

Adaptive Graphical Pattern Recognition Beyond Connectionist-Based Approaches

Giovanni Adorni¹, Stefano Cagnoni¹, and Marco Gori²

¹ Dipartimento di Ingegneria dell'Informazione
Università di Parma

² Dipartimento di Ingegneria dell'Informazione
Università di Siena
Via Roma, 56 - 53100 Siena, Italy

Abstract. This paper proposes a general framework for the development of a novel approach to pattern recognition which is strongly based on graphical data types. These data keep at the same time the highly structured representation of classical syntactic and structural approaches and the subsymbolic capabilities of decision-theoretic approaches, typical of connectionist and statistical models. Like for decision-theoretic models, the recognition ability is mainly gained on the basis of learning from examples, that, however, are strongly structured.

1 Introduction

As early pointed out by Wiener, a pattern can often be regarded as an *arrangement characterized by the order of the elements of which it is made, rather than by the intrinsic nature of these elements*. Hence, the causal, hierarchical, and topological relations between parts of a given pattern yield a significant information that seems to be useful in most human recognition processes. Among others, these motivations have given rise to the impressive development of approaches to syntactic and structural pattern recognition.

In the last three decades the emphasis on the research in the area of pattern recognition has hovered pendulum-like from decision-theoretic to structured approaches. Decision-theoretic methods are essentially based on numerical features which provide a global representation of the pattern by means of an appropriate pre-processing. Many different decision-theoretic methods have been massively experimented in the framework of connectionist models, which operate on sub-symbolic pattern representations. On the opposite, syntactic and structural pattern recognition and, additionally, artificial intelligence-based methods have been developed which emphasize the symbolic nature of patterns. Since their main focus is on expectations that can be derived from previous knowledge of the components that have to be detected in the patterns under consideration, such methods are often referred to as “knowledge-based” methods.

Such different approaches to pattern recognition have given rise to the long-standing debate which takes place between traditional AI methods, based on

symbols, and computational intelligence methods, which operate on numbers. However, both purely decision-theoretic or syntactical/structural approaches are limited when applied to many interesting real-world problems for opposite reasons.

It has been recently pointed out that traditional connectionist models conceived for processing static data types can properly be extended so as to deal with structured domains (see [12] for a survey on the topic). The basic idea is that the input graphs are processed by attaching a state variable to each node and performing a computation which is independent of the node. This assumption of independence gives rise to computational models which are also referred to as *stationary models* and represents at the same time the strength and the weakness of these models.

In pattern recognition, the remarkable feature of these connectionist-based models is that data can be graphs with real-valued nodes. As a result, the nodes can contain typical real-valued features and the links among the nodes can express the relationship among the pattern components.

In this paper we review briefly the connectionist approaches for structural domains and show how can they profitably be applied to different pattern recognition tasks. We discuss the extraction of appropriate pattern graphical representations and present general ideas for the application to classification and retrieval. We emphasize the potential advantages with respect to either traditional adaptive pattern recognition or to structural pattern recognition, but we also point out most severe limitations inherently related to the stationary assumptions in the propagation of the states attached to the input graphs. The extension to non-stationary models is currently investigated which would allow one to process input graphs depending on the particular node and preserve the general connectionist-based training scheme [13]. These models are likely to improve significantly the chance to deal with very hard pattern recognition problems.

This paper is organized as follows. In the next session we discuss the limitations of decision-theoretic and structural approaches so as to motivate the study of the proposed approach. In section 3 we show possible a possible extraction of data structure and briefly review the literature in the field. In section 4, we review briefly the basic ideas behind connectionist models for processing in structured domains and in section 5 we discuss the application to different pattern recognition tasks. Finally, limitations and future investigations are discussed in section 6.

2 Decision-Theoretic vs Structural Approaches

Syntactical and structural methods can nicely take the structure of patterns into account. The classical structural approaches to pattern classification consider each pattern class as a set of “strings” belonging to an appropriate grammar. Classifying a pattern then means matching such a string against the production rules of the grammar under consideration, to check if it is a valid string for such a grammar. This approach, for its inherent symbolic nature, and for its

inability to cope with corrupted instances of a class, is too little flexible when it comes to handling the noisy nature of patterns typically found in most real-world applications.

This problem has been early recognized and faced in different ways, to incorporate statistical properties into structured approaches to pattern recognition. The symbols used for either syntactical or structural approaches have been properly enriched with *attributes*, which are in fact vectors of real numbers representing appropriate features which are expected to allow for some statistical variability in the patterns under consideration.

Error-correction mechanisms have been introduced to deal with either noise or distortions [1]. Additionally, symbolic string parsing has been extended using stochastic grammars to model the uncertainty and the randomness of the accepted strings [1,16]. In these approaches, a probability measure is attached to the productions of the grammar G and, finally, accepted strings can be assigned an attribute representing the probability with which they belong to the class represented by G . Lu and Fu [21] combined error-correction parsing and stochastic grammars to attain a better integration with statistical approaches. A very detailed survey on the incorporation of statistical approaches into syntactical and structural pattern recognition can be found in [29]. Likewise, related approaches to integrate AI-based and decision-theoretic methods can be found in [6].

On the other hand, either parametric or non-parametric statistical methods can nicely deal with distorted patterns and noise, but are severely limited in all cases in which the patterns are strongly structured. The feature extraction process in those cases seems to be inherently ill-posed; the features are either global or degenerate to the pixel level.

The renewal of interest in artificial neural networks started in the middle eighties [25] suggested that pattern recognition methods shift from the complex task of feature selection and extraction to the development of effective architecture and learning algorithms. In principle, neural networks should be capable of extracting themselves optimal features for classification during the learning process. Moreover, “learning from examples”, which is typical of connectionist models, does not require that assumptions on the data probability distribution be made and is conceived to approximate virtually any non-linear discrimination function.

The field of neural networks, however, has now reached the maturity which is necessary to state that the previous belief is neither theoretically nor experimentally founded. There is evidence to claim that complex pattern recognition tasks, for instance highly structured, require architectures with many parameters to make the loading of the weights effective. This choice, however, makes generalization to new examples very hard [5]. On the other hand, the adoption of architectures with few parameters, which would facilitate the generalization, results in a very hard loading of the weights. The nature of this problem is partially addressed in the critical analyses on connectionist models by Fodor and Pylyshyn [11] and Minsky [24].

3 Graphical Representations of Patterns

The graphical representation of patterns has been extensively studied especially in conjunction with structural approaches to pattern recognition. In the last few years, most approaches have also incorporated grey levels and colors, thus giving rise to an impressive number of techniques with different performance (see e.g. [27,26,20,10,19,22,3,4,2,15]). Amongst others, a possible way to create structured representations is that of defining a relationship to generate the children of a given node. Different objects may be related to one another to give a scene its meaning. Such relationships can be of two kinds, the *is-a* relationship and the *part-of* relationship, which correspond to two fundamental operations, the combination of parts into wholes and the abstraction, respectively.

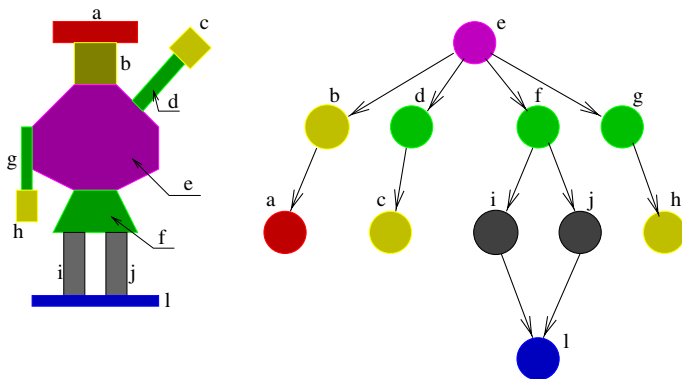


Fig. 1. A graphical item and a possible graphical representation based on border relationships.

As an example, Fig. 1 shows a possible way of creating a pattern graphical representation which is based on the *border relationship*. The construction, which is supposed to take place in a noisy-free environment, is based on a *supersource* that is chosen on the basis of the largest area of the building block. Consequently, the children are defined by finding the blocks at the border and then acting recursively¹.

4 Neural Networks on Structured Domains

Like data, the model can itself be structured in the sense that the generic variable $X_{i,v}$ might be independent of $q_k^{-1}X_{j,v}$, where, following the notation introduced in [12], q_k^{-1} is the operator which denotes the k -th child. The structure of independence of some variables represents a form of prior knowledge.

¹ An artificial set of pictures can be created using an attribute plex grammars. The creation of pattern recognition benchmarks to assess the experimental results of the methods described in this paper is currently under evaluation.

Basically, the knowledge of a recursive network yields topological constraints which often make it possible to cut the number of trainable parameters significantly. Let us consider a directed ordered graph so as for any node v one can identify a set, eventually empty, of ordered children $ch[v]$. For each node

$$\begin{aligned} \mathbf{X}_v &= f(\mathbf{X}_{ch[v]}, \mathbf{U}_v, v, \Theta_x) \\ \mathbf{Y}_v &= g(\mathbf{X}_v, v, \Theta_y). \end{aligned} \quad (1)$$

From the encoding network depicted in Figure 2 we can see a pictorial representation of the computation taking place in the recursive neural network. Each `nil` pointer is associated with a *frontier state* $\hat{\mathbf{X}}_v$, which is in fact an initial state that turns out to be useful to terminate the recursive equation. The graph plays its own role in the computation either because of the information attached to its nodes or for its topology. A formal description of the computation of the input graph requires sorting the nodes, so as to define for which nodes the state can be computed first. In the literature, this problem is referred to as *topological sorting*. A sort of data flow computation takes place where the state of a given node can only be computed once all the states of its children are known. To some extent, the computation of the output \mathbf{Y}_v can be regarded as a *transduction* of the input graph \mathbf{U} to an output \mathbf{Y} with the same skeleton² as \mathbf{U} . These *IO-isomorph* transductions are the direct generalization of the classic concept of transduction of lists. When processing graphs, the concept of IO-isomorph transductions can considerably be extended to the case in which also the skeleton of the graph is modified. Because of the kind of problems considered in this paper, however, this case will not be treated in this paper. The classification of DOAGs is in fact the most important IO-isomorph transduction for applications to pattern recognition. The output of the classification process corresponds with \mathbf{Y}_s , that is the output value of the variables attached to the supersource in the

² The skeleton of a graph is the structure of the data regardless of the information attached to the nodes.

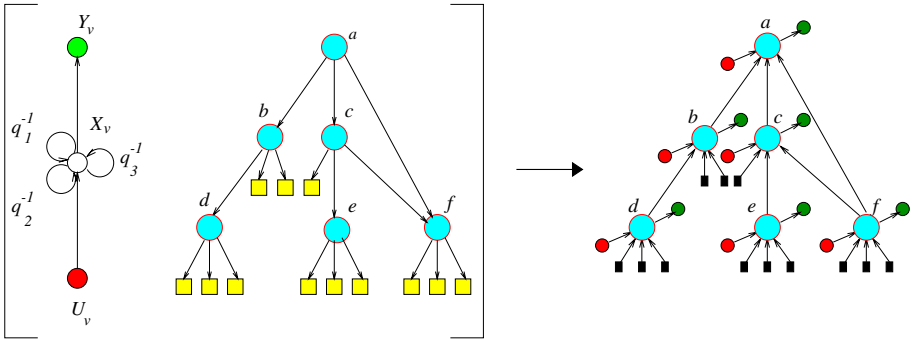


Fig. 2. Compiling the encoding network from the recursive network and the given data structure.

encoding network. Basically, when the focus is on classification, we disregard all the outputs \mathbf{Y}_v apart from the final values \mathbf{Y}_s of the forward computation.

The information attached to the recursive network, however, needs to be integrated with a specific choice of functions f and g which must be suitable for learning the parameters. The connectionist assumption for functions f and g turns out to be adequate especially to fulfill computational complexity requirements.

Let o be the maximum outdegree of the given directed graph. The dependence of node v on its children $ch[v]$ can be expressed by *pointer matrices* $\mathbf{A}_v(k) \in \mathcal{R}^{n,n}$, $k = 1, \dots, o$. Likewise, the information attached to the nodes can be propagated by weight matrix $\mathbf{B}_v \in \mathcal{R}^{n,m}$. Hence, the first-order connectionist assumption yields

$$\mathbf{X}_v = \sigma \left(\sum_{k=1}^o \mathbf{A}_v(k) \cdot q_k^{-1} \mathbf{X}_v + \mathbf{B}_v \cdot \mathbf{U}_v \right). \quad (2)$$

Like for list processing the output can be computed by means of $\mathbf{Y}_v = \sigma(\mathbf{C} \cdot \mathbf{X}_v)$.

The strong consequence of this graphical representation for f and g is that, for any input graph, an encoding neural network can be created which is itself a graph with neurons as nodes. Hence, the connectionist assumption makes it possible to go one step further the general independence constraints expressed by means of the concept of recursive network. The corresponding encoding network turns out to be a graph whose links arise either because of the graph topology or because of independence between variables or because of the connectionist representations of the functions f and g themselves. The encoding networks associated with equation 2 and equation 2 are depicted in Fig. 3 in the particular case of stationary models, in which the parameters are independent of the node v . Encoding neural networks turns out to be weighed graphs, that is there is always a real variable attached to the edges (weight). Note that the architectural choice expressed by equation 2 can be regarded as a way to produce a multilayer-based map of the state which, however, transforms the input \mathbf{U}_v by means of one layer only. Obviously, one could also adopt a multilayer-based architecture for implementing $f(\mathbf{X}_{ch[v]}, \mathbf{U}_v, \Theta_x)$. Likewise, the function $g(\mathbf{X}_v, \Theta_y)$ can be implemented by a multilayer perceptron. In Figure 3 this function is created by means of one layer of sigmoidal neurons only. Finally, in the framework of supervised learning, we can easily extend backpropagation to adjust the shared parameters of the encoding neural networks. The backpropagation takes place on neural nets which inherit the data structure and, therefore, the corresponding learning algorithm is referred to as *backpropagation through structure*.

5 Facing Pattern Recognition Problems

The term adaptive graphical pattern recognition was firstly introduced in [9], but early experiments using this approach were carried out in [14]. Graphs are either in the data or in the computational model, since the adopted connectionist models inherit the structure of the data graph and, moreover, have their

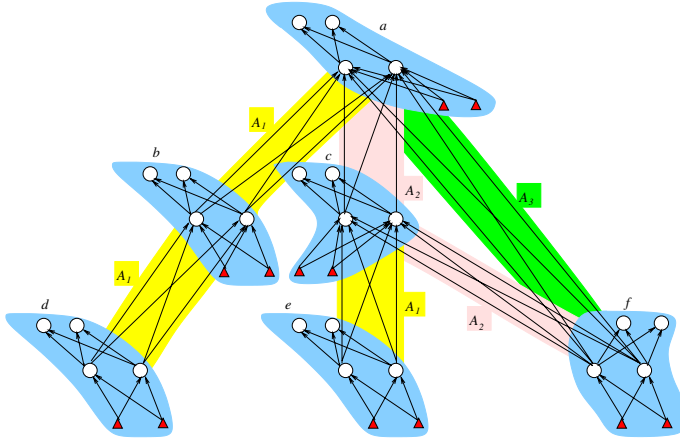


Fig. 3. The construction of a first-order recursive neural network from the encoding network. The construction holds under the assumption that the frontier states are null.

own graphical structure which expresses the dependencies on the single variables. Basically, graphical pattern recognition methods integrate structure into decision-theoretic models. The structure can be introduced at two different levels. First, we can introduce a bias on the map (e.g. receptive fields). In so doing the pattern of connectivity in the neural network is driven by the prior knowledge in the application domain. Second, each pattern can be represented by a corresponding graph. As put forward in the previous section, the hypothesis of directed ordered graphs can be profitably exploited to generalize the forward and backward computation of classical feedforward networks.

The proposed approach can be pursued in most interesting pattern recognition problems.

– *classification*

Recursive neural networks seem to be very appropriate for either classification or regression. Basically, the input structured representation is converted to a static representation (the activation in the hidden layer), which is subsequently encoded into the required class. This approach shares the advantages and disadvantages of related MLP-based approaches for static data. In particular, the approach is well-suited for complex discrimination problems, but it is not very adequate for verification purposes. Methods like growing and pruning can be successfully used for improving the learning process and make the trial and error approach more systematic. It's worth mentioning that recursive nets can profitably be used for classification of highly structured inputs, like documents using XY-tree representations. Unfortunately, in this particular kind of application the major limitation turns out to be that the number of classes is fixed in advance. More appropriate models in this case can be based on the preliminary ideas reported in [17].

- *verification*

Neural networks in structured domains can be used in verification problems, where one wants to establish whether a given pattern belongs to a given class against any possible input. Unlike pattern classification, one does not know in advance the kind of inputs to be processed. It has been pointed out that sigmoidal multi-layered neural networks are not appropriate for this task [18] and, consequently, also recursive neural networks which represents the natural extension to the case of structured domains are not appropriate for verification tasks. However, like for multilayer networks, the adoption of radial basis function units suffices to remove this limitation.

- *retrieval*

The connectionist models introduced in this paper and the related extensions are very good candidates to deal with many interesting image retrieval tasks. In particular, the proposed models introduce a new notion of similarity which is constructed on the basis of the user's relevance feedback. In most approaches proposed in the literature, the queries either involve global or local features, but disregard the pattern structure. The proposed approach makes it possible to retrieve patterns on the basis of a strong involvement of the pattern structure, since the graph topology plays a crucial role in the computation. On the other hand, since the nodes contain a vector of real-valued features, the proposed approach can also potentially exhibit retrieval capabilities which arise from the sub-symbolic nature of the patterns.

6 Discussion

The renewal of interest in neural networks raised an impressive interest in the community of pattern recognition, where many people were stimulated by the potential capabilities of these models to deal with any probability distribution. Nowadays, there is theoretical and practical evidence to conclude that these learning-based models cannot deal effectively with many interesting problems in which patterns exhibit a significant structure. In this paper, we claim that the new wave of connectionist models for processing in structured domains is likely to offer many opportunities to face new complex pattern recognition tasks, including the retrieval of images in visual data bases. Moreover, the basic ideas behind the extension of connectionist models to structured domains also apply to statistical models. There are, however, a number of limitations of the models reviewed briefly in this paper.

- *graph topology*

The proposed models operates on directed ordered graphs. This hypothesis makes it possible to carry out a forward computation and extend feedforward networks and backpropagation straightforwardly. When dealing with cyclic graphs, however, at the moment the only computational schemes that have been devised are based on and relaxation methods [28].

- *causality*

The proposed models represent a natural extension of the processing of sequences by causal dynamical systems. In pattern recognition, the hypothesis of causality could be profitably removed, since there is no need to carry out an on-line computation at node level.

- *stationary*

The homogeneous computation which takes place at node level may not be adequate in many pattern recognition problems. This has been already pointed out in [9], where a simple solution has been adopted to face the non-stationarity. The graphs are partitioned into different sets depending on the number of nodes and are processed separately. A more general and promising computational scheme has been devised in [13], where the non-stationarity can be given a linguistic description.

The development of a systematic theory of connectionist and statistical models to deal with structured domains is a new promising research field which has already yielded interesting results. However, the actual application of this theory to pattern recognition is still in its infancy and there are only a few preliminary encouraging results. A massive comparative application to classical pattern recognition problems is the only way to follow to assess the effectiveness of the proposed approach.

Acknowledgements

This paper is the result of the discussion with many people who contributed in different way to the theory of adaptive computation on structured domain and its applications. In particular we would like to thank Andreas Küchler, Paolo Frasconi, Alessandro Sperduti, and Marco Maggini who contributed to shape most of the ideas for the application to pattern recognition herein presented.

References

1. A. Aho and T.G. Peterson, "A minimum distance error-correcting parser for context-free languages," *SIAM Journal of Computing*, no. 4, pp. 305–312, 1972.
2. H. Asada and M. Brady, "The curvature primal sketch," in *Proc. Workshop on Computer Vision* (M. Caudill and C. Butler, eds.), (Annapolis, MD), pp. 609–618, 1984.
3. D. Ballard, "Strip trees: a hierarchical representation for map features," in *Proc. of the 1979 IEEE Computer Society Conference on Pattern Recognition and Image Processing*, (New York, NY), pp. 278–285, IEEE, 1979.
4. D. Ballard, "Strip trees: a hierarchical representation for curves," *Communications of the ACM*, vol. 24, pp. 310–321, 1981.
5. E. Baum and D. Haussler, "What size net gives valid generalization?," *Neural Computation*, vol. 1, no. 1, pp. 151–160, 1989.
6. H. Bunke, "Hybrid pattern recognition methods," in *Syntactic and Structural Pattern Recognition: Theory and Applications* (H. Bunke and A. Sanfeliu, eds.), ch. 11, pp. 307–347, World Scientific, 1990.

7. Clarke, F., Ekeland, I.: Nonlinear oscillations and boundary-value problems for Hamiltonian systems. *Arch. Rat. Mech. Anal.* **78** (1982) 315–333
8. Clarke, F., Ekeland, I.: Solutions périodiques, du période donnée, des équations hamiltoniennes. *Note CRAS Paris* **287** (1978) 1013–1015
9. M. Diligenti, M. Gori, M. Maggini, and E. Martinelli, “Graphical pattern recognition,” in *Proceedings of ICAPR98*, 1998.
10. C. Dyer, A. Rosenfeld, and H. Samet, “Region representation: Boundary codes from quadrees,” *Communications of the ACM*, vol. 23, pp. 171–179, 1980.
11. J. Fodor and Z. Pylyshyn, “Connectionism and cognitive architecture: A critical analysis,” *Connections and Symbols*, pp. 3–72, 1989. A Cognition Special Issue.
12. P. Frasconi, M. Gori, and A. Sperduti, “A general framework for adaptive processing of data structures,” *IEEE Transactions on Neural Networks*, vol. 9, pp. 768–786, September 1998.
13. P. Frasconi, M. Gori, and A. Sperduti, “Integration of graphical-based rules with adaptive learning of structured information,” in *Hybrid Neural Systems*, Springer Verlag, R. Sun and S. Wermeter Eds, March 2000
14. P. Frasconi, M. Gori, S. Marinai, J. Sheng, , G. Soda, and A. Sperduti, “Logo recognition by recursive neural networks,” in *Proceedings of GREC97*, pp. 144–151, 1998.
15. H. Freeman, “On the encoding of arbitrary geometric configurations,” *IRE Trans.*, vol. EC-10, pp. 260–268, 1961.
16. K. Fu, *Syntactic pattern recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
17. C. Goller and M. Gori, “Feature extraction and learning vector quantization for data structure,” in *Proceedings of SOCO’99*, (Genoa (Italy)), June 1999.
18. M. Gori and F. Scarselli, “Are multilayer perceptrons adequate for pattern recognition and verification?”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1121–1132, 1998.
19. G. Hunter, *Efficient Computation and Data Structures for Graphics*. PhD thesis, Dept. of Electrical Engineering and Computer Science, Princeton University, Princeton, NJ, 1978.
20. G. Hunter and K. Steiglitz, “Operations on images using quadrees,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, pp. 145–153, 1979.
21. S. Lu and K. Fu, “Stochastic error-correction syntax analysis for recognition of noisy patterns,” *IEEE Transactions on Computers*, no. 26, pp. 1268–1276, 1977.
22. D. Meagher, “Geometric modelling using octree encoding,” *Computer Graphics and Image Processing*, pp. 129–147, 1982.
23. Michalek, R., Tarantello, G.: Subharmonic solutions with prescribed minimal period for nonautonomous Hamiltonian systems. *J. Diff. Eq.* **72** (1988) 28–55
24. M. Minsky and S. Papert, *Perceptrons - Expanded Edition*. Cambridge: MIT Press, 1988.
25. D. Rumelhart, J. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1. Cambridge: MIT Press, 1986.
26. P. Salembier and L. Garrido, “Binary partition tree as an efficient representation for filtering, segmentation and information retrieval,” in *IEEE Int. Conference on Image Processing, ICIP’98*, vol. 2, (Los Alamitos, CA), pp. 252–256, IEEE Comp. Soc. Press, 1998.
27. H. Samet, “Spatial data structures,” in *Modern Database Systems: The Object Model, Interoperability, and Beyond* (W. Kim, ed.), pp. 361–385, Reading, MA: Addison Wesley/ACM Press, 1995.

28. A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures" *IEEE Trans. on Neural Networks*, vol. 8, no. 3, pp. 714–735, 1997.
29. W.-H. Tsai, "Combining statistical and structural methods," in *Syntactic and Structural Pattern Recognition: Theory and Applications* (H. Bunke and A. Sanfeliu, eds.), ch. 12, pp. 349–366, World Scientific, 1990.