A Connectionist Approach to Structural Similarity Determination as a Basis of Clustering, Classification and Feature Detection

Kristina Schädler, Fritz Wysotzki
Technical University of Berlin
FR 5-8, Franklinstr. 28/29, D-10587 Berlin, Germany
Phone 0049-30-314 25491, e-mail: schaedle@cs.tu-berlin.de

Abstract. Many algorithms in machine learning, knowledge discovery, pattern recognition and classification are based on the estimation of the similarity or the distance between the analysed objects. Objects with higher structural complexity often cannot be described by feature vectors without losing important structural information. These objects can adequately be represented in the language of logic or by labeled graphs. The similarity of such descriptions is difficult to define and to compute. In this paper, a connectionist approach for the determination of the similarity of arbitrary labeled graphs is introduced. Using an example from organic chemistry, the application of the approach within one distance based and one generalisation based classfication algorithm is demonstrated. The generalisation based algorithm forms clusters or subclasses of similar examples of the same class and extracts the parts of the objects which determine the class of the object. The algorithms perform very satisfactorily in comparison with recent logical and feature vector approaches. Moreover, being able to handle structural data directly, the algorithms need only a subset of the given features of the objects for classification.

1 Introduction

Many tasks in intelligent data analysis require the estimation of the similarity between the entities of the data base and the query objects. In some methods of clustering, the data set is divided into clusters with high intra-class and low inter-class similarity. In classification, distance based algorithms use the similarity between classified objects and the query object for determining the class of the query object. In machine learning and knowledge discovery [30, 31], objects are compared with the aim to find common characteristic features of objects having similar properties or belonging to the same class. If the objects are represented by feature vectors, often some kind of Euclidian or generalised Minkowski metric is employed.

Objects having a complex structure often cannot be described as fixed length feature vectors without losing important structural information (see [15, 16, 17]). In machine learning, these objects often are represented in a logical framework (see [1, 26] for an overview). Inductive Logic Programming is used to tackle the problems of learning and classification. In many real world applications, it is more natural to describe complex objects or other structures by labeled graphs,

for instance chemical structures by structural formulas or computer programs by trees or flow charts, respectively. In this paper, the estimation of the similarity between graphs is discussed and a connectionist approach for computing the similarity of two graphs is introduced. It is a neural net approach to the graph theoretic problem of subgraph isomorphism. It is able to find an approximate solution of the NP-complete graph matching problem efficiently including domain knowledge about the similarