Learning Decision Trees and Tree Automata for a Syntactic Pattern Recognition Task*

José M. Sempere and Damián López

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, Valencia (Spain)
{jsempere,dlopez}@dsic.upv.es

Abstract. Decision trees have been widely used for different tasks in artificial intelligence and data mining. Tree automata have been used in pattern recognition tasks to represent some features of objects to be classified. Here we propose a method that combines both approaches to solve a classical problem in pattern recognition such as Optical Character Recognition. We propose a method which is organized in two stages: (1) we use a grammatical inference technique to represent some structural features of the characters and, (2) we obtain edit distances between characters in order to design a decision tree. The combination of both methods benefits from their individual characteristics and is formulated as a coherent unifying strategy.

1 Introduction

Syntactic Pattern Recognition is a well known research area from Artificial Intelligence in which the target task is to recognize objects from the real world (speech, image, medical signals, ...) which are represented as formal languages [HU79]. Mainly, the most common representations in these tasks have been some families of string languages (regular, context-free, ...), some families of tree languages (regular ones) or some families of graph languages (graphs based on vertex substitutions or hypergraphs with edge replacement). So, the goal in any syntactic pattern recognition learning task is to guess the hidden formal language from examples (strings, trees or graphs).

By the other hand, decision trees [Qu93] can be considered as tree-like representations of finite sets of *if-then-else* rules. This representation allows to take some decisions for the analysis of a set of attributes of a given concept. Mainly, the decision can be applied to a classification task, a predictive task or an advisement task (i.e. expert systems). During the last years, decision trees have been applied in the very promising area of *data mining* [MBK98] to extract knowledge from large databases.

In this work, we combine these two different approaches to the learning problem in order to construct a system to solve an Optical Character Recognition (OCR) task. Here, we will work only with handwritten isolated digits from 0

* Work supported by the Spanish CICYT under contract TIC2000-1153.

F.J. Perales et al. (Eds.): IbPRIA 2003, LNCS 2652, pp. 943–950, 2003.

to 9. Our solution is based on a two stages system. First, the system learns a set of tree automata (one per digit) by using an error-correcting technique based on a grammatical inference method. Basic concepts and methods on grammatical inference can be viewed in [AS83, Sa97]. Then, the system obtains a set of edit distances of every digit to every tree automaton. In the last stage, the system learns a decision tree from the last set of distances that will classify any digit according to a set of rules based on distances.

The structure of this work is as follows: First, we will explain the OCR task that the method attempts to solve. We will explain the learning methods on every phase (i.e. learning of tree automata and learning of decision trees). Finally, we will show some preliminary results from an experimentation using our approach and we will give some research guidelines for future works.

2 The Problem: Optical Character Recognition

The target problem of this work is related to the working area of Handwritten Recognition. Here, the general goal is to construct a robust system which be able to recognize any phrase or text that has been previously handwritten by a human being. This task has not yet completely solved. So, some subproblems are involved to solve this task. For example, there exists an increasing area that attempts to construct good segmentation rules in order to factorize any phrase in words an any word in letters or digits. Other researchers have focused their interest on constructing good language models for task-oriented systems (for example, some systems are focused on medical writings, or mathematics writings and so on). We will focus on another task which consists on isolated digits recognition. The solution to this task is important to construct more sophisticated systems. Here, the task is quite simple given that phrase and word segmentation tasks are avoided. This is the problem that we try to solve with a syntactic pattern recognition approach.

2.1 Representation of the Digits

First, we will consider how the real world objects will be represented. Let us observe in Figure 1 a digit 2 that has been obtained from a handwriting scanning.

Under our approach, the first stage to represent any digit is to obtain a *quad tree* (*qtree*) [HS79] from its digital image. A qtree can be constructed by drawing a square window around the digit and splitting the window in four windows of the same size recursively up to a predefined depth. In Figure 2 we can observe how the window of digit 2 is recursively split.

