Learning Relational Structures: Applications in Computer Vision

Adrian R. Pearce[1], Terry Caelli[1] and Walter F. Bischof[2,3]

Email: adrianp@cs.mu.oz.au, tmc@cs.mu.oz.au,

wfb@psych.ualberta.ca

Fax: +613 348 1184

[1]Department of Computer Science, The University of Melbourne

Parkville, Victoria, 3052, Australia

[2]Department of Psychology BSP-577, The University of Alberta

Edmonton, Alberta,T6G 2E9, Canada

[3]This project was funded by a grant from the

Collaborative Research Centre for Intelligence-Based Systems.

WFB was also supported by a grant from the Natural Science

and Engineering Research Council of Canada.

Abstract

We present and compare two new techniques for Learning Relational Structures (RS) as they occur in 2D pattern and 3D object recognition. These techniques, Evidence-Based Networks (EBS-NNet) and Rulegraphs (RG) combine techniques from Computer Vision with those from Machine Learning and Graph Matching. The EBS-NNet has the ability to generalize pattern rules from training instances in terms of bounds on both unary (single part) and binary (part relation) numerical features. It also learns, the compatibilities between unary and binary feature states in defining different pattern classes. Rulegraphs check this compatibility between unary and binary rules by combining Evidence Theory with Graph Theory. The two systems are tested and compared using a number of different pattern and object recognition problems.

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 (see Shapiro and Haralick, 1981). 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.

In this paper, we discuss a new class of relational learning algorithms which 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 ID3 and FOIL (Quinlan, 1986, 1990) - though the applications in Vision are so specialized that the generic use of those standard symbolic learning technologies is inefficient.

For these techniques, patterns are encoded as vectors in characteristic unary and binary feature spaces which are chosen to optimize representational uniqueness of patterns belonging to different classes and to preserve uniqueness under specific feature transformations. Pattern classification is then achieved by partitioning feature spaces into regions which can maximally evidence individual classes while minimizing the number of partitions (clusters).

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. However, these part and part-relation features have to be linked together into a relational structure (RS) in order to define patterns uniquely.

In relational graph matching pattern classification is achieved when a (new) sample pattern is matched to a model pattern by searching for a label assignment that maximizes some objective similarity function. Pattern classes are usually represented by enumeration of instances and classification is achieved by searching through all model graphs to determine the one producing the best match. Indeed, this representation and the associated graph matching approach - in the form of interpretation trees and feature indexing - has been the preferred architecture for object recognition (Flynn and Jain, 1993).

The relational graph matching approach to pattern and object recognition has several weaknesses. First, the computational complexity is exponential. This is a significant problem since the cardinality of such algorithms is defined by the number of model and sample parts. Second, pattern generalization is difficult to represent. In general, traditional pattern recognition techniques rarely use Machine Learning techniques, and, in particular, they rarely consider generalization of rules corresponding to vertices and edges of relational structures.

Our aim is to combine the RS representation with generalization in two different ways, defined by Evidenced-based Networks(EBS-NNet) and Rulegraphs (RG). In both cases, *rules* are defined by region (volume) bounds in unary and binary feature spaces which are derived to optimally *evidence* different patterns or classes by evidence weights. Such weights are typically derived from the relative frequencies of different classes per region (Jain and Hoffman, 1988) or, more recently, by minimum entropy and Neural Network techniques (Caelli and Pennington, 1993). In addition, it is also necessary to check the consistency between parts and relations associated with rules or generalizations. This is what we term the *label-compatibility* problem and it is related to checking the compatibilities between the instantiations of unary and binary rules.

Other methods for generating RS for visual recognition include, for example, constraint-based decision trees (Grimson,1990), pre-compiled tree generation (Ikeuchi and Kanade, 1988), heuristic search techniques (for example, Bolles and Horaud, 1986), dynamic programming (for example, Fischler and Elschlager, 1973), relaxation labeling (for example, Mohan and Nevatia, 1989) and hierarchical model fitting (for example, Lowe, 1987). However, methods for *generalization* of relational structures have only been addressed sporadically in the literature, such as, for example, by Michalski and Stepp(1983) within the framework of inductive learning of symbolic structural descriptions.

2 The Evidenced-Based Networks: EBS-NNet

Model Construction: To illustrate our current EBS-NNet system we consider the problem of recognizing synthetic (CAD-generated) objects, where the learning of their relational structures (RS) is attained through a finite number of views (see Section 2.3). The input data consisted of view-dependent depth maps. We chose view-dependent input samples in order to restrict the computations of surface curvatures, or pixel labels in general, to what is visible (see Figure 1).

Insert Figure 1 about here

We have used the zero-crossings of the determinant of the Hessian:

$$f_{xx}f_{yy} - f_{xy}^2 \tag{1}$$

as our segmentation procedure - which determines convex, concave and planar regions in a way which minimizes noise amplification which typically occurs when full H and/or K zero-crossings are evaluated (Yokoya and Levine, 1989). Such a segmentation procedure applies equally to models and data and is invariant to rigid motions. Once these parts are extracted then part attributes (unary features) and relations (binary features) can be computed.

2.1 Automatic Rule Generation

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 (for example 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-NNet 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.

