefore, there is a need to balance the loss incurred by this state conditional independence assumption. Our idea is to consider the sequence of visible observations as input to a grammar. The sequence of observations is viewed as an input string to a grammar, and a conclusion (or a head) is computed given this string.

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  $s$  subsequences  $O_i = (o_{i_1} o_{i_2} \dots o_{i_{r_i}})$  ( $1 \leq i \leq s$ ). The observations in a subsequence are related in the sense that they represent evidences  $o_i$  that contribute to the production of a conclusion  $C_j = f_G(o_{i_1}, o_{i_2}, \dots, o_{i_{r_i}})$ , ( $1 \leq j \leq L$ ), where  $f_G$  is the derivation in the grammar that accepts the input string  $o_{i_1}, o_{i_2}, \dots, o_{i_{r_i}}$  and  $L$  is the number of conclusions. The whole sequence of observations can be written directly as:

$$O = (o_{1_1} \dots o_{1_{r_1}}, C_1, o_{2_1} \dots o_{2_{r_2}}, C_2, \dots, o_{s_1} \dots o_{s_{r_s}}, C_s) = (O_1 C_1 O_2 C_2 \dots O_s C_s).$$

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. We consider the overall observation sequence  $O$  as a sequence of  $O_i$  with their corresponding conclusions  $C_j$ .

A sequence of visible observations is the input string presented to the grammar, the organization of the symbols in this sequence  $O_i$  produces a conclusion  $C_j$  with a certain probability  $P(C_j | O_i)$ . The higher the complexity of the pattern, the higher the number of conclusions needed to describe the structure of this pattern. Each conclusion  $C_j$  interacts with another conclusion  $C_i$  with some transition probability values  $P(C_j | C_i)$ . Therefore, we can define a Structural HMM as:

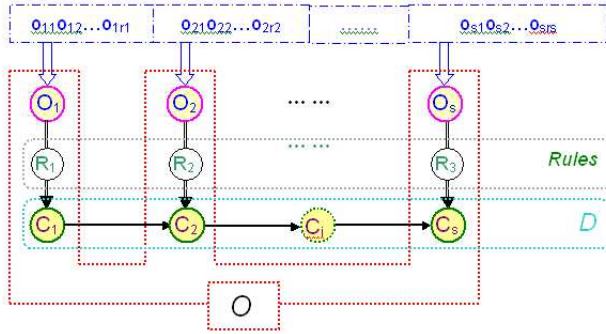
**Definition 01** A structural hidden Markov model is a quintuple  $\lambda = (\pi, \mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D})$ , where: (i),  $\pi$  is the initial state probability vector; (ii),  $\mathcal{A}$  is the state transition probability matrix; (iii),  $\mathcal{B}$  is the state conditional probability matrix of the visible observations; (iv),  $\mathcal{C}$  is the posterior probability matrix of a conclusion given a sequence of observations; (v),  $\mathcal{D}$  is the conclusion transition probability matrix.

A SHMM is characterized by the following elements:

- **N**, the number of states in the model. We label the individual states as  $1, 2, \dots, N$ , and denote the state at time  $t$  as  $q_t$ .
- **M**, the number of distinct observations in one state. We use a symbol to represent each observation  $o_i$ , and the set of symbols is denoted as  $V = \{v_1, v_2, \dots, v_M\}$ .
- $\pi$ , the initial state distribution,  $\pi = \{\pi_i\}$ , where  $\pi_i = P(q_1 = i)$  and  $1 \leq i \leq N$ .
- $\mathcal{A}$ , the state transition probability distribution matrix,  $A = \{a_{ij}\}$ , where  $a_{ij} = P(q_{t+1} = j | q_t = i)$  and  $1 \leq i, j \leq N$ ,  $\sum_j a_{ij} = 1$ .
- $\mathcal{B}$ , the state conditional probability matrix of observations,  $B = \{b_j(k) = P(o_k | q_j)\}$ .
- **F**, the number of distinct conclusions. We use a symbol to represent each conclusion, and the set of conclusions is denoted as  $U = \{u_1, u_2, \dots, u_F\}$ .

- $\mathcal{C}$  is the posterior probability matrix of a conclusion given its corresponding observation sequence,  $\mathcal{C} = P(C_j | O_i) = c_i(j)$ . For each particular input string  $O_i$ , we have:  $\sum_{All j} c_i(j) = 1$ . A particular application requires a particular grammar  $G_h$  in which a conclusion  $C_i$  is assigned to  $O_i$  via a rule  $R_k$  which is written as:  
 $G_h : C_i \xleftarrow{R_k} (o_{i1}, o_{i2}, \dots, o_{it}), k$  is the number of rules in the grammar  $G_h$ . Since we are using a stochastic context free grammar, there is only one conclusion (the most likely) that is assigned to a subsequence of observations.
- $\mathcal{D}$ , the conclusion transition probability matrix.  
 $D = \{d_{ij}\}$ , where  $d_{ij} = P(C_{t+1} = j | C_t = i), \sum_j d_{ij} = 1, 1 \leq i, j \leq F$ .

Figure 1 depicts a representation of a structural hidden Markov model



**Fig. 1.** A graphical representation of a structural hidden Markov model.

### 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.

### PROBABILITY EVALUATION

The evaluation problem in SHMM consists of computing:

$$P(O | \lambda) = P(O_1 C_1, O_2 C_2, \dots, O_s C_s | \lambda)$$

$$= \prod_{i=1}^s [P(C_i | O_i C_{i-1} O_{i-1} \dots C_2 O_2 C_1 O_1 \lambda) P(O_i | C_{i-1} O_{i-1} \dots C_2 O_2 C_1 O_1 \lambda)].$$

We assume that the conclusion  $C_i$  depends only on the observation sequence  $O_i$  and the preceding conclusion  $C_{i-1}$ . Therefore, we have:

$$P(O | \lambda) = \prod_{i=1}^s [P(C_i | O_i C_{i-1} \lambda) \times P(O_i | C_{i-1} O_{i-1} \dots C_2 O_2 C_1 O_1 \lambda)].$$

Because  $O_i$  depends only on  $C_i$ , we obtain:  $P(O | \lambda) = \prod_{i=1}^s [P(C_i | O_i, C_{i-1}, \lambda) \times P(O_i | \lambda)]$ .

Now we have to focus on the term  $P(C_i | O_i C_{i-1})$  which is the most challenging term for estimation. However, we know that:  $P(C_i | O_i C_{i-1}) = \frac{P(O_i | C_i C_{i-1}) \times P(C_i | C_{i-1})}{P(O_i | C_{i-1})}$ . Given the fact that the conclusion  $C_i$  is the only conclusion that might produce  $O_i$ , therefore we can write this conditional independence as:  $P(O_i | C_i C_{i-1}) = P(O_i | C_i)$ . Using Bayes'

rule, we obtain:  $P(O | \lambda) = \prod_{i=1}^s \frac{[P(C_i | O_i \lambda) P(C_i | C_{i-1}, \lambda) \times P(O_i | \lambda)]}{P(C_i)}$ .

Finally, this provides:

$$P(O|\lambda) = \prod_{i=1}^s \frac{c_i(i)}{P(C_i)} \times \sum_{q_1 \dots q_T} [\pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} \dots a_{q_{T-1} q_T} b_{q_T}(o_T) d_{C_1 C_2} \dots d_{C_{s-1} C_s}].$$

## STATISTICAL DECODING

The statistical decoding problem consists of determining the optimal state sequence  $q^* = \underset{q}{argmax} (P(O, q | \lambda))$  that best “explains” the sequence of observations. It can be computed using Viterbi algorithm as in traditional HMM’s.

## STRUCTURAL DECODING

The structural decoding problem consists of determining the optimal rule conclusion sequence

$C^* = \langle C_1^* C_2^* \dots C_t^* \rangle$  such that:  $C^* = \underset{C}{argmax} (P(O, C | \lambda))$ .

We define:  $\delta_t(i) = \underset{C}{max} P(O, C_1 C_2 \dots C_t = i | \lambda)$ , that is,  $\delta_t(i)$  is the highest probability along a single path, at time  $t$ , which accounts for the first  $t$  observations and ends in rule conclusion  $i$ . Then, we estimate the following by induction:

$$\delta_{t+1}(j) = \left[ \underset{i}{max} \delta_t(i) d_{ik} \right] b_j(o_{t+1}).$$

Similarly, this can be computed using Viterbi algorithm. However, we estimate  $\delta$  in each step *through conclusion transition probability matrix instead of state transition probability matrix*. This optimal sequence of conclusions describes the structural pattern piecewise.

