Hidden Markov Models vs Syntactic Modeling in Object Recognition[*]

Ana L. N. Fred
IT
Instituto Superior Técnico
Lisboa, Portugal

Jorge S. Marques
INESC
Instituto Superior Técnico
Lisboa, Portugal

Pedro M. Jorge
INESC
Instituto Superior Técnico
Lisboa, Portugal

Abstract

This paper addresses the problem of object recognition based on contour descriptions. Two approaches, namely hidden Markov models (HMM) and syntactic modeling based on stochastic finite-state grammars (SFSG), are analyzed and applied to the classification of hardware tools.

1 Introduction

The recognition of 2D objects requires two steps: the extraction of image features and their classification. Several methods exist to perform each of these operations. A large attention has been focused on the analysis of the object boundary using global and local representations. Global representations (e.g., Fourier descriptors, invariant moments) are easy to manipulate and classify. However, its computation depends on the entire shape and, therefore, local deformations and occlusions often produce strong deviations in all the features. Local features (e.g., differential chain codes) are more robust against deformations and occlusions but they require sophisticated classification techniques.

Object recognition based on string contour descriptions are typically performed using string matching techniques [1]; preliminary results on shape clustering using stochastic grammars are reported in [3]. In this paper, a model-based approach is addressed. Two modeling / classification strategies, which are usually studied separately, are considered: hidden Markov models (HMM) [9] and syntactic models [11]. Hidden Markov models have been intensively used in the context of speech recognition as a tool to represent the dynamic properties of the speech features (e.g., spectral coefficients) during an utterance [9, 6]. The application of HMMs in object and shape recognition is less popular. However, they have been used with success in shape and cursive handwritten recognition (e.g., see [5, 10]). Typical applications of syntactic methods have been in the areas of pattern recognition [5], speech recognition [9], natural languages and programming languages. In this paper, HMMs and syntactic models based on stochastic finite-state grammars are applied and compared in object recognition based on contour descriptions extracted from images of hardware tools.

2 Hidden Markov Models and Stochastic Finite State Grammars

HMMs and SFSG share many common characteristics, being instances of a more general class of models designated by *stochastic finite state networks* [9]. They both generate an internal (non-observable) sequence of symbols (states) and a sequence of external (observable) symbols, using probabilistic rules. Based on the theory of stochastic processes, they however assume different formalisms and distinct mechanisms of inference.

Formally, a HMM is characterized by: $H = (Q, \Sigma, A, B, \pi)$, where Q is a finite set of states q_i ; Σ is a set of possible observation symbols; A is a matrix with the state transition probabilities ($A : Q \times Q \rightarrow [0, 1]$); B represents the observation symbol probability distribution in a state; and π is the initial state distribution. Given a HMM, the probability of the observation sequence $x = x_1 x_2 \ldots x_n (x_i \in \Sigma)$ is given by

$$\begin{aligned} p(x|H) = \; & \textstyle\sum_{q_1 \ldots q_n} \pi(q_1) B(q_1, x_1) A(q_1, q_2) B(q_2, x_2) \\ & \ldots A(q_{n-1}, q_n) B(q_n, x_n) \end{aligned}$$

In order to define a HMM one needs to know its structure (number of states and allowed state transitions) and parameter values. The structure of HMMs is chosen *a priori*; tuning to the data is typically performed on a trial basis, eventually conditioned by existent knowledge. Two types of architectures are usually considered: 1-totally connected models, where all state transitions are allowed; 2- left-to-right models, where temporal constraints are taken into account by defining unique initial and final states, and states are ordered in such a way that if $q_i < q_j$ then $A(q_j, q_i) = 0$. Totally connected models are general but they may depend on a large number of parameters which are hard to estimate. Therefore they are not necessarily the best solution. Parameter estimation is performed in a non optimal way. Two popular methods

[*] This work has been partly supported by JNICT projects PECS/C/SAU/212/95, PBIC/TPR/2562/95 and PRAXIS 2/2.1/TIT/1580/95.

are the Baum-Welch re-estimation algorithm and the Viterbi estimation algorithm [9]. The Baum-Welch algorithm tries to optimize the likelihood function of the training set applying an EM approach. The algorithm converges to local maxima of the likelihood function which depend on the algorithm initialization. The Viterbi estimation is simpler but sub-optimal; the highest probability state sequence associated with each training sequence is computed. The model parameters are then estimated by computing the relative frequencies of transitions and output symbols, assuming that the training sequences were generated by the estimated state sequence. The output of the Viterbi algorithm also depend on the initial estimates of the model parameters.

A SFSG [11] is defined by $G = (N, \Sigma, P, \sigma)$, where N a finite set of non-terminal symbols; Σ is the set of observable symbols (vocabulary); P is a finite set of productions of the form $p_{ij} : A \rightarrow aB$ or $p_{ij} : A \rightarrow a$, $A, B \in N$, $a \in \Sigma$, p_{ij} being the rule probability; $\sigma \in N$ is the start symbol. Given a SFSG, G, the probability of $x = x_1 x_2 \ldots x_n$ being generated by G is given by

$$p(x|G) = \sum_{j=1}^{k} p_j| \left(\sigma \overset{p_j}{\underset{*}{\Rightarrow}} x \right),$$

where j denotes one of the k possible derivations and $p_j| \left(\sigma \overset{p_j}{\underset{*}{\Rightarrow}} x \right)$ represents the probability of the jth derivation of x from the start symbol σ according to the rules in P. A SFSG can be represented by a graph where each node is associated with a non-terminal symbol, with an initial node labeled σ, a final absorbing node, and the arcs link nodes associated by rules in P. The number and structure of the rules are automatically inferred from the training data set by grammatical inference methods [4, 8]. Most of these procedures use heuristic information to detect common structures in string patterns, modeled in terms of rules. Maximum likelihood techniques applied to the estimation of rule probabilities of non-ambiguous grammars (only one derivation is possible for each string in the language) lead to the computation of relative frequencies of rules usage in the derivation of the training set. According to the method of stochastic presentation, the probabilistic nature of the patterns are represented by the relative frequency of occurrence of samples in the training set. When ambiguous grammars are considered, non-optimal estimates based on single derivation per sample are common practice.

It can be shown [2] that the above formalism of HMM can be put in an equivalent form with observation probability distributions in the transitions, designated by HMMT. While this general model is not equivalent to a SFSG, HMMs with observation probability distributions in the transitions and with final state are equivalent to SFSGs. The two approaches therefore differ mainly on the way of definition of the network structure, the first assuming an *a priori* structure and the latter inferring it from the data, based on some heuristic information underlying the grammatical inference procedure. Estimates of rules probabilities based on the method of stochastic presentation for SFSGs are comparable to the estimates provided by the Viterbi algorithm for the HMMs.

3 Object Recognition

HMMs and SFSGs were used in 2D shape recognition of hardware tools. The following presents the methodology and experimental setup.

3.1 Images Database

A database with images of 15 hardware tools (some of them having moving parts) was used (see figure 1). The database contains 50 images of each tool acquired with different poses and shapes, in the case of tools with moving parts. The database was split in two equal length sets of data, one used in model training and the other for performance evaluation.

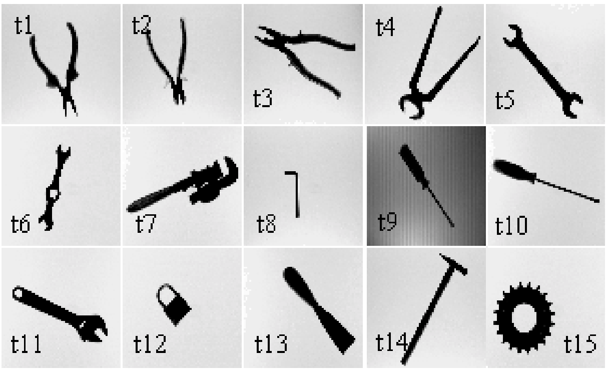

Figure 1: Images of hardware tools.

3.2 String Contour Descriptions

Each image was segmented to separate the object from the background and the object boundary was sampled at 50 equally spaced points (see figure 2). Objects shapes were encoded using an 8-directional differential chain code [7] (the representation is approximately invariant to translation rotation and scaling transformations).

3.3 Contour Modeling

HMMs and SFSGs were used for modeling of object's contours. Tests with HMM included fully connected and left-to-right models (LRHMM), trained by Viterbi and Baum-Welch algorithms. Furthermore, the number of states was modified in these tests in order to evaluate its influence on the final results.

Concerning the syntactic approach, the k-tail method [4] was adopted for grammatical inference. This algorithm, based on the concept of k-tail equivalence between states, forces some sort of alignment between the modeled strings, emphasizing and describ-

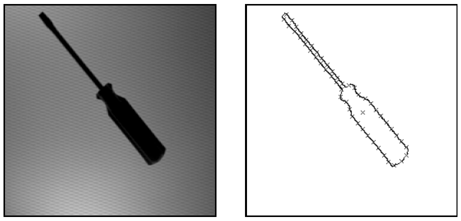

Figure 2: Example of tool and corresponding boundary sampled at 50 equally spaced points.

ing with greater detail their k-length tails. Several values for the k parameter were tested. Estimates of rules probabilities used the method of stochastic presentation.

