ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 51-56

A Topological Transformation for Hidden Recursive Models

Fabrizio Costa, Paolo Frasconi, and Giovanni Soda
Dipartimento di Sistemi e Informatica, Università di Firenze
Via S. Marta 3, 50139 Firenze, Italy

Abstract. Discriminant hidden Markov models can be generalized from strings to labeled acyclic structures and, in particular, ordered trees [6, 7]. Inference and parameter estimation algorithms for this class of models can be derived in a straightforward way as special instances of inference and learning algorithms for Bayesian networks. However, if we are interested in building a discriminant model, in which arrows are directed towards the root of the tree, the model turns out to be intractable since the number of parameters grows exponentially with the number of neighbors of each node. In this paper we describe a topological transformation that maps ordered trees into binary trees, thus making the total number of parameters independent of the number of neighbors, as for the case of generative models. Besides reducing complexity, it also permits to deal with general ordered trees without imposing a priori a limit on the maximum outdegree. We show that the topological transformation maps regular sets of trees into regular sets of binary trees and, as a result, it does not affect the possibility of classifying trees with a finite state device. Finally, experimental results from a logo classification task are shown.

1. Introduction

We are interested in the classification of structured information. Instances in our learning domain are labeled graphs where the labels (attached to nodes) contain the attributes (numerical or categorical) used to describe atomic pieces of information, and the edges (not labeled) represent some sort of relationship between the atoms [7]. Connectionist models, that can learn in these structured domains have been recently introduced. In particular, recursive neural networks are a generalization of recurrent neural networks for strings that can learn about directed acyclic graphs. Hidden Markov models (HMMs) are another important architecture for learning strings or temporal sequences. Since the last few years, HMMs have been seen as a special case of Bayesian network [9]. Although mainly used within the unsupervised learning framework (i.e., as generative models) HMMs can be easily reformulated as discriminant models simply by reversing the direction of the arrows in the Bayesian network, and by adding an output variable associated with the class [2]. In this paper we focus on a generalization of discriminant HMMs for dealing with labeled trees. The

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 51-56

architecture is referred to as hidden recursive model (HRM) [7, 6] and has very close relationships to stochastic tree grammars [8]. The derivation of inference and learning algorithms is straightforward once one is able to describe HRMs as Bayesian networks. Given an ordered tree (such as the tree depicted in Figure 1) the model is constructed as follows. For each node v in the tree, two nodes in the Bayesian network are introduced: a hidden state variable X_v (white nodes in the figure) and input nodes (gray nodes in the figure) associated with the labels U_v. Then an output node Y for the class is connected to the hidden state associated with the root. When arrows are directed bottom up we obtain a discriminant model, whose message propagation scheme is similar to a frontier-to-root automaton [8]. The network is a model of the conditional distribution of the class Y given the labeled tree. Reversing the arrows and removing the output variable we would obtain a generative model of the *unconditional* probability distribution over labeled trees.

Unfortunately, the discriminant model suffers two major problems. The first one is a complexity problem: As a matter of facts, when k-ary trees are considered, the size of the conditional probability tables in the Bayesian network for the HRM grows exponentially with k. The model would be intractable from several points of view: space, time, and parametric (degrees of freedom) complexity.

The second problem is that k must be chosen to be the maximum outdegree found in the available data. Some domains (such as the logo recognition task that we discuss later) are likely to produce trees whose average branching factor is small, yet the maximum outdegree is relatively high. In this paper we develop a technique for overcoming these limitations that severely limit the practical applicability of HRMs.

There is another advantage of the proposed transformation when the automata to be inferred are characterized by sparse transition matrices (i.e., with many don't care conditions). In these cases the reparameterization induced by the transformation becomes a possible way to reduce the computational power of the HRM model working on equivalent binary trees to actual needs of the problem at hand.

In the next section we formally define the tree transformation. In section 3 we show preliminary results on a commercial logo recognition task, demonstrating the practical applicability of the methodology.

2. Topological transformation

Here we describe a simple technique for converting k-ary trees into binary trees. The basic idea is to shape the graphical model using the leftmost-child-right-sibling representation of the tree. This is achieved by removing arcs from children to parents and by forming a right-to-left linear chain connecting sibling state variables. In so doing, information flows through siblings and the state variable of the rightmost child summarizes the whole information associated with its siblings. For simplicity, we describe the transformation for trees labeled

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 51-56

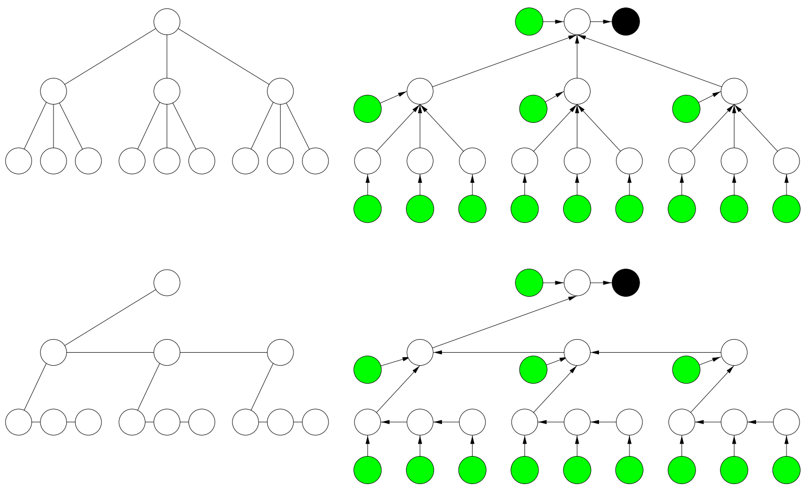

Figure 1: Top: an ordered tree (left) and its associated HRM (right). Bottom: the same tree after applying the topological transformation (left) and its associated HRM (right).

by categorical variables.

2.1. Definitions

Our domain is a set $\mathcal{T}$ of k-ary trees with nodes labeled by symbols belonging to a finite alphabet Σ. An instance $t \in \mathcal{T}$ is characterized by the triplet (V, E, f) where V and E denote the vertex set and the edge set respectively, and f is a labeling function that maps a vertex v into a label $u_v \in \Sigma$. For each node $v \in V$ let E_v denote the set of edges incident from v and R_v a total order on E_v. Moreover, let R_{vw} denote the position of edge (v, w). A (frontier-to-root) tree automaton is a tuple $A = (Q, \Sigma, \delta, F, G)$ where Q is a finite set of states, $F \subseteq Q$ is a finite set of accepting states, $G \in Q$ is the frontier state, and $\delta : \Sigma \times Q^k \to Q$ is the transition function (note that the arguments of δ are filled-in with the frontier state when a node has less than k children). By $\hat{\delta}(G, t)$ we indicate the state reached at the root when the automaton with state transition δ processes the tree t using G as frontier state. We also consider the case of don't care conditions (i.e., δ is not specified for some of its arguments). In this case, we denote by N the number of specified transitions. Clearly, in a fully specified automaton $N = |\Sigma| \cdot |Q|^k$.

2.2. Domain transformation

Let $t = (V, E, f) \in \mathcal{T}$. The corresponding transformed tree (V, E', f) is obtained as follows:

$$E' \quad = \quad \{(v_i, v_j) : (v_i, v_j) \in E, R_{ij} = 1\}$$

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 51-56

$$\cup \quad \{(v_i, v_j) : \exists v_k \in V, (v_k, v_i) \in E, (v_k, v_j) \in E, R_{ki} + 1 = R_{kj}\} \quad (1)$$

An example of this transformation is depicted in Figure 1.

This transformation does not affect the type of recognizer needed to classify trees in the domain. In particular, the following results hold true [5]:

Theorem 1 *For each tree automaton $A = (Q, \Sigma, \delta, F, G)$ operating on k-ary trees there exists a tree automaton $A' = (Q', \Sigma, \delta', F, G')$ for binary trees such that $Q \subset Q'$ and for each $t \in \mathcal{T}$ we have $\hat{\delta}(G, t) = \hat{\delta}'(G, t')$, where t' is obtained from t by applying the transformation 1.*

An obvious consequence of the above theorem is that the transformation 1 maps regular sets of k-ary trees into regular sets of binary trees. Clearly, the size of the set of states Q' can significantly grow with k. It is however possible to give a bound on $|Q'|$ based on the number of significant transitions N [5]:

$$\frac{N}{|\Sigma| \cdot |Q|} < |Q'| < k \cdot N \quad (2)$$

3. Case study: logo classification

Much attention has been paid to the problem of logo recognition working principally with methods capable of analyzing the structure of patterns. More recently, autoassociators have proved to be effective with Baird noise [1] and spot noise [3]. Our approach uses the topological technique applied to the HRM hybrid system to drastically reduce the spatial resources employed. The images are in the PNM format (Portable AnyMap) with 256 gray levels of dimension from 15 KByte to 160 Kbyte. All the logos come from the database built by the Document Processing Group, Center for Automation Research, University of Maryland. In the data set are pure text logos, pure graphic logos and mixed text-graphic logos.

To simulate the case of images obtained from photocopying or fax machines it has been added noise. The types of simulated noise are: stripe type (black rectangles), blob type (black circles), impulsive type (the color value of a pixel is changed with a defined probability) and rotational type (the logo is being rotated randomly by an angle in a specified range to simulate the practical impossibility of an exact positioning of the image to recognize). Other noise sources such as scaling, mirroring, smoothing, and smearing have not been considered in this study.

The algorithm used to extract a labeled graph from the image is a variant of the well known contour-tree [4]. The algorithm applies the following steps:

- the root of the tree is a contour that contains all other contours, if it exists, or the entire scene otherwise;
- a different contour corresponds to a different node of the tree ;
- a contour surrounded for more than 270 degrees by another contour is represented as a child node of the latter.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 51-56

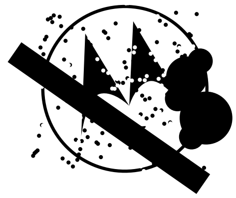

Figure 2: Example of a logo after all considered types of noise have been applied.

Table 1: Experimental results

Max stripe height	100 %
Max stripe width	30 %
Impulsive noise	8 %
Max rotation	± 60
Num. Hidden units	6
State variable cardinality	12
Training set accuracy	89 %
Test set accuracy	88 %

Max num. contiguous blobs	40
Max dimension blob	10 %
Impulsive noise	8 %
Max rotation	± 30
Num. Hidden units	12
State variable cardinality	12
Training set accuracy	89 %
Test set accuracy	88 %

Max stripe height	100 %
Max stripe width	30 %
Impulsive noise	8 %
Max rotation	± 180
Num. Hidden units	12
State variable cardinality	12
Training set accuracy	90 %
Test set accuracy	88 %

Max stripe height	100 %
Max stripe width	30 %
Num. Hidden units	12
State variable cardinality	12
Training set accuracy	98 %
Test set accuracy	98 %

The number of features extracted from each contour is 11 among which the area of the contour, the perimeter of the contour, the distance between the baricenter of the image of the whole image and that of the contour, the maximum curvature ray for convex and concave vertex, the number of concave and convex vertexes, etc.

For the experiment more than one type of noise has been added to the images varying the parameters in a random way within specified ranges obtaining 4096 elements (1024 for each class) half of which are used for training and half for test. The maximum dimensions for the stripes are normalized to the maximum dimension of the rectangle that contains the whole image.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 51-56

4. Conclusions

We have introduced a novel way of learning general k-ary trees. The topological transformation we have defined can significantly reduce the model complexity thus enabling the use of a discriminant Markovian approach that would be otherwise intractable. Our theoretical analyses indicate that good results should be expected whenever the grammar underlying the data is simple enough in terms of number of production rules. Clearly, productions in the transformed domain may be more complex and thus harder to learn. However, in our preliminary experiments on a real world task we have achieved a good classification performance while taking advantage of the complexity reduction offered by the proposed transformation.

References

[1] Henry S. Baird. Document image defect models. In H. Bunke H.S. Baird and K. Yamamoto, editors, *Structured Document Image Analysis*, pages 546–556. Springer Verlag, 1992.

[2] Y. Bengio and P. Frasconi. Input-output HMM's for sequence processing. *IEEE Trans. on Neural Networks*, 7(5):1231–1249, 1996.

[3] F. Cesarini, E. Francesconi, M. Gori, S. Marinai, J.Q. Sheng, and G. Soda. A neural based architecture for spot-noisy logo recognition. In *Proc. Int. Conf. on Document Analysis and Recognition*, 1997.

[4] G. Cortelazzo, G. A. Mian, G. Vezzi, and P. Zamperoni. Trademark Shapes Description by String-Matching Techniques. *Pattern Recognition*, 27(8):1005–1018, 1994.

[5] F. Costa. Tecniche probabilistiche per l'apprendimento di strutture dati con applicazioni al riconoscimento di forme. Master's thesis, University of Florence, 1998.

[6] P. Frasconi, M. Gori, and A. Sperduti. Hidden recursive models. In *Proc. of the 9th Italian Workshop on Neural Networks*. Springer, 1997.

[7] P. Frasconi, M. Gori, and A. Sperduti. A general framework for adaptive processing of data structures. *IEEE Trans. on Neural Networks*, 9(5):768–786, 1998.

[8] R. C. Gonzalez and M. G. Thomason. *Syntactic Pattern Recognition*. Addison Wesley, 1978.

[9] P. Smyth, D. Heckerman, and M.I. Jordan. Probabilistic independence networks for hidden Markov probability models. *Neural Computation*, 9(2):227–269, 1997.