## PARAMETER ESTIMATION

The re-estimation phase of the parameters  $\{\pi_i\}$ ,  $\{a_{ij}\}$ ,  $\{b_j(k)\}$  and  $\{d_{ij}\}$  is conducted as in traditional HMM’s [Rabiner1993], using the Baum-Welch optimization technique.

However, *the most difficult problem* is the estimation of  $P(C_j | O_i)$ . We need to construct a grammar  $G$  that enables the estimation of this term. Usually, it is the user who constructs an appropriate grammar based on personal knowledge and experience regarding a particular application. To construct such a grammar, a set of primitives is selected depending on the type of data involved in the application. *It is very important that the primitives provide a reasonable description of the patterns and the structural relations*. To derive a conclusion, we use “stochastic context-free grammar” [Jurafsky1995]. Similar to the decoding process, a Viterbi algorithm is used to find the best derivation.

Assume an incoming  $x = O_i$  derives  $S = C_j$ , this can be illustrated as:

$$S = (C_j) \xrightarrow{R_1} \alpha_1 \xrightarrow{R_2} \alpha_2 \dots \xrightarrow{R_t} \alpha_t = x (= O_i)$$

where  $R_i$  are the production rules used in the productions,  $\alpha_i$  ( $1 \leq i \leq t$ ) is the intermediate level of derivation and  $\alpha_{i+1}$  is obtained from  $\alpha_i$  using a rule that is related with the conclusion  $C_j$  in  $\alpha_i$ . The production is performed with the probability  $P(R_i)$  which is the probability of the production rule  $R_i$  being used in the production. Thus, in our model,

$$P(C_j | O_i) \text{ can be estimated as: } P(C_j | O_i) = P(R_1) \prod_{i=1}^{n-1} P(R_{i+1} | R_i).$$

Suppose  $C_j$  is the starting symbol of our grammar,  $o_i$ ’s are the terminal symbols, and we have only one production rule  $R_1$  defined as  $C_j \leftarrow O_i (= o_{i_1} o_{i_2} \dots o_{i_r})$ . Then we have the production procedure:  $S \rightarrow C_j \xrightarrow{R_1} O_i \rightarrow o_{i_1} o_{i_2} \dots o_{i_r}$ .

Therefore, we have:  $P(C_j | O_i) = P(R_1) = P(C_j \leftarrow o_{i_1} o_{i_2} \dots o_{i_r})$ .

## APPLICATIONS: DATA MINING CUSTOMERS' PREFERENCES FOR AUTOMOTIVE DESIGNS

We have applied the concept of Structural Hidden Markov Model in order to mine customers' preferences for automotive designs. This data mining aids automotive design engineers in predicting customers' perceptions on particular car make's exterior contour before these cars are put into making. 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 manual information processing, and eventually make cars that match the need of customers.

### DATA COLLECTION

We collected 500 images of regular cars with their three exterior views (front, side and rear, i.e., 1500 images). A pre-processing phase of car images was performed in order to remove such influence of some features as color, lamp shapes, or tires. We then extracted the contours of the three views and presented them to 300 university students and ask them to give their opinions (adjectives expressing students' feelings of the car) on the three views separately. 300 students would probably give as much as 300 different opinions to one contour. We adopted the "simple majority voting" method to obtain a unique opinion that is assigned to a contour. Thus we obtained 1500 adjectives (some are identical) clustered with synonymy using the online lexical database WordNet [Fellbaum1998]. Each centroid of a cluster is called a *perception* which is a conclusion in the SHMM modeling. Conclusions are the customer's perceptions regarding the automotive contours and observations are standard 8-directions *chain code* string representing the contours. Therefore, we extracted the contour of "front (f)" and represented it as  $O_f = (o_1 o_2 \dots o_{r_f})$ , where  $o_i$ 's is the chain code string. The customer's opinion assigned to this view is represented by  $C_f$ , where  $C_f$  is the conclusion assigned to rule  $R_f$  that defines how the opinion of this view is obtained from the chain code description of its contour. We used the stochastic grammar to estimate the matrix  $C$ . The grammar  $G_c$  we used is defined as follows:

- (1). The set of non-terminals  $V_N^c$  is {contour perceptions}.
- (2). The set of terminals  $V_T^c$  is {0-8}, which is the chain code 8 directions.
- (3). The set of production rules  $R^c$  is {"Perception of car contour"  $\leftarrow$  "chain code description of that contour"}.
- (4).  $Q^c$  is set of probabilities of all rules in set  $P^c$ .

