Spatial Vision, Vol. 13, No. 2,3, pp. 297–304 (2000)

Learning to recognize objects

WALTER F. BISCHOF*

Department of Computing Science, University of Alberta, Edmonton, Canada
E-mail: wfb@ualberta.ca

Received 10 June 1999; revised 18 February 2000; accepted 14 March 2000

Abstract—Several aspects of systems for learning pattern or object recognition rules are discussed. First, how are recognition rules developed and to what extent is structural pattern information embedded into these recognition rules. Second, how are these rules applied to the recognition of complex patterns such as objects embedded in scenes and how is evidence from different rules combined into a single evidence vector. Third, how can learned rules be improved through performance evaluation and feedback to rule generation stages.

1. INTRODUCTION

In traditional Pattern Recognition, patterns are often represented as vectors of characteristic features which are chosen to optimize representational uniqueness of patterns belonging to different classes. Pattern classification is achieved by partitioning feature space into regions associated with different pattern classes. This approach works well for simple and complete patterns presented in isolation, but is inadequate for complex patterns or for objects embedded in complex scenes.

Alternatively, patterns can be decomposed into constituent parts and described in terms of (unary) features of parts and (binary) features of part relations. These part and part relation features can be linked together into relational structures that define patterns uniquely. Pattern classification is achieved using relational graph matching where a sample pattern is matched to a model pattern by searching for a sample-model mapping that maximizes some objective similarity function. The relational graph matching approach has several drawbacks. First, relational graph matching has an exponential computational complexity and is therefore not feasible for typical object recognition applications. Second, pattern generalization is difficult

*Supported by grant OGP38521 from the Natural Science and Engineering Council of Canada.

to represent and pattern classes have often been represented by enumeration of all instances. Third, matching of partial patterns cannot be dealt with in a satisfactory manner.

In the following, we present several methods for learning and generalizing relational structure descriptions of objects that overcome some of the problems outlined above. We consider three issues regarding the learning of object recognition rules:

(i) Rule generation: How are object recognition rules learned?

(ii) Rule application: How are the rules applied to the recognition of objects?

(iii) Rule evaluation: How is performance in rule application evaluated in order to generate refined recognition rules?

2. RULE LEARNING

We are considering several rule-learning systems that differ with respect to the extent that they use structural pattern information for rule generation. In all cases, rules are defined by regions in unary and/or binary feature spaces. In *attribute-indexed* systems, rules are of the form

```
if attribute1 ∈ bounds1 and attribute2 ∈ bounds2 . . .
then pattern is likely to belong to class c;
```

whereas in *part-indexed* systems, rules are of the form

```
if part $i$ has these attributes
and the relation between part $i$ and part $j$ has these attributes
and part $j$ has these attributes and . . .
then part $i$ is likely to belong to class c,
```

The difference between the two systems is illustrated by the two patterns in Fig. 1. Attribute-indexed systems refer to an unstructured set of unary and binary attributes

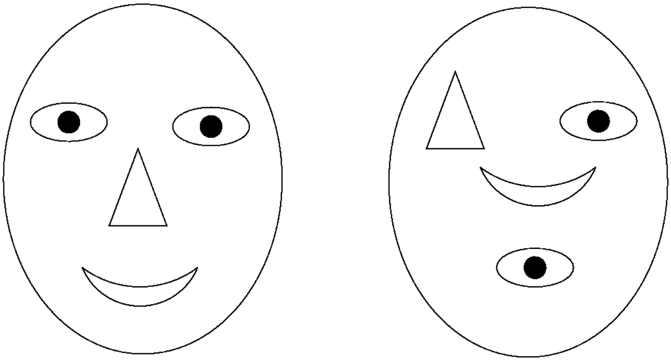

Figure 1. Two patterns that are indistinguishable without correct indexing between unary and binary features, i.e. without part-indexing.

and thus are not able to discriminate between the two patterns. Part-indexed systems, on the other hand, are able to discriminate between the two because they refer to particular part combinations and their attributes.

