Rulegraphs for Pattern Recognition

Adrian Pearce, Terry Caelli and Walter F. Bischof
Computer Vision and Pattern Recognition Laboratory (CVPRL)
Department of Computer Science, University of Melbourne, Parkville, 3052, Australia
adrianp@cs.mu.OZ.AU, tmc@cs.mu.oz.au, wfb@psych.ualberta.ca Fax: +613 348 1184

Abstract

We present and compare two new techniques for Learning Relational Structures (RS) as they occur in 2D pattern and 3D object recognition. The two techniques, Evidence-Based Systems (EBS) and Rulegraphs (RG) combine techniques from Computer Vision with those from Machine Learning, Uncertainty Theory, and Graph Matching. EBS have the ability to generalize pattern rules from training instances in terms of bounds on both unary (single part) and binary (part relation) numerical features. Rulegraphs check the compatibility between rules by combining Evidence Theory with Graph Matching techniques. The two systems are tested and compared using a difficult pattern recognition problem.

1 Introduction

In the context of Computer Vision, Relational Structures (RS) refer to the representation of patterns or shapes in terms of attributes of parts and part relations. To this date, little attention has been devoted to the problem of developing techniques for the automated learning of such relational descriptions - and to the associated problem of generalization. This paper is concerned with these issues and how they can be used to solve difficult pattern and object recognition problems. There are at least three goals of recognition systems:

- Accuracy in identification or classification,
- Generalization from the data used during Learning to other data sets at run time, and,
- Efficiency in both Learning and Test Modes.

The degree to which these goals are achieved depends on the methods used for signal processing, segmentation, feature extraction and matching. Further, we show how a new class of relational learning algorithms can be used to develop such recognition systems. These methods combine principles from Evidenced-Based Systems (EBS) , typically used in Expert Systems, with the automatic generation of RS for pattern recognition and can be seen to have some parallels to current Machine Learning programs such as FOIL [8] - though the applications in Vision are so specialized that the generic use of those standard symbolic learning technologies is inefficient.

It goes without saying that traditional pattern recognition has been quite successful for simple isolated patterns. However, traditional methods do not perform well when pattern complexity is high, as is the case with 3D object recognition or with complex and highly similar 2D patterns. This can be attributed to a number of reasons. Descriptions of complex patterns in terms of features characterizing the whole pattern are often inadequate to encode the variability of class samples.

In 3D object recognition, for example, an object may be described by features characterizing surface parts (unary features) such as average curvatures or boundary shape descriptors, and by features describing part relations (binary features) such as centroid distance or mean normal angle differences (see [9]). However, these part and part-relation features have to be linked together into a relational structure (RS) in order to define patterns uniquely. The problem with traditional pattern recognition techniques is, in general, that they rarely use Machine Learning techniques, and, in particular, that they rarely consider generalization of rules corresponding to vertices and edges of relational structures. Our aim here is to combine the RS representation with generalization in two different ways, defined by Evidenced-based Systems (EBS) and Rulegraphs (RG).

2 Evidenced-Based Systems

Object Recognition is a difficult problem because parts of different objects can be quite similar, sharing similar regions in feature space and thus within-class variance may exceed between-class variance substantially (*the back of your head is more similar to the back of my head than the front of my head is to the back of my head*). The EBS solution to this problem involves the development of an intermediate representational stage, a so-called *rule* stage, where an attempt is made to capture the predominant characteristics of the sample densities by grouping them into spatially-delimited regions in feature space. By defining the bounds on such regions as conditions for their activation, *evidence-*

weights are associated with each cluster which correspond to the likelihood that activation of a given rule contributes positive or negative evidence for the existence of a given object. The rules or clusters define the degree of generalization from samples and, for simplicity, such clusters are typically defined by hyper-rectangles and oriented along the feature space axes to allow for rules of the conjunctive form:

IF Bounds$_{lower,upper}$ (feature$_1$,....,feature$_n$) **THEN** Evidence Weights(w$_1$,...,w$_m$) **ELSE** no evidence

where the feature indices (1,..,n) refer to the unary features of each patch or the relational(binary) features defined between patches and the weight indices refer to the actual object or classes (1,..,m) (see Figure 1 Top left).