Once the system obtains the windows of the digit, then it assigns a label to every window of the smallest size. The systems assigns one label to every window depending on the grey scale (black, white or grey). So, every smallest window is represented by a label of a three symbols alphabet (i.e. $\{a, b, c\}$). The relationships between windows can be represented by a tree by using an up-down and left-to-right scanning of the qtree. In Figure 3 we can observe the tree

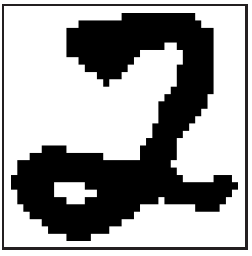

Fig. 1. Handwritten digit 2

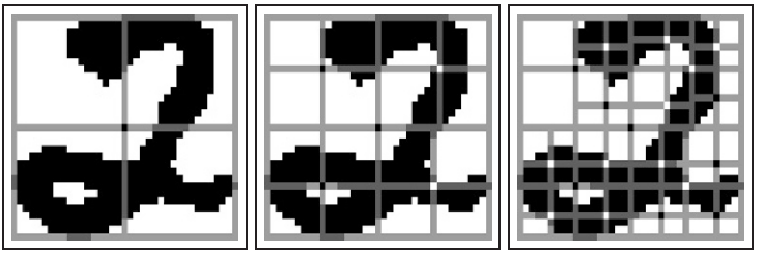

Fig. 2. Constructing a qtree for digit 2

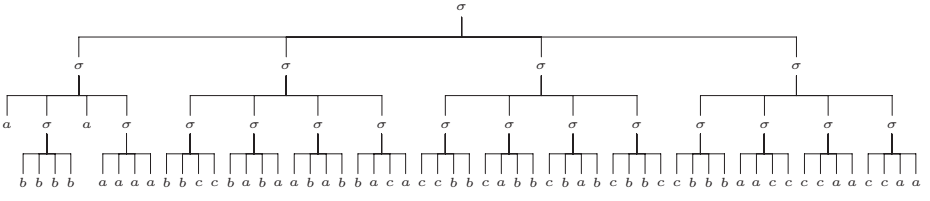

Fig. 3. Handwritten digit qtree with a depth that equals to 3. Label `a` corresponds to a at least 75% white square, label `b` corresponds to a at least 75% black square, and label `c` corresponds to a grey square

obtained for digit 2 by using a depth that equals to 3 while constructing the qtree. From now on, we will use this tree representation.

3 Learning Methods

We will use two different learning paradigms to solve the learning stage of the previously defined problem. First, we will use grammatical inference methods to construct a tree automaton from every set of trees representing the same digit. Then, we will go to a second learning stage to obtain a different representation of the digit based on distances of every digit (every tree) to every concept (every

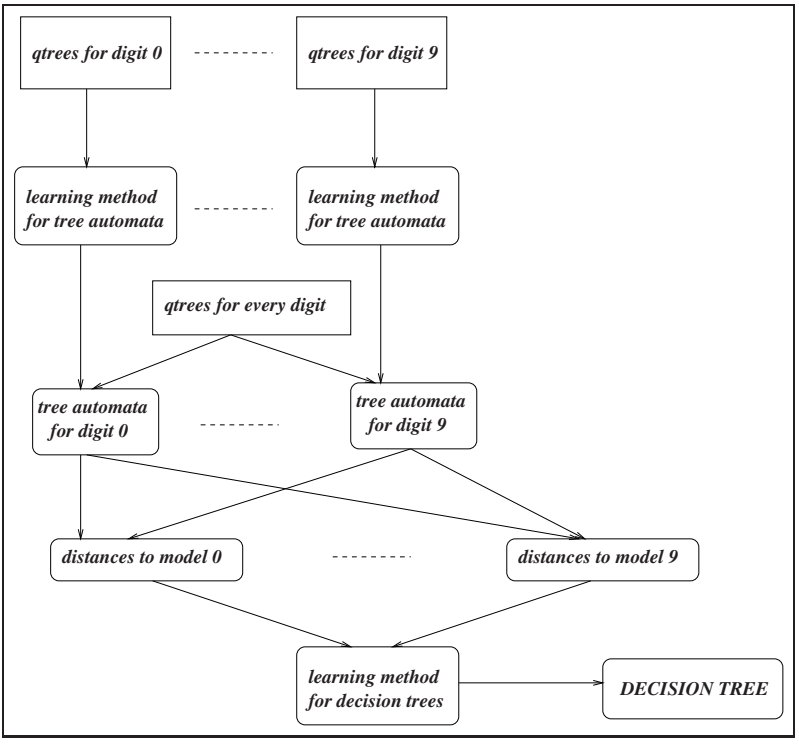

Fig. 4. Our learning strategy to solve the OCR problem

automaton). From this second representation we will infer a decision tree by using standard methods based on the entropy of the examples and distances. The learning scheme is showed in Figure 4.

Now we will explain the different methods that we have used at every learning stage.

3.1 Grammatical Inference of Error-Correcting Tree Automata

The first stage of our learning approach is based on a grammatical inference method for tree languages. Grammatical inference [AS83, Sa97] is an inductive approach to the learning problem based on the representation of concepts as formal languages. Here, as previously explained, we use trees to represent the digits for the OCR task.

Several methods have been proposed to infer tree languages from examples [Ga93, GO93, Sa92]. We will apply a method based on error-correcting distances from trees to tree automata. The definition of such distance is based on classical editing distances for strings to finite string automata [LSG00]. Once,

the distance has been defined then, the learning method is an error-correcting grammatical inference technique [LE02].

3.2 C4.5 Learning Algorithm

Learning decision trees is a classical topic on machine learning. A decision tree is a representation of a finite set of *if-then-else* rules. The main characteristics of decision trees are the following:

1. The examples can be defined as a set of numerical and symbolic attributes.
2. The examples can be incomplete or contain noisy data.
3. The main learning algorithms work under *Occam's razor principle* and *Minimum Description Length* approaches.

The main learning algorithms for decision trees have been proposed by Quinlan [Qu93]. First, Quinlan defined $ID3$ algorithm based on the *information gain* principle. This criterion is performed by calculating the entropy that produces every attribute of the examples and by selecting the attributes that save more decisions in information terms. Later, Quinlan defined $C4.5$ algorithm [Qu93] which is an evolution of $ID3$ algorithm. We will use $C4.5$ algorithm for the second learning phase. The main characteristics of $C4.5$ are the following:

1. The algorithm can works with continuous attributes (i.e. real data).
2. Information gain is not the only learning criterion.
3. The trees can be post-pruned in order to refine the desired output.

4 Experiments and Results

We have performed two experiments in order to carry out a first evaluation of our learning strategy. The digits that we have used for training and test is a subset from the data set "NIST SPECIAL DATABASE 3, NIST Binary Images of Handwritten Segmented Characters" [Ga94].

The protocol that we have performed in both experiments is the following one: First, we obtain the *qtree* representations of every digit in the data set. Then, we divide this set in two disjoint subsets (Set 1 and Set 2) and we apply to Set 1 the Error-Correcting inference technique in order to obtain a tree automaton for every digit. We calculate the distance of every digit to every automaton (so, every digit has ten attributes that represent the distances to every model). Then, we calculate the distances of every digit in Set 2 to every tree automaton.

From Set 1 and Set 2 we perform a learning plus testing phase for decision trees. We have used an implementation of $C4.5$ algorithm in C which is available from internet at J.R. Quinlan's Home Page [QuHTTP]. Observe that for every digit at Set 1 there is at least one distance with value 0, while this is not true in general for digits of Set 2.

Experiment 1

We have selected 3000 digits for Set 1 (300 samples for every digit) and 1000 digits for Set 2 (100 samples for every digit). We have performed three rounds on $C4.5$ algorithm in order to use different samples with or without distance 0. The results of this experiment are showed in Figure 5.

Round 1

Evaluation on training data (2666 items)

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
189	44(1.7 %)	173	48(1.8 %)	5.8 %

Evaluation on test data (1334 items)

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
189	122(9.1%)	173	120(9.0%)	5.8%

Round 2

Evaluation on training data (2667 items)

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
167	54(2.0%)	159	55(2.1%)	(5.7%)

Evaluation on test data (1333 items)

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
167	116(8.7%)	159	114(8.6%)	(5.7%)

Round 3

Evaluation on training data (2667 items)

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
205	56(2.1%)	187	60(2.2%)	6.5%

Evaluation on test data (1333 items)

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
205	87(6.5%)	187	86(6.5%)	6.5%

Fig. 5. Results for the first experiment

Experiment 2

We have selected 3000 digits for Set 1 (300 samples for every digit) and 2000 digits for Set 2 (200 samples for every digit). We have performed three rounds on $C4.5$ algorithm in order to use different samples with or without distance 0. The results of this experiment are showed in Figure 6.