For Evidence-Based Systems, rules are defined by regions in feature space which, when activated (triggered), provide weighted evidence for different objects. Consequently, such systems must solve the problem of generating appropriate rules and evidence weights and this inevitably involves the use of clustering algorithms. In this sense, 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:

$$\textbf{IF} \;\; \text{Bounds}_{lower,upper} \; (\text{feature}_1,....,\text{feature}_n)$$

$$\textbf{THEN} \;\; \text{Evidence Weights}(\text{w}_1,...,\text{w}_m)$$

$$\textbf{ELSE} \; \text{no evidence}$$

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). The lower and upper feature bounds define the hyper-rectangle oriented parallel to the feature axes, as illustrated in Figure 2a. Such a structure is useful in so far as it lends itself to either parallel or serial search procedures. Further, since such clusters need not be disjoint, more complex logical definitions of such rules may apply. For example, non-convex regions can be defined logically by rules which include given regions of feature space but explicitly exclude parts.

Insert Figure 2 about here

In previous work, for example, the work of Jain and Hoffman (1988), rules were generated by first clustering the samples in feature space using a minimum spanning tree technique. Our aim was to develop rules and evidence weights which can optimally discriminate objects by deriving clusters and their evidence weights which satisfy different types of cost functions. 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:

$$H_{min} = \min_{i \in j} \left\{ - \sum_{j} \sum_{k} p_{jk} \ln p_{jk} \right\} \tag{2}$$

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. This is a Combinatorial Optimization problem (see Aarts and Korst, 1989) requiring the use of Simulated Annealing where all samples are relabeled for clusters to minimize the entropy function.

2.2 Relational Structures and Evidence Weights

Rather than use relative frequencies of class samples to determine weights (as was used by Jain and Hoffman, 1988), we 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, 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 unary and binary rule. 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 (RS) learning**.

The relationship between rules and objects is formulated by the standard Neural Network equations (Hinton, 1989) and thus constitutes

$$x_i = \sum_j w_{ij} y_j \tag{3}$$

and

$$y_j = \frac{1}{1 + \exp^{-x_j}} \tag{4}$$

which corresponds to the non-linear transducer function. The network is schematically illustrated in Figure 2b.

Weights were first initialized using the relative frequencies of classes within clusters and then optimized by gradient descent (in the back-propagation form (Hinton, 1989)) to maximize recognition performance. That is, in the Supervised Learning Mode, we have estimated weights between input-hidden and hidden-output layers which minimize the difference between observed classification (binary string corresponding to one object node being *activated*(1), the others *not*(0)) and predicted classification from the input binary string $(1, 0, 0, 1..$ etc.) where 1 corresponds to a given rule being activated and 0 to no features being observed within the rule bounds for the given object). This replaces the direct *evidence weights* used by others with an architecture which determines the appropriate weighting of rules by their associated connections within the network.

Again, it should be noted that without full part labeling (without full description of the associations between specific unary and binary features), the matching process is under-determined and the ensemble of features do not uniquely define a model. Albeit, the hidden layer structure of the Neural Network does allow for binding on unary and binary features states from given views of objects and so it does learn a form of RS but without guarantee that specific model parts and relations are encoded in a given hidden unit (see Figure 2). On the other hand, the benefit of this Evidence-Based System lies in it's ability to determine necessary conditions for the presence of specific types of objects. Furthermore, the EBS generates a restricted class of possible matches that can be further analyzed to provide unique classification of patterns.

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 (see Lippmann, 1987).

2.3 3D Object Recognition Example

The EBS-NNet System was tested using a database of 8 objects (see Figure 3): the Pieces database, each with 62 views defined over 30 degree steps in azimuth and elevation of a view-sphere. In addition to these 496 views, an extra 192 new views were generated (24 for each object) where each new view was oriented half-way between the training set views.

Insert Figure 3 about here

Segmentation, feature extraction and rule generation, as described above, was performed on each view of the training data. In this case the unary features were average Mean and Gaussian curvatures, areas, 3D spanning distances, perimeters, boundary curvature and torsion values. The binary features were length of jumps, creases, bounding, centroid and maximum distances, differences in average surface normal angles and average bounding angle between surfaces. Results obtained with new views are shown in Figure 3b.

Perhaps the main limitation of the EBS-NNet approach is that the representation, from an analytic viewpoint, is not unique in so far as rules are generated without *explicitly* considering the relationships between specific unary and binary feature states that define specific objects. That is, both unary and binary rules were generated but not linked together into a label-compatible representation of each model. This was attained *implicitly* via the hidden units in the Neural Network (Figure 2b) where unary and binary feature states occurring in the same view would simultaneously activate one or more hidden units. However, this process, again, does not *guarantee* a unique representation of structural relations in the data.

Two *sources* of generalization emerge from this perspective. The first relates to the issue of training the system with the representative views to enable the development of fully view-independent rules. As already discussed, generalization of this type is dependent upon the nature of the object and surface views. For example, it is impossible to generalize *back* views of objects when training using only *front* views if the back views have no relation to the fronts! The second type of generalization refers to the extent to which the rules cover large volumes of feature space which is not densely represented by sample data (see Figure 2). Ideally, these two aspects of generalization fit together - but this is not always the case in ORS. Using an Object Recognition System (ORS), the best generalization from *given* data is achieved using invariant features, good rule generation and evidence weight estimation techniques. What is needed, however, is an objective way of *proving*, a priori, what constitutes the minimum number of views for ORS. One important characteristic of EBS is that they also determine the types of non-rigid deformations of objects which would be permitted for a given object class. Such deformations are determined by the bounds on specific features as generated from the training data.

It does, however, provide an objective definition of the *difficulty* of a object recognition problem via the entropy values of each rule and evidence weight statistics. Clearly, rules which cover the largest volumes of feature space with the least entropy are to be preferred over solutions with high entropy, the latter demonstrating the need for more rules with smaller, less general regions in feature space.