2.1. Conditional rule generation

The fundamental idea of Conditional Rule Generation (CRG, Bischof and Caelli, 1994) is to generate classification rules for patterns or pattern fragments that include structural pattern information to the extent that is required for classifying correctly a set of training patterns. CRG analyzes unary and binary features of connected pattern components and creates a tree of hierarchically organized rules for classifying new patterns. Generation of a rule tree proceeds in the following manner (see Fig. 2):

First, the unary features of all parts of all patterns are collected into a unary feature space U in which each point represents a single pattern part. The feature space U is partitioned into a number of clusters U_i. Some of these clusters may be unique with respect to class membership and provide a classification rule: If a pattern contains a part p_r whose unary features $u(p_r)$ satisfy the bounds of a unique cluster U_i then the pattern can be assigned a unique classification. The non-unique clusters contain parts from multiple pattern classes and have to be analyzed further. For every part of a non-unique cluster we collect the binary features of this part with all other parts in the pattern to form a (conditional) binary feature space UB_i. The binary

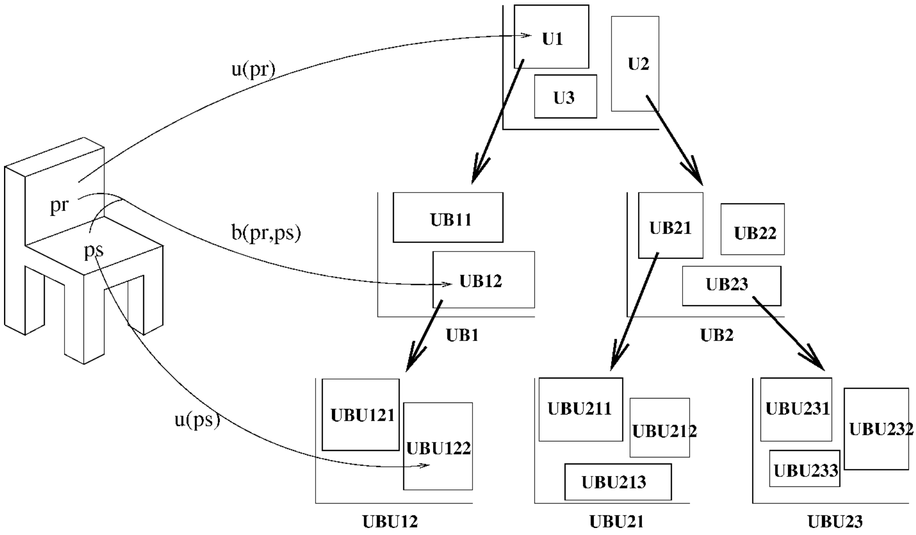

Figure 2. Cluster tree generated by the CRG method. Grey clusters are resolved (i.e. contain elements of a single pattern class). Unresolved clusters (e.g. U_1 and U_2) are expanded to the binary feature spaces (e.g. UB_1 and UB_2), from where clustering and expansion continues until either all rules are resolved or the predetermined maximum rule length is reached.

feature space is clustered into a number of clusters UB_{ij}. Again, some clusters may be unique and provide a classification rule: If a pattern contains a part p_r whose unary features satisfy the bounds of cluster U_i, and there is an other part p_s, such that the binary features $b(p_r, p_s)$ of the pair $\langle p_r, p_s \rangle$ satisfy the bounds of a unique cluster UB_{ij} then the pattern can be assigned a unique classification. For non-unique clusters, the unary features of the second part p_s are used to construct another unary feature space UBU_{ij} that is again clustered to produce clusters UBU_{ijk}.

In the basic form of CRG, rules are generated using a pure splitting method, i.e. simple rules with a predetermined maximum length are generated first, and then are refined into more discriminating rules using an entropy-based splitting procedures where the elements of a cluster are split along feature dimension such that the normalized partition entropy $H_P(T) = (n_1 H(P_1) + n_2 H(P_2))/(n_1 + n_2)$ is minimized, where H is entropy. Rule splitting continues until all classification rules are unique. A completely resolved rule tree provides a set of deterministic rules for classification of patterns. Every rule in the classification tree corresponds to a sequence $U_i - B_{ij} - U_j - B_{jk} - \cdots$ of unary and binary features associated with a chain of pattern parts and their relations. CRG thus produces classification rules for (small) pattern fragments based on unary and binary features.

2.2. *Fuzzy conditional rule generation*

Rule generation in CRG degrades rather quickly with noisy data because it is controlled by a few general parameters such as maximum rule length, minimum feature difference thresholds or minimum entropy thresholds for classification rules. These parameters are typically not sufficient to avoid over-fitting to training data. One way to overcome this problem (McCane and Caelli, 1997) involves replacing the crisp decision boundaries of rule conditions by a fuzzy membership function μ_{ij} such as

$$\mu_{ij} = \frac{1}{1 + \left(\dfrac{d_{ij}^2}{\eta_i} \right)^{1/(f-1)}}, \tag{1}$$

where d_{ij} is the distance between point j and cluster i, f is a fuzzyness factor $(1 < f \leqslant \infty)$ and η_i determines the width of the fuzzy cluster (see McCane and Caelli (1997) for details). Fuzzy classification rules are generated and refined as in the case of crisp rules. Differences arise only at the level of rule application because a chain of pattern parts can match multiple classification rules. Hence measures have to be taken to pursue several, say the best N, classification alternatives rather than all as in the case of crisp rules.