3.4 Object Classification

According to the above probabilistic models, the probability of an observation sequence can be computed. Classification uses Bayes decision criterion.

4 Experimental Results

Figure 3 shows the typical network structures of the models under evaluation, illustrated for the tool class in figure 2. It can be seen that the k-tail method tends to enhance string tail alignment, while no emphasis is put to any particular string region by the HMMs.

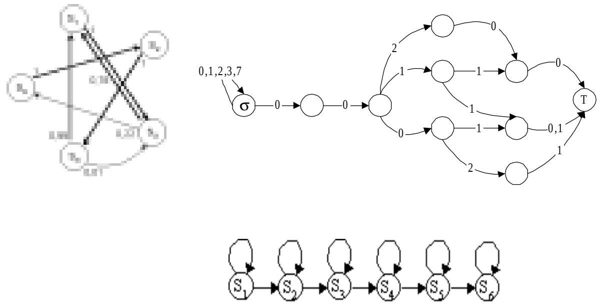

Figure 3: Fully connected HMM (5 states), SFSG (k=4) and left-to-right HMM (6 states) for the tool class in figure 2 (in the case of HMM only transitions with significant probability were represented).

Test	K=2			K=4			K=6		
set	P_e	P_m	P_{ec}	P_e	P_m	P_{ec}	P_e	P_m	P_{ec}
Global	7.76	0	7.76	8.03	.3	7.76	9.14	5.3	3.88
t1	20	0	20	20	0	20	20	0	20
t2	0	0	0	0	0	0	0	0	0
t3	36	0	36	36	0	36	36	0	36
t4	0	0	0	0	0	0	0	0	0
t5	0	0	0	0	0	0	28	28	0
t6	0	0	0	0	0	0	14	14	0
t7	0	0	0	0	0	0	4	4	0
t8	33	0	33	33	0	33	0	0	0
t9	0	0	0	0	0	0	3	3	0
t10	0	0	0	0	0	0	0	0	0
t11	0	0	0	0	0	0	0	0	0
t12	0	0	0	0	0	0	7	7	0
t13	0	0	0	0	0	0	0	0	0
t14	0	0	0	0	0	0	0	0	0
t15	0	0	0	5	5	0	38	38	0

Table 1: Classification results of test samples for several values of k. (P_e - probability of error; P_m - percentage of strings not recognized; P_{ec} - nearest-neighbor classification error rate).

Concerning the estimation of HMM parameters, two methods were evaluated. The best results were achieved with the Baum-Welch algorithm that produced higher values for the likelihood function $p(x|H)$. This algorithm was then used to train the models associated with the different tools. The best recognition rates were obtained with left-to-right HMM models (perfect recognition was achieved with a 20 state model). In this case the HMM becomes equivalent to a Dynamic Time Warping algorithm although it still has the advantage of allowing an automatic estimation of all the cost weights). Fully connected models achieved slightly worse results (99.7% with a 10 state model) and their training is slower. It must be stressed that some tools have moving parts and generate chain codes with different structures. The recognition score obtained with both types of models show that HMMs are able to capture the variability contained in the data, preserving the capability to discriminate between different objects.

In the previous tests the initial points of the objects boundaries were carefully selected. To assess the robustness of the HMM recognizer with respect to errors in the initial point estimate, the same methodology was applied to a set of data consisting of shifted versions of all training sequences. Recognition scores obtained with arbitrary initial point are 99.5% for fully connected models (10 states) and 98.9% for left-to-right models (20 states), confirming the robustness of the HMM approach.

According to the syntactic approach, grammars were inferred for different values of the tail parameter. The selection of k controls the level of generalization of patterns beyond the training set. Small values lead to over-generalization, while high values limits recognition to samples within the training set, the concept of *high* and *low* being pattern dependent. This is illustrated in table 1. Two types of errors are considered: 1- the test sample is recognized by the grammar associated with the correct class but is also recognized by another grammar with a higher probability; 2- the sample cannot be recognized according to the grammar for its class. The last type of errors is represented in the columns labeled P_m. The error probability (columns P_e) sums both type of errors. For $k = 2$, over generalization leads to errors in the classification of tools t1, t3 and t8 (t1 and t3 are articulated objects with moving parts). Higher values

Iter	Rec	t1	t2	t3	t4	t5-t7	t8	t9-t11	t12	t3-t15
1	92.24	k2	k2	k2	k2	*k2*	k2	*k2*	k2	*k2*
2	92.24	k3	k3	k3	k3	k2	k3	k2	k3	k2
3	92.24	k4	k4	k4	k4	k2	k4	k2	k4	k2
4	96.12	k5	k5	k5	k5	k2	*k5*	k2	*k5*	k2
5	96.12	k6	k6	k6	k6	k2	k5	k2	k5	k2
6	96.12	k7	k7	k7	k7	k2	k5	k2	k5	k2
7	96.12	k8	k8	k8	k8	k2	k5	k2	k5	k2
8	98.06*	k9	k9	*k9*	k9	k2	k5	k2	k5	k2
9	99.44*	*k10*	*k10*	k9	k10	k2	k5	k2	k5	k2
10	100*	k10	k10	k9	K8	k2	k5	k2	k5	k2