3 The Rulegraph System

Although the EBS-NNet systems encode some relational structure (RS) information in the hidden layers of the Neural Network (see Figure 2), 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. Such graphs have weighted vertices and edges in the form of class evidence vectors.

In the Rulegraph system, we use rule evidence weights together with *explicit* label-compatibilities to prune the search space in graph matching. The evidence weights are not used to *combine* unary and binary feature states - as occurs in the EBS-NNet system - but rather, the technique relies on two separate processes.

First, we reduce the cardinality of the graph matching problem by replacing the original Relational Structure (RS) of each model (specific parts and relations) by a graph of rules (Rulegraph). Each unary rule (Rulegraph vertex) and binary rule (Rulegraph edge) has a class evidence vector determined from the relative frequencies of class samples in such regions - as used by Jain and Hoffman (1989).

Second, we search for subsets of compatible labels between rules. However, this is constrained using evidence weights produced by an clustering procedure. 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.

——————————

Insert Figure 4 about here

——————————

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. In Figure 4a, two rulegraph models are shown which represent the training patterns for class 1 and class 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.

3.1 The Label Compatibility Checking Method

At Recognition time, compatibility of a sample rulegraph and a model rulegraph is is checked as follows. After activation of unary rules, a modified existence check is carried out for each pair of mapping states for R_i^u and R_j^u. The (multiple) mapping states are then updated by instantiation (if the label is *not* yet mapped) or elimination (if the label *is* mapped) using the mappings generated from the new existence checks and the old mapping states. The mapping states are further updated in order of decreasing evidence weights of rules into which they map (since several binary rules can exist between two unary rules). This ensures that labels which have strongest evidence for a particular class will be mapped first. Finally, we check that at least one binary rule is satisfied which links the target unary rules and associated labels (U_i,B_{ij},U_j).

The Label Compatibility Checking Method offers a technique for checking compatibility between two individual rules. The problem of finding the *best* match now reduces to that of finding the largest evidenced set of rules which are all pairwise compatible - a *clique*. 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. Furthermore, 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 a simple evidence weight metric to allow probabilistic pruning of the search tree.

The current match is obtained by evaluating the evidence weight for the rules in the current clique of compatible rules and the upper bound of potential match possible can be calculated based on the (optimistic) assumption that all presently compatible rules turn out to be compatible with one another. For example, the sample in Figure 4c activates rules R_1^u, R_2^u and R_1^b. Initially, evidence for the classes is calculated simply using all active rules - since the cliques are all empty.

An initial queue of active rules from all classes is constructed based on weights from activated (instantiated) rules for each rulegraph model, and the queue is sorted in decreasing order of weights. This is a (relational) numerical analogue to the literal expansion process used in FOIL (see Quinlan, 1990). The queue contains rulegraph interpretations from *all* classes of data and is searched simultaneously. This maximizes the pruning effect of A* search by only extending those cliques which have the highest potential for being the best match. This results in alternate classes being examined during the course of the search. A clique is extended with rules in decreasing order of their evidence weights, thus ensuring that the sample parts are first assigned to the model parts to which they most likely correspond.

The best match has been found when the clique at the head of the queue cannot be further extended. The queue order guarantees that extensions of cliques further down the queue cannot possibly yield a better match. The result of such a search for the sample in Figure 4c is shown in Figure 4d. Here the best *rulegraph interpretation* for the sample is shown in terms of the rulegraph model for class 1. The system produces an overall evidence weight for the interpretation corresponding to the likelihood of the sample coming from the class (For further details see Pearce, Caelli and Bischof, 1993).

4 Comparison of EBS-NNet and Rulegraphs

We have compared classification performance and complexity of Rulegraph Matching to that of EBS-NNet and that of Traditional Subgraph Isomorphism using Branch-and-Bound for the patterns shown in Figure 5. For the training set (TS), four segments (parts) were extracted from each of the 15 patterns (see Figure 5b). Similarly, four different segments were extracted from each of the 15 patterns for the test sample set (SS). This scheme of pattern sampling simulates occlusion and data loss and is consistent with sampling regimes found in 3D-Object Recognition and other complex Pattern Recognition problems where only partial data is available in any given sample. The features extracted from the segments were: Unary - perimeter and colour and Binary - distance between centers and sum of distance between corners. 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 moved the corners, colour and position of the polygons relative to the class from which they were sampled (see Figure 5c).

Only adjacent and non-overlapping edges were generated for the binary features. Graphs with different numbers of vertices were generated by varying the numbers of parts, but in all cases the SS consisted of 60 different samples over which performance was averaged. The data is not guaranteed to be perfectly classifiable and exhibits many characteristics fundamental to problems encountered in Pattern Recognition.

Insert Figure 5 about here

For RG's, the rule generation scheme used a nearest neighbor clustering method (Leader clustering, see Hartigan, 1975) 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 and possibly overlapping - 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. Note that, for Rulegraphs, we do not combine unary and binary rules/features in the training phase.

For comparison purposes, the Evidence-Based System (EBS-NNet, Section 2) with one hidden layer was used. The number of nodes was equal to the maximum of the input (number of unary and binary rules) or output (number of classes) - whichever was larger. Backpropagation was used to minimize the error in the network and 1000 sets of training epochs were used over several different learning rates and the best performing trained network were used for classification. For further comparison, tests were performed using Traditional Subgraph Isomorphism matching and using the same relative frequency evidence weights that was used in Rulegraph Matching. This was done in order to find the best possible classification for each data set. For Subgraph Isomorphism, a depth first search strategy was used utilizing Branch-and-Bound (SI-BB) which constrains the search when it is not possible to reach a better match result via extension of the current interpretation. Depth-first search of this type is typically preferred for problems of high cardinality since Breadth-first search can lead to exponential space requirements. Indeed, without the use of Branch-and-Bound we would not have been able to obtain exhaustive search results.