Conclusions

It can be observed that, for every round that we have carried out on the experiments, the error median in training data is less than the one in test data. This is a trivial result that all learning methods would hold.

After, the pruning of the decision trees the median error decreases. It implies that some rules that $C4.5$ obtains are not useful for the classification task.

The results on the first experiment are better than in the second. Here, the input sample defines how the rules are extracted. In the first experiment, there is a number of examples with distance 0 to any tree automata which is three times those examples whose distances to every tree automata is not equal to 0. So, the input sample for constructing the tree automata is very important to obtain not only the distances but the decision tree.

Round 1

Evaluation on training data (3333 items)

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
315	88(2.6%)	293	93(2.8%)	8.1 %

Evaluation on test data (1667 items)

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
315	189(11.3%)	293	185(11.1%)	8.1%

Round 2

Evaluation on training data (3333 items)

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
305	86(2.6%)	281	93(2.8%)	7.9%

Evaluation on test data (1667 items)

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
305	186(11.2%)	281	185(11.1%)	7.9%

Round 3

Evaluation on training data (3334 items)

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
323	88(2.6%)	289	97(2.9%)	8.1%

Evaluation on test data (1666 items)

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
323	207(12.4%)	289	208(12.5%)	8.1%

Fig. 6. Results for the second experiment

Finally, an important remark is that the method has a better performance than some other methods that uses only a grammatical inference approach [LE02]. Furthermore, if we compare this method with some other methods based on geometrical approaches then, the differences between median errors can be balanced with the complexity behaviors (i.e. geometrical methods have a worst behavior than our approach under time and space complexities).

5 Future Works

From the initial results that we have obtained, our approach to the OCR problem has showed itself as a promising one. Anyway, we can point out to the following research guidelines in order to improve this work.

- We should enrich the attributes of every digit by including not only the distances but some other structural features.
- The criteria for decision tree learning could be change in order to take into account the distribution of the distances obtained from tree automata.
- Finally, we should apply this method to other pattern recognition tasks.

References

[AS83] D. Angluin, C. Smith. *Inductive Inference : Theory and Methods.* Computing Surveys, vol. 15. No. 3, pp 237-269. 1983. 944, 946

[Ga93] P. García. *Learning k-testable tree sets from positive data.* Technical Report DSIC II/46/1993. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. 1993. 946

[GO93] P. García and J. Oncina. *Inference of recognizable tree sets.* Technical Report DSIC II/47/1993. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia. 1993. 946

[Ga94] M. D. Garris. *Design and Collection of a handwriting sample image database.* Encycl. of Comp. Sci. & Tech. Marcel Dekker, N. Y. Vol 31, Supp. 16, pp 189-214. 1994. 947

[HU79] J. Hopcroft, J. Ullman. *Introduction to Automata Theory, Languages and Computation.* Addison-Wesley Publishing Co. 1979. 943

[HS79] G. M. Hunter and K. Steiglitz. *Operations on images using quad trees.* IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 1, No. 2, pp 145-153. 1979. 944

[LSG00] D. López, J. M. Sempere, P. García. *Error Correcting Analysis for Tree Languages.* International Journal on Pattern Recognition and Artificial Intelligence, Vol. 14, No.3, pp 357-368. 2000. 946

[LE02] D. López, S. España. *Error-correcting tree language inference.* Pattern Recognition Letters 23, pp 1-12. 2002. 947, 949

[MBK98] R. Michalski, I. Bratko and M. Kubat. *Machine Learning and Data Mining. Methods and Applications.* John Wiley and Sons LTD. 1998. 943

[Qu93] J. R. Quinlan. *C 4.5: programs for machine learning.* Morgan Kaufmann. 1993. 943, 947

[QuHTTP] R. Quinlan's Home Page **http://www.cse.unsw.edu.au/~quinlan/** 947

[Sa92] Y. Sakakibara. *Efficient learning of context-free grammars from positive structural examples.* Information and Computation 97, pp 23-60. 1992. 946

[Sa97] Y. Sakakibara. *Recent advances of grammatical inference.* Theoretical Computer Science 185, pp 15-45. 1997. 944, 946