2.3. *Relational evidence theory*

In CRG, all rules are expanded and refined in a depth-first manner until all fragments of all training patterns can be identified correctly. If one wants to be able to recognize objects from almost any fragment, then this is an adequate way to proceed. However, if one is simply interested in good recognition performance for complete objects then these rules are clearly too specific and one may end up with unnecessary over-fitting to training data. Intuitively, what has to be done is to generate and refine recognition rules only to the extent that is required for correct classification of *complete* training patterns.

This is exactly what is being done in the Relational Evidence Approach (Pearce and Caelli, 1997). Classification rules are learned in an iterative scheme consisting of the following stages:

(i) Least-generalization rules involving unary and binary features are generated and refined in a step-wise best-first manner rather than in a depth-first manner as in CRG.

(ii) After every generation/refinement step it is evaluated whether the current rule set is sufficient to uniquely differentiate all training patterns. If this is not the case the rule set is improved with a single expansion/refinement step. Search for an adequate set of rules is found using dynamic programming techniques that rely on heuristic estimates of an upper bound of rule quality.

3. RULE APPLICATION

CRG generates classification rules for (small) pattern fragments. When the classification rules are applied to some new pattern one obtains one or more (classification) evidence vectors for each pattern fragment, and the evidence vectors have to be combined into a single evidence vector for the whole pattern. This is more or less straightforward for single (isolated) patterns, but difficulties arise in scenes composed of multiple patterns where it is unclear whether a chain $p_i - p_j - \cdots - p_n$ of pattern parts belongs to the same pattern or whether it is 'crossing the boundary' between different patterns. We present a heuristic solution in the context of a system that makes only weak and general assumptions about the structure of scene and objects. It is based on the analysis of the relationships within and between instantiated rules (Bischof and Caelli, 1997).

The first stage involves direct activation of the rules in a parallel, iterative deepening method. Starting from each scene part, all possible chains of parts are generated and classified using the CRG rules. The evidence vectors of all rules instantiated by a chain $S = \langle p_1 p_2 \ldots p_n \rangle$ are averaged to obtain the evidence vector $\vec{E}(S)$ of the chain S, and the set $\mathcal{S}_p$ of all chains that start at p is used to obtain an initial evidence vector for part p:

$$\vec{E}(p) = \frac{1}{\#(\mathcal{S}_p)} \sum_{S \in \mathcal{S}_p} \vec{E}(S), \tag{2}$$

where $\#(\mathcal{S})$ denotes the cardinality of the set $\mathcal{S}$. Evidence combination based on (2) does not take into account that some chains may not be contained completely within a single object but ‘cross’ boundaries between objects. These chains are likely to be classified in a arbitrary way, and to the extent that they can be detected and eliminated, the part classification based on (2) can be improved.

The compatibility measure adopted here involves a measure of the compatibility of the evidence vectors of the constituent parts with the evidence vector of the chain. More formally, this measure can be characterized by the following equation. For a chain $S_i = \langle p_{i1}, p_{i2}, \ldots, p_{in} \rangle$:

$$\vec{w}_{intra}(S_i) = \frac{1}{n} \sum_{k=1}^{n} \vec{E}(p_{ik}), \tag{3}$$

where $\vec{E}(p_{ik})$ refers to the evidence vector of part p_{ik}. Initially, this can be found by averaging the evidence vectors of the chains which begin with part p_{ik}. This compatibility measure can be used with a relaxation labeling scheme for updating the part evidence vectors (McCane and Caelli, 1997):

$$\vec{E}^{(t+1)}(p) = \Phi\left(\frac{1}{Z} \sum_{S \in S_p} \vec{w}_{intra}^{(t)}(S) \otimes \vec{E}(S) \right), \tag{4}$$

where Φ is the logistic function, Z a normalizing factor $Z = \sum_{S \in S_p} w_{intra}^{(t)}(S)$, and the binary operator $\otimes$ is defined as a component-wise vector multiplication $[a\ b]^T \otimes [c\ d]^T = [ac\ bc]^T$.