Table 2: Classification results.

of k lead to increased discriminating power but some test samples are not recognized by the inferred grammars (non-zero P_m). As the later are treated as errors, the probability of error P_e deteriorates. This problem can be circumvented by probabilistic nearest-neighbor classification based on error-correcting parsing [11] at the expense of higher computational costs. Table 1 shows that error rates thus obtained (P_{ec}) decrease as k increases: unrecognized samples are once again correctly classified and higher separability between pattern grammars is obtained. A trade-off must therefore be accomplished in order to achieve adequate modeling of the data and hence low classification error rates.

Considering the above results, the following approach was adopted: starting with low values for k, the classification confusion matrix was analyzed and the value of k was frozen for the set of tools correctly classified; the value of k was increased for the remaining tools and the process was repeated until no improvement was obtained. The results are summarized in table 2 where the second column indicates the global percentage of correct classifications. Frozen k values are represented in italics. The star in column 2 indicates that nearest-neighbor classification (error-correcting parsing) was performed.

It should be emphasized that the number of states thus obtained for each tool is variable, depending on the complexity of the contours involved, as opposed to the fixed length structure imposed in the HMM approach. For instance, for $k = 2$ the number of states ranges from 3 to 9 (average 6) as for the configurations of the 9th and 10th iterations the average number of states is 12 (ranging from 3 to 45). Therefore perfect object recognition was achieved with the syntactic approach with a less complex model than the one obtained using HMMs.

5 Summary and Conclusions

HMMs and SFSGs were used in 2D shape recognition of hardware tools. Objects shapes were encoded using a differential chain code. The k-tail method was adopted for grammatical inference and several values for the k parameter were evaluated. Tests with HMM included fully connected and left-to-right models.

Classification results obtained with the syntactic approach are comparable with the ones obtained with the LRHMM, meaning that, for classification purposes, it is sufficient to analyze only part of the contour. This is corroborated by experiments with the k-tail method applied to the last half of the string descriptions. For optimal object recognition, models inferred using the grammars paradigm were less complex than the corresponding LRHMMs. On the other hand, the syntactic approach is less robust than the HMM in the sense that larger training data sets are needed in order to achieve exact recognition, as the structure is conditioned by the training sets and structurally complete data sets are essential in order to obtain accurate identification of models. Future work on the subject includes comparative studies using other inference methods and models validation concerning the probabilistic structures.

References

[1] H. Bunke, U. Buhler, "2-D Invariant Shape Recognition using String Matching", *Proc. 2nd Int. Conf. On Automation, Robotics and Computer Vision,* 1992.

[2] F.Casacuberta, "Some Relations Among Stochastic Finite State Networks Used in Automatic Speech Recognition", *IEEE PAMI,*pp 691-695,1990.

[3] A. L. N. Fred, "Clustering of Sequences using a Minimum Grammar Complexity Criterion", *Grammatical Inference: Learning Syntax from Sentence,*Springer-Verlag, pp 107-116, 1996.

[4] K. S. Fu, T. L. Booth, "Grammatical Inference: Introduction and Survey - Part I and II", *IEEE PAMI,* pp 343-374, 1986.

[5] Y. He, A. Kundu, "2-D Shape Classification Using Hidden Markov Models", *IEEE PAMI,* pp 1172-1184, 1991.

[6] D. Huang, Y. Ariki, M. Jack, *Hidden Markov Models for Speech Recognition*, Edingburg University Press, 1990.

[7] A. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.

[8] L. Miclet, *Grammatical Inference, in Syntactic and Structural Pattern Recognition - Theory and Applications*, Scientific P., pp 237-290, 1990.

[9] L. Rabiner, "A Tutorial on Hidden Markov Models in Isolated Work Recognition", *Proceedings of IEEE,* vol. 77, pp 257-285, 1989.

[10] T. Starner, J. Makhoul, R. Schwartz, G. Chou, "On-line Cursive Handwritten Recognition Using Speech Recognition Methods", *Proc. of ICASSP,* Vol. 5, 125-128, 1994.

[11] M. G. Thomason, "Syntactic Pattern Recognition: Stochastic Languages", *Handbook of Pattern Recognition and Image Processing,*Academic Press, pp 119-142, 1986.