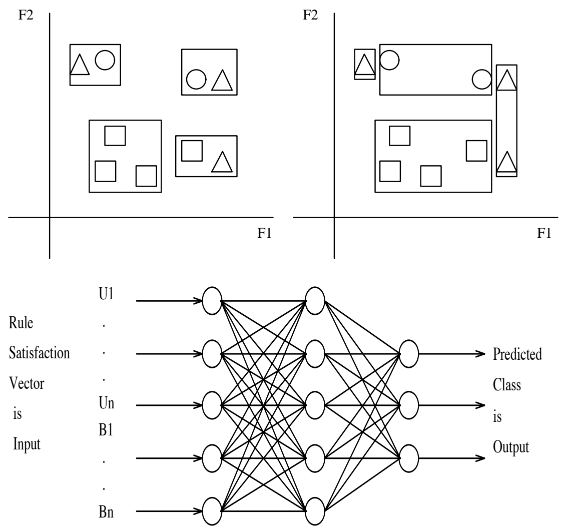

Figure 1: **Top:** A simple example showing a 2-D feature space with features F1 and F2, where clusters are not coextensive with classes, and the minimum distance (left: K-Means), and minimum entropy (right) clustering solutions. Here there are three classes and four clusters. **Bottom:** Neural Network for an evidence weight estimation problem with five input rules and output classes. The input is a vector representing unary (U) and binary (B) rule satisfaction. For the output layer, each node corresponds to a class, and the result of the classification is determined as the node with the greatest activity (From [2]).

In previous work, for example, the work of Jain and Hoffman [5], rules were generated by first clustering the samples in feature space using a minimum spanning tree technique. The relative frequencies of class samples was then used to determine rule weights. Our clustering algorithm - Minimum Entropy Clustering - endeavors to change the position and size of a fixed number of rectangles(clusters) to maximally separate the occurrences of class samples per cluster. In other words we relabel the cluster membership of each sample to minimize the entropy function:

$$H_{min} = \min_{i \in j} \left\{ -\sum_{j} \sum_{k} p_{jk} \ln p_{jk} \right\} \qquad (1)$$

where p_{ij} is the probability of class i occurring in cluster j and the probability is determined from the relative frequency of class samples within a given cluster solution (see Figure 1 Top right). This is a Combinatorial Optimization problem [1] requiring the use of Simulated Annealing where all samples are relabeled for clusters to minimize the entropy function (see [2]). Rules are generated over both unary (U) and binary (B) feature spaces .

We then use Supervised Learning to solve the weight estimation problem and, at the same time, *to learn the relationships between unary and binary feature states.* We have used a specific type of Neural Network model where: Input nodes correspond to clusters, output nodes correspond to classes and there is one hidden layer, with the number of nodes being the larger of the input or output node numbers. The evidence weights (w_{ij}) are then determined by the connections between input-hidden-output layer nodes. In particular, each hidden layer node is connected to each and every unary and binary rule (see Figure 1). This allows for the reinforcement of co-occurrences between unary and binary feature states: parts and relations - up to the set of equivalent such pairings over the different views: a form of **implicit relational structure learning**. The relationship between rules and object classes is formulated by the standard Neural Network equations [4].

This hybrid approach differs from direct Neural Net implementations in two respects. First, feature space partitioning is not the same as that obtained with multi-layered Perceptrons, and second, we have defined constraints on the hidden layers to determine evidence weights that accord with the conjunctive forms. For these reasons the types of rules and weights are guaranteed to satisfy the representational constraints - something which is not guaranteed in direct Neural Net implementations [6].

3 Rulegraphs

Although the EBS-NNet systems encode some relational structure (RS) information in the hidden layers of the Neural Network (see Figure 1), it does not guarantee *solutions* to the label-compatibility problem since different combinations of unary and binary feature states can trigger the same hidden node. That is, the EBS-NNet essentially creates a multi-labeled graph representation in which specific combinations of labels correspond to specific sample occurrences of parts and relations. However, in EBS such graphs have weighted vertices and edges in the form of class evidence vectors. In contrast to Neural Networks, the idea behind Rulegraphs is to use these weights together with explicit label-compatibilities to prune the search space in graph matching: a form of **explicit relational structure learning.**

The technique relies on two simple principles: First, sets of model graphs and their vertices are reduced by generalization (collecting like features for different classes into hyper-rectangles in feature space). Second, search for subsets of compatible labels between rules is constrained using evidence weights produced by an EBS. The matching process involves graphs of cardinality no greater than the number of unary rules (as they correspond to the Rulegraphs vertices), and thus is more efficient than classical Graph Matching procedures.

A *rulegraph is a graph of rules* in which vertices correspond to unary rules and edges correspond to binary rules according to the following connection criterion:

Two unary rules R_i^u and R_j^u are connected by a binary rule R_k^b if there exists labels X, Y such that $X \in R_i^u$ and $Y \in R_j^u$ and $XY \in R_k^b$.