Thus, we defined a rule as: "*front is ordinary*"  $\leftarrow$  "0200 . . . 22220000".

### TRAINING SHMM

Training a Structural HMM is an iterative process that seeks to maximize the probability that the SHMM accounts for the example sequences. To re-estimate  $\{a_{ij}\}$ ,  $\{b_j(k)\}$  and  $\{d_{ij}\}$ , we applied the Baum-Welch re-estimation algorithm. Table 1 show the transition probability matrix of 5 major conclusions. From the training data set, we generate 1000 rules and each rule is assigned a certain probability.

$\frac{to}{from}$	ugly	ordinary	nice	attractive	beautiful
ugly	.2100	.1053	.2105	.4211	.0526
ordinary	.0906	.2727	.2273	.2273	.1818
nice	.2603	.1304	.1304	.3043	.1739
attractive	.1532	.1795	.2564	.3846	.0256
beautiful	.0098	.3989	.1002	.4000	.0000

Table 1. Transition probability matrix  $\mathcal{D}$  of 5 major conclusions estimated from the data set.

## TESTING AND RESULTS

Once the SHMM for this application is built, we used the testing data set to evaluate the accuracy. Observation sequences are fed into the SHMM, then conclusions are predicted. If our predicted conclusion (or category) is  $C_p$  and the true conclusion obtained from survey is  $C_t$ , then our precision is defined as:

$$Precision = \sum \delta(C_p - C_t) / |input\ patterns|$$

where  $\delta(x-a)$  is the Kronecker symbol which is "1" if  $x=a$ , and "0" otherwise, the denominator  $|input\ patterns|$  represents the total number of patterns. We have compared the SHMM approach with the traditional Hidden Markov Model (HMM) classification technique. The training of both the SHMM and HMM were coded using MATLAB. Preliminary performance results are depicted in Table 2.

Precision (%) Sample Size	HMM	SHMM
100 cars	65.6	70.1
450 cars	72.5	83.4
500 cars	78.5	83.2

Table 2. Performances comparison between the HMM and the SHMM classifiers.

## CONCLUSION AND FUTURE WORK

We have presented in this paper a novel modeling technique that merges syntactical and statistical information in a single probabilistic framework. Our approach relates visible observations through their contribution to a same conclusion (or consequence) of a syntactic rule. The SHMM concept represents a preliminary fusion between statistics and syntax. The automotive application shows that SHMM concept is promising since it has outperformed the traditional hidden Markov model classifier. However, this is an ongoing research, more data need to be collected, and comparisons with other classifiers are necessary in order to measure the real contribution of SHMM's. Our future work is twofold: (i), to use some other techniques to represent the visible observation sequences rather than chain codes; (ii), to apply SHMM's to other areas such as proteins identification and natural language processing where structure is prominent.

## References

- [Fellbaum1998] C. Fellbaum, "WordNet: an Electronic Lexical Database", Published by Bradford Book, 1998.
- [Fu1982] K.S. Fu, "Syntactic Pattern Recognition and Applications", Prentice-Hall, Englewood Cliffs, N.J., 1982.
- [Heckerman1995] D. Heckerman. "A Tutorial on Learning with Bayesian Networks", Technical Report MSR-TR-95-06, Microsoft Research, Redmond, Washington, 1995.
- [Jurafsky1995] D. Jurafsky, C. Wooters, J. Segal, A. Stolcke, E. Fosler, G. Tajchman and N. Morgan, "Using a Stochastic Context-Free Grammar as a Language Model for Speech Recognition", In Proc. ICASSP'95 (pp. 189-192), 1995.
- [Krogh1994] Krogh, A., M. Brown, I.S. Mian, K. Sjolander, and D. Haussler. "Hidden Markov Models in Computational Biology: Applications to Protein Modeling", J. Mol. Biol., Vol. 235, pp.1501-1531, 1994.
- [Rabiner1993] L. Rabiner and B.H. Juang, "Fundamentals of Speech Recognition", Prentice Hall, 1993.