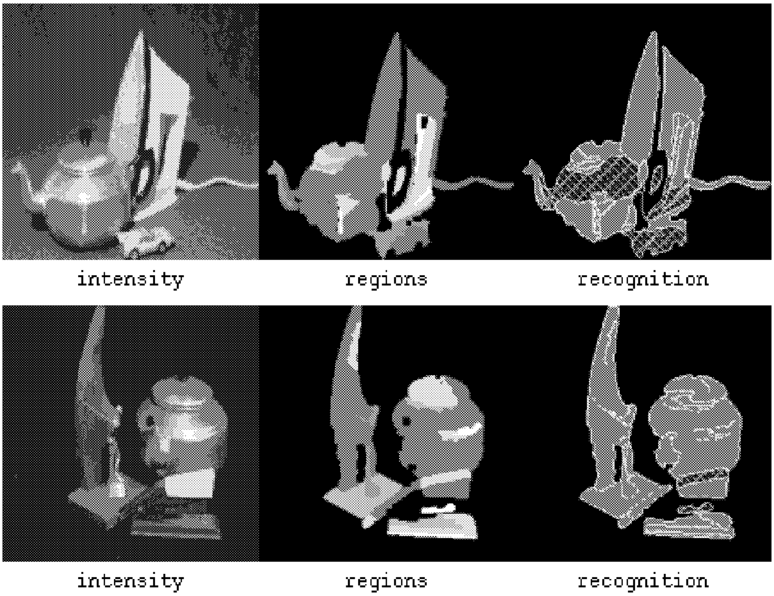

Figure 3. Two scene examples. Left: input images, Center: result of segmentation, Right: interpretation results. Grey regions correspond to correctly labeled regions and hashed ones to erroneous labelings. The classification system was trained with 18 different objects, each presented in isolation from 6 different viewpoints.

For a given chain $S_i = \langle p_{i1}, p_{i2}, \ldots, p_{in} \rangle$ of parts, the updating scheme (4) not only takes into account the compatibility between evidence vectors of all parts p_i but also the compatibility between the average evidence vectors and the chain's evidence vector. Experiments show that classification using (4) leads to very satisfactory classification performance, as is illustrated for two simple scenes in Fig. 3.

4. RULE EVALUATION

The methods described in the previous sections all follow a simple linear scheme: image segmentation → feature extraction → rule generation. The parameters controlling the three stages are fixed *a priori* and are not modified even if performance of the generated rule set is inadequate. This deficiency is addressed in rule evaluation, where the performance of rule sets is tested and the results are used to provide feedback to earlier stages.

In this approach, rule evaluation proceeds in the following way: Classification rules are generated with a training set of images and evaluated using a test set. Feedback from performance analysis is used to improve all stages of the rule generation system, from image segmentation over feature extraction/selection to the rule generation stage. Using best-first search, parameters controlling each of the stages (e.g. resolution level and segmentation threshold at the image segmentation level, feature selection and combination at the feature extraction level, rule length and entropy thresholds at the rule generation level) are optimized. Once performance at one level has been optimized, control parameters are frozen and optimization continues at the next stage. Results show that rule evaluation leads to a significant improvement in classification performance.

5. CONCLUSIONS

Several closely related approaches to rule-based pattern and object recognition were presented, and problems of rule generation, application and evalution were discussed. All rule generation systems included structural pattern information in rules to the extent that is required for classifying all training data. Inclusion of structural pattern information necessitated special considerations regarding the combination of and consistency analysis between rules when applied to scenes containing multiple objects. It was shown that this can be achieved using general rule combination heuristics. Finally, it was shown that classification performance can be improved when results from rule performance analysis are fed back to earlier stages of the rule generation system.

REFERENCES

Bischof, W. F. and Caelli, T. (1994). Learning structural descriptions of patterns: A new technique for conditional clustering and rule generation, *Pattern Recognition* **27**, 1231–1248.

Bischof, W. F. and Caelli, T. (1997). Scene understanding by rule evaluation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**, 1284–1288.

McCane, B. and Caelli, T. (1997). Fuzzy conditional rule generation for the learning and recognition of 3d objects from 2d images, in: *Machine Learning and Image Interpretation*, Caelli, T. and Bischof, W. F. (Eds), pp. 17–66. Plenum, New York, NY.

Pearce, A. and Caelli, T. (1997). Relational evidence theory and interpreting schematics, in: *Machine Learning and Image Interpretation*, Caelli, T. and Bischof, W. F. (Eds), pp. 67–118. Plenum, New York, NY.