A *rulegraph model* for a training pattern corresponds to a graph where unary and binary rules replace model parts and their relationships (see Figure 2). Rulegraphs explicitly represent the rules produced by EBS and their interrelations via shared label instances and they capture compatibility information about the structural aspects of the pattern description. The *likelihood* of a rulegraph corresponding to each class in the training set may be determined from the evidence weights of each rule vertex and edge. In this implementation of the EBS, we have used the relative class frequencies to determine evidence weights. For each unary and binary EBS-rule, we simply determined the evidence weights from the class frequencies within the rule's bounds in feature space.

At run time, parts in the sample pattern activate unary and binary rules based on their feature states and Compatibility between the sample and model relies on a consistency of the mapping states and can be checked using Constraint Propagation Methods. Among the methods for checking compatibility between individual parts, Subgraph Isomorphism has been the most effective in differentiating between samples [3]. In rulegraphs several labels may exist in each rule vertex and this gives rise to *multiple* mapping states involving the same labels. A new technique which determines label compatibility between *rules* instead of *parts* - the Label Compatibility Checking Method - is outlined for rulegraphs in [7]. This method uses a modified existence crite-

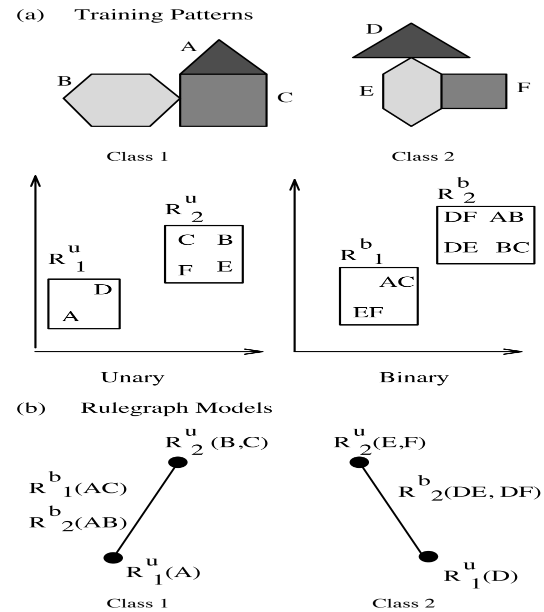

Figure 2: Training patterns are used in (a) to label the unary and binary rules according to the mapping of the parts and their relationships into each feature spaces. Unary rules are labeled with single labels and binary rules are labeled with label pairs. Rulegraph models may then be formed, according to the connection criterion, and these are shown in (b). (From [7]).

rion capable of handling multiple labels and binary evidence weights are used for updating the Mapping States. The cardinality of the search problem (disregarding label-compatibility checks) has already been reduced to the number of unary rules instead of the number of primitive parts. The evidence weights can be used to direct the search toward rules and models for which strong evidence exists. To achieve this, we use A* search combined with the Bayesian evidence weight metric to allow probabilistic pruning of the search tree.

4 Comparison of Systems

We have compared classification performance and complexity of Rulegraph Matching to that of EBS using the Neural Network (EBS-NNet) and that of Traditional Subgraph Isomorphism using Branch-and-Bound for the patterns shown in Figure 3a. The features extracted were: Unary - perimeter and colour and Binary - distance between centers and sum of distance between corners. For the training set (TS), four fragments were extracted from each of the 15 patterns (see Figure 3b). Similarly, four *different* fragments were extracted from each of the 15 patterns for the test sample set (SS) (see Figure 3c). In addition, both unary and binary feature attributes were distorted using additive Gaussian noise with a variance corresponding to five percent of the original feature variance. This scheme of pattern sampling simulates occlusion and data loss is consistent with sampling regimes found in 3D-Object Recognition and other complex Pattern Recognition problems. Indeed, the data is not guaranteed to be perfectly classifiable and exhibits many characteristics fundamental to problems encountered in Pattern Recognition.

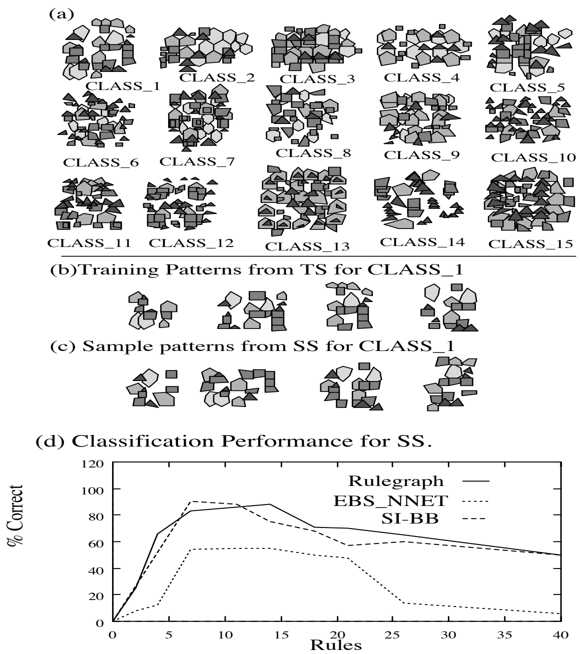

Figure 3: In (a) all 15 classes for the Blocks Data are shown. In (b) four training patterns (or views) are shown for class 1 and four different test patterns which have been distorted features and have missing (occluded) parts are shown in (c). Classification performance is shown for different numbers of rules for the distorted and occluded Sample Set (SS) is shown in (d) (From [7]).

The rule generation scheme used Leader clustering and is based on the nearest neighbor method and required only a single parameter, a distance threshold. Smaller thresholds generate more specific - and more numerous - rules with lower class entropy values with respect to the TS and higher thresholds generate more general - less numerous - rules that are resilient to variation and distortion of the data. As a result there is an optimum number of rules associated with any particular Pattern Recognition problem though, in this example, we have run tests with different numbers of rules.

In summary, the results indicate that the rulegraphs (88 percent) offer a classification performance close to the obtainable optimum using Traditional Subgraph Isomorphism (90 percent) and a significant improvement over Evidence-Based Systems (55 percent), in particular for occluded and distorted data. The high classification performance of Rulegraph Matching can be attributed to its ability to encode more class information through the use of labels, while, at the same time, allowing for general rules that are resilient to variation and distortion of data. Using the same data sets, we also compared the computational complexity of the different methods. The computational complexity of the rulegraph method was much lower than that of Subgraph Isomorphism (using Branch-and-Bound) and similar to that of the Neural Network. Further, it should be noted that rulegraphs are superior to neural nets at *learning time*: frequencies and labels of training data are merely recorded, while Neural Nets require substantial training time for Backpropagation (for details see [7]).

5 Discussion

In this paper, we have discussed one of the more difficult problems in pattern and object recognition: that of developing efficient and accurate methods for developing prototypical descriptions of shapes which involve the definitions of part and part relations. We have argued that techniques from Machine Learning can help solve this problem, as well as to address the generalization problem and the problem of *pre-compiling* search strategies for matching. In particular, we have discussed two techniques developed in our group to attain these goals - all of which involve various combinations of standard representation and search methods from the literature. What differentiates this work is just how we have compiled each method and how they have been adapted to solve problems in vision.

References

[1] E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines.* New York: Wiley, 1989.

[2] T. Caelli and A. Pennington. "An improved rule generation method for evidence-based classification systems." *Pattern Recognition*, Vol. 26, No. 5, pp733-740, 1993.

[3] N. Deo. *Graph Theory with Applications to Engineering and Computer Science* N.J.: Prentice-Hall Inc., 1974.

[4] G. Hinton. "Connectionist Learning Procedures." *Artificial Intelligence*, **40**, 1, 1989.

[5] A. Jain and D. Hoffman. "Evidence-Based Recognition of Objects." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **10**, 783-802, 1988.

[6] R.P. Lippmann. "An Introduction to Computing with Neural Nets." *IEEE:ASSP Magazine*, April, 1987).

[7] A. Pearce and T. Caelli and W. Bischof. "Rulegraphs for Graph Matching in Pattern Recognition", (In Submission), 1993.

[9] J.R. Quinlan. "Learning Logical Definitions from Relations." *Machine Learning*, **5**, 239-266, 1990.

[10] L. Shapiro and R. Haralick. "Structural Descriptions and Inexact Matching." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **3**, 504-519, 1981.

References

[1] E. Aarts and J. Korst. *Simulated Annealing and Boltzmann Machines.* New York: Wiley, 1989.

[2] Terry Caelli and Andrew Pennington. An improved rule generation method for evidenced-based classification systems. *Pattern Recognition*, 26(5):733–740, 1993.

[3] Narsingh Deo. Prentice-Hall, Inc. N.J., 1974.

[4] G. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40(1), 1989.

[5] Anil K. Jain and Richard Hoffman. Evidence-based recognition of 3-d objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):783 – 801, 1988.

[6] R. P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, April 1987.

[7] Adrian Pearce, Terry Caelli, and Walter F. Bischof. Rulegraphs for graph matching in pattern recognition. *Submitted for publication*, 1993.

[8] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[9] Linda G. Shapiro and Robert M. Haralick. Structural descriptions and inexact matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3(5):504 – 519, September 1981.