Classification performance using the occluded and distorted Sample Set best classification performance was achieved with between five and fifteen rules (see Figure 6a). It can be seen, here, that the best classification performance for the Rulegraph Matching (88 percent) is considerably better than for the EBS-NNET (55 percent) and it is almost as high as is possible using Traditional Subgraph Isomorphism (SI-BB) (90 percent).

————————————

Insert Figure 6 about here

————————————

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 can also compare the computational complexity of the different methods. For the case of the EBS-NNet the complexity is determined by the number of weight additions at each node of the network during the feed-forward operation. For a fully connected network the complexity is $O(n^2)$ with n being the number of nodes in the hidden layer, provided, of course, that this number is larger than the sum of unary and binary features activating the rules. For both SI-BB and Rulegraph Matching, the complexity is determined by the total number of operations which compare a single edge in the model with respect to the existence of a single label-compatible edge in the sample. As a result, we have expressed the computational cost of SI-BB and Rulegraphs in terms of existence checks. For comparison purposes, we equate an existence check of Rulegraph Matching and SI-BB with one weight addition in EBS-NNet.

Results for the average computational cost for the same Blocks Sample Set (SS) are shown in Figure 6b. It is apparent that Label Compatibility Method used by the Rulegraph Matching system requires only a fraction of the operations required by Traditional Subgraph Isomorphism (SI-BB). In fact, it is close to the number of operations required by the EBS-NNet. The numbers of existence checks was consistent with the observed run times. EBS-NNet and Rulegraph Matching system matched nearly instantaneously while SI-BB consumed large amounts of computation time. In terms of the worst case complexity, Traditional Subgraph Isomorphism is determined by the number of vertices, v and is $O(2^{v/3})$ (see Tarjan and Trojanowski for details). In Rulegraph Matching, the cardinality of the search is reduced to the number of rules, r, resulting in worst case complexity of $O(e^2 2^{r/3})$ for e edges.

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.

In summary, the results indicate that the rulegraphs offer a classification performance close to the obtainable optimum and a significant improvement over Evidence-Based Systems, in particular for occluded and distorted data. The computational complexity of the rulegraph method is much lower than that of Subgraph Isomorphism (using Branch-and-Bound) and similar to that of the Neural Network.

5 Discussion

In this paper, we have described some methods for developing prototypical description which involve the definitions of parts and their 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.

In both systems the heads of rules are defined by bounded conjunctions of attributes (as in AQ11 - see Michalski and Stepp (1983)) and evidence for the same class can occur in many regions of the same or different attribute spaces of different arities. The main differences, however, are in how each system encodes, and tests for, label compatibilities between rules and data. In the EBS-NNet system the emphasis is on generating rules which can filter evidence for the possible existence of given objects from the presence of specific unary and binary features. The Rulegraph method essentially checks whether there is an object or pattern in the data which not only satisfies the EBS constraints, but which also has the *specific* co-occurrences of unary and binary feature states.

6 References

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

P. Besl and R. Jain. "Segmentation through variable-order surface fitting." *IEEE Trans. Pattern Anal. Machine Intell.*, **10**, 167-192, 1988.

P. Besl and R. Jain. "Invariant Surface Characteristics for 3D Object Recognition in Range Images." *Computer Vision, Graphics and Image Processing*, **33**, 33-80, 1986.

P. Besl. "Geometric Modeling and Computer Vision." *Proceedings of the IEEE* , **76**, 936-958, 1988.

R. Bolles and P. Horaud. "3DPO: A three-dimensional part orientation system." *The International Journal of Robotics Research*, **5**, 3-26, 1986.

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

T. Caelli and A. Dreier. "Variations on the Evidenced-Based Object Recognition Theme." *Proceedings of the eleventh IAPRA International Conference on Pattern Recognition*, 450-454, 1992.

T. Caelli and A. Dreier. "Variations on the Evidenced-Based Object Recognition Theme." *Pattern Recognition*, (In Press), 1993.

T. Caelli and A. Pennington. "An improved rule generation method for evidence-based classification systems." *Pattern Recognition*, vol 26, 5, 733-740, 1993.

M. do Carmo. *Differential Geometry of Curves and Surfaces.* Englewood Cliffs: Prentice-Hall, 1976.

T. Fan, G. Medioni and R. Nevatia. "Segmented Descriptions of 3-D Surfaces." *IEEE Journal of Robotics and Automation*, **RA-3**, 527-538, 1987.

T. Fan, G. Medioni and R. Nevatia, "Recognizing 3-D Objects Using Surface Descriptions." *IEEE Trans. Pattern Anal. Machine Intell.*, vol 11, 11, pp.1140-1157, 1989.

U. Fayyad and K. Irani. On the Handling of Continuous-Valued Attributes in Decision Tree Generation. *Machine Learning*, 8, 87-102 (1992).

M. A. Fischler and R. A. Elschlager, The representation and matching of pictorial structures, *IEEE Transactions on Computing*, **22**, 67-92 (1973).

P. Flynn and A.K.Jain. 3D Object Recognition Using Invariant Feature Indexing of Interpretation Tables. *Computer Vision, Graphics, and Image Processing*, 55(2): 119-129 (1992).

P. Flynn and A. Jain. Three-Dimensional Object Recognition. In Tzay Y. Young(Ed.) *Handbook of Pattern Recognition and Image Processing, Volume 2: Computer Vision.* New York: Academic Press (1993).

W.E.L. Grimson. *Object Recognition by Computer.* MIT Press (1990).

K. Ikeuchi and T. Kanade. Automatic Generation of Object Recognition Programs. *Proc. IEEE*, 76(8):1016-1035 (1988).

J. Hartigan. *Clustering Algorithms.* New York: Wiley, 1975.

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

R. Hoffman and A. Jain. "Segmentation and Classification of Range Images." *IEEE Trans. Pattern Anal. Machine Intell.*, **9**, 608-620, 1987.

A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data.* Englewood Cliffs: Prentice

Hall, 1988.

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

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

D.G. Lowe. "Three-Dimensional Object Recognition from Single Two-Dimensional Images." *Artificial Intelligence*, **31**, 355-395, 1987.

R. Michalski and R. E. Stepp. "Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **5**, 396-409, 1983.

R. Mohan and R. Nevatia. "Using perceptual organization to extract 3-D structures." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 1121-1139, 1989.

S. Muggleton and W. Buntine. "Machine invention of first-order predicates by inverting resolution." *Proceedings of the Fifth International Conference on Machine Learning*, 339-352. San Mateo: Morgan Kaufmann, 1988.

A. Pearce and T. Caelli and W. Bischof. "Rulegraphs for Graph Matching in Pattern

Recognition", Technical Report, Department of Computer Science, The University of Melbourne, Parkville 3053, Australia. 1993.

J.R. Quinlan. "Induction of Decision Trees." *Machine Learning*, **1**, 81-106, 1986.

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

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

R. Tarjan and A. Trojanowski. "Finding the maximum independent set." *SIAM Journal of Computing*, bf 6, 537-546, 1977.

N. Yokoya and M. Levine, "Range Image Segmentation Based on Differential Geometry: A Hybrid Approach." *IEEE Trans. Pattern Anal. Machine Intell.*, **11**, 643-649, 1989.

7 Figure Captions

Figure 1: Shows segmented range data in terms of convex, concave and planar region types (from zero-crossings of Gaussian curvature). Surfaces were smoothed by an isotropic Gaussian filter with $\sigma = 6$ pixels before determining the derivatives (and the determinant of the Hessian, see Eqn. 1; From Caelli and Dreier, 1993).

Figure 2: a) 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 clustering, see Hartigan, 1975), and minimum entropy (right) clustering solutions. Here there are three classes and four clusters. b) 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 Caelli and Pennington, 1993).

Figure 3: **Top:** Shows one example view for each of the objects in the Pieces database. There were 62 views per object in the training views set of images. **Bottom:** Shows the performance on the Pieces database using the Entropy clustering technique. The classification results are based on the (new) test images. (From Caelli and Dreier, 1993).

Figure 4: 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). At run time, parts in the sample pattern activate unary and binary rules based on their feature states as shown shown in (c). The search for label-compatible rules between the sample and the model results in rulegraph interpretation (best match) as is seen in (d).

Figure 5: 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) segments are shown in (c). The features extracted from the segments were: Unary - perimeter and colour and Binary - distance between centers and sum of distance between corners.

Figure 6: Classification performance is shown for different numbers of rules for the distorted and occluded Sample Set (SS) in (a). A comparison of the average case computational complexity expressed in $\log_2$ of the total number of existence checks required to find best match and is shown for the different systems using the Sample Set (SS) from the Blocks data for different number of parts is shown in (b).

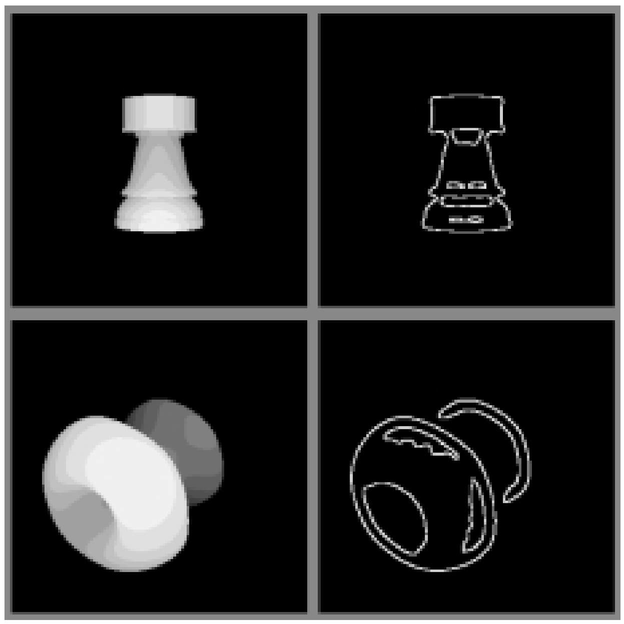
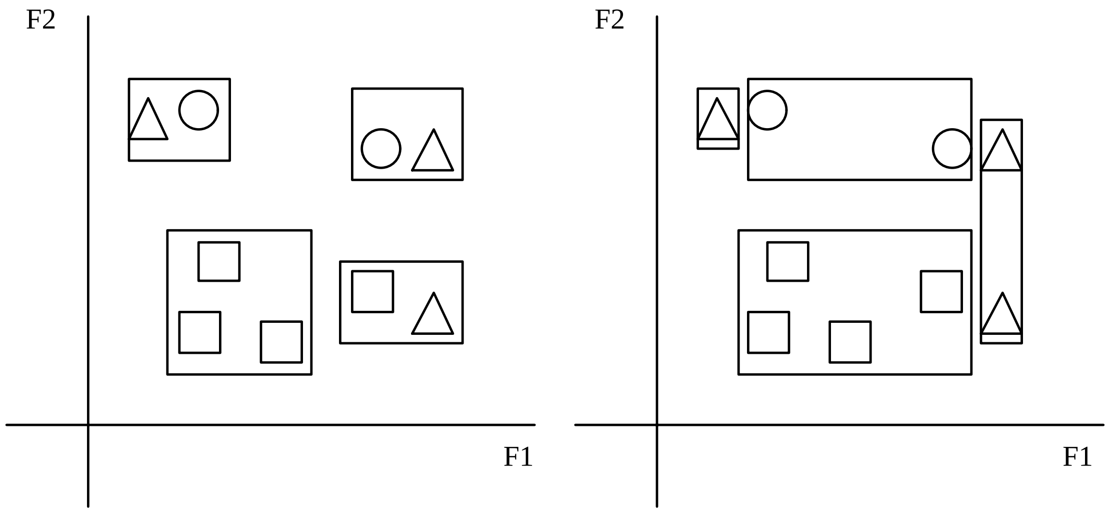


Figure 1:

(a)


F2
F1
F2
F1

(b)

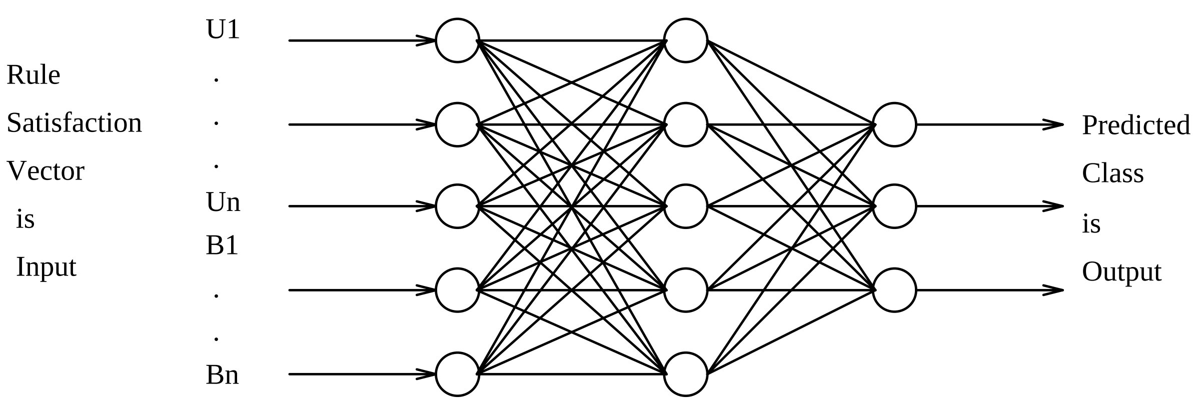

Rule
Satisfaction
Vector
is
Input
U1
.
.
.
Un
B1
.
.
Bn
Predicted
Class
is
Output

Figure 2:

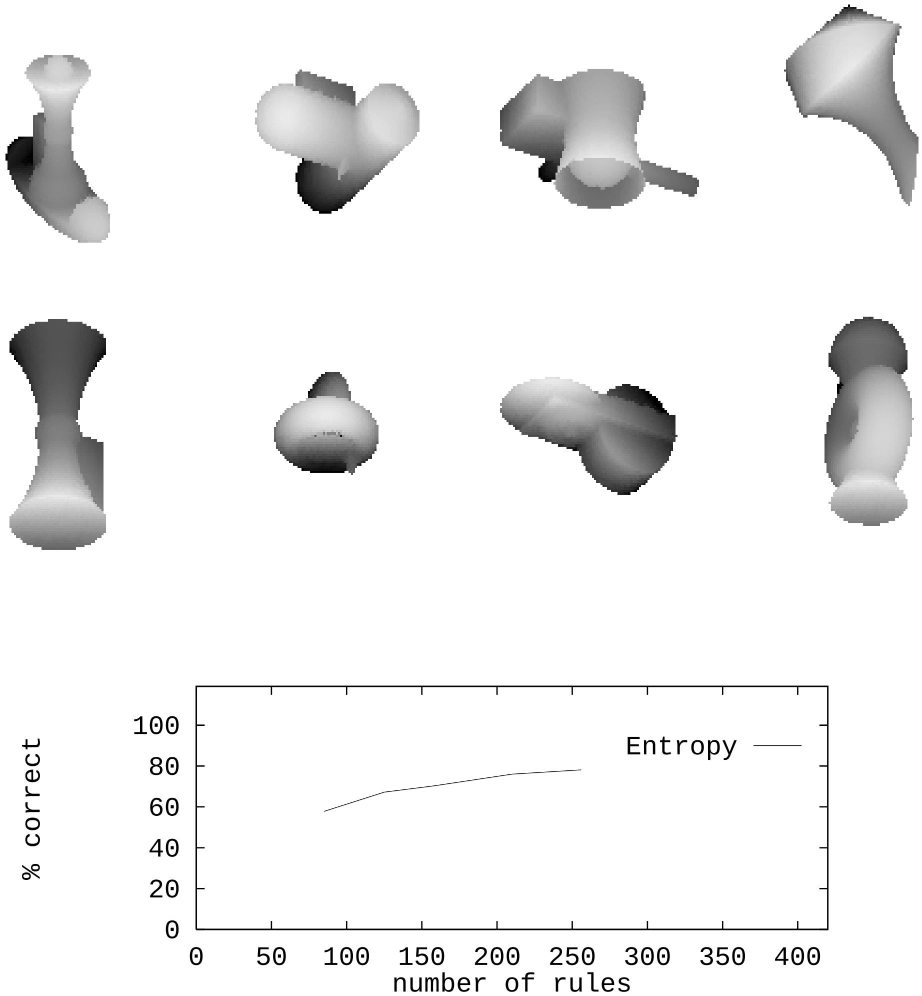

Entropy
% correct
number of rules
0
20
40
60
80
100
0
50
100
150
200
250
300
350
400

Figure 3:

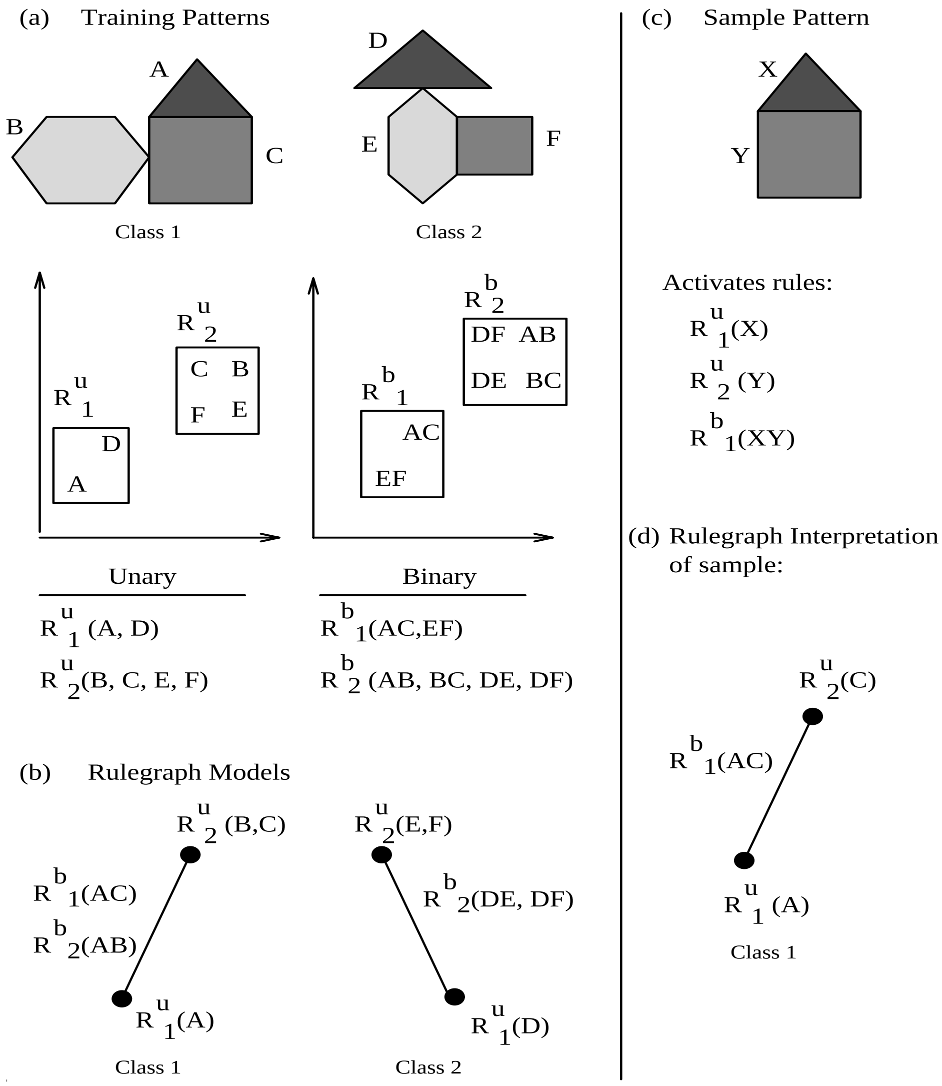
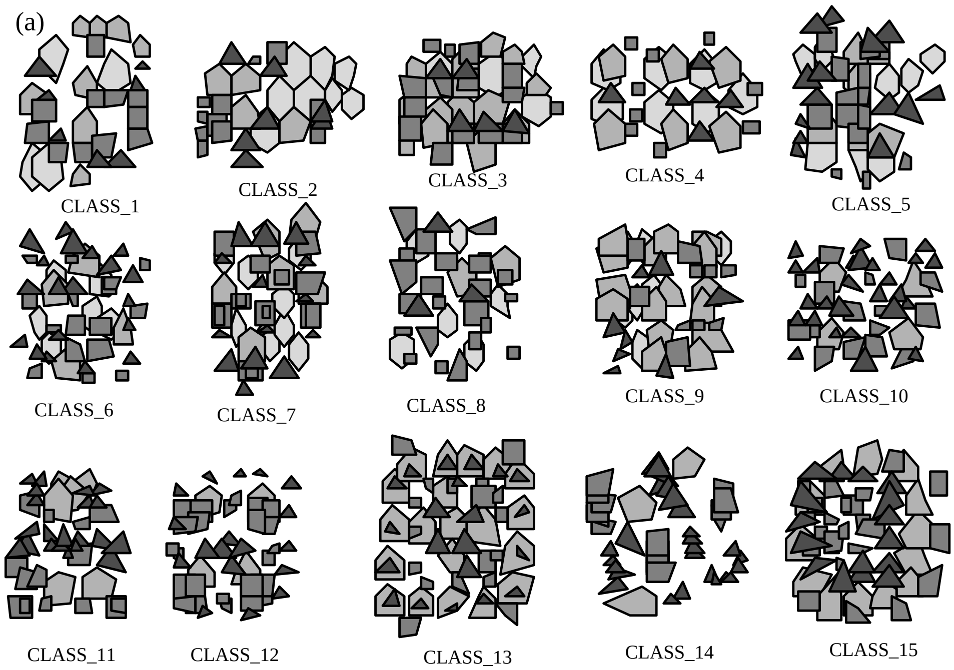
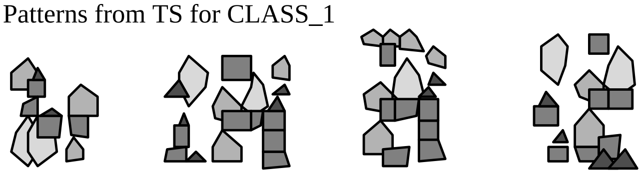

(a) Training Patterns
A
B
C
D
E
F
Class 1
Class 2
R u 2
C B
F E
R u 1
D
A
R b 2
DF AB
DE BC
R b 1
AC
EF
Unary
R u 1 (A, D)
R u 2(B, C, E, F)
Binary
R b 1(AC,EF)
R b 2 (AB, BC, DE, DF)
(b) Rulegraph Models
R u 2 (B,C)
R b 1(AC)
R b 2(AB)
R u 1(A)
Class 1
R u 2(E,F)
R b 2(DE, DF)
R u 1(D)
Class 2
(c) Sample Pattern
X
Y
Activates rules:
R u 1(X)
R u 2 (Y)
R b 1(XY)
(d) Rulegraph Interpretation of sample:
R u 2(C)
R b 1(AC)
R u 1 (A)
Class 1

Figure 4:



(b)Training Patterns from TS for CLASS_1



(c) Sample patterns from SS for CLASS_1

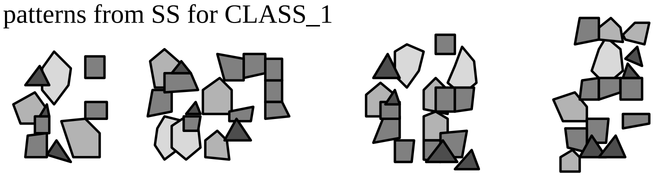
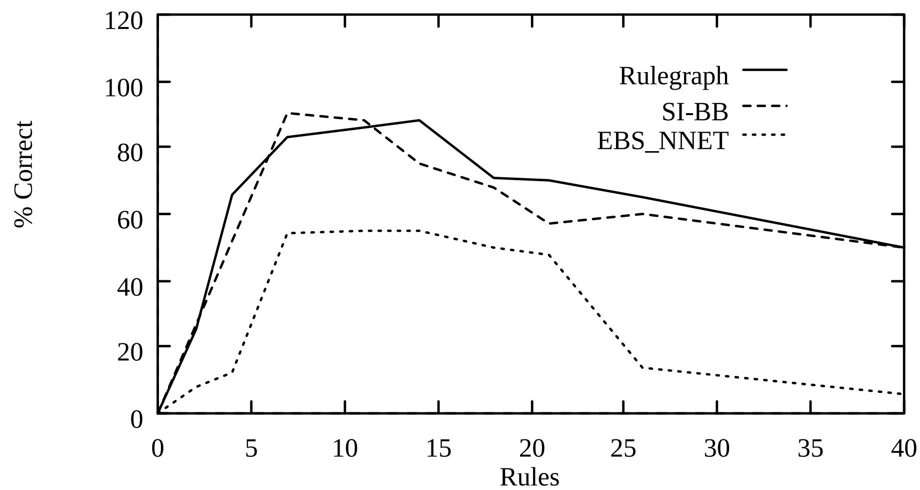


Figure 5:

(a) Blocks SS - 20 Parts



(b) Blocks SS - 20 Rules

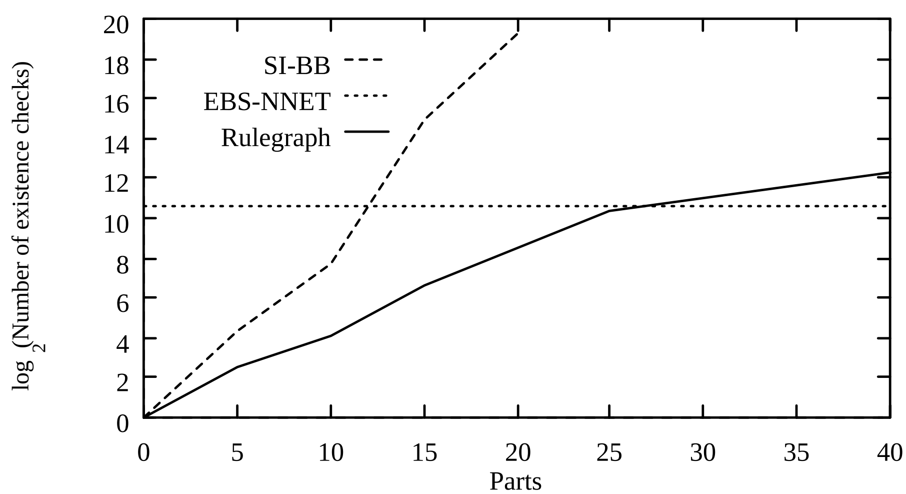


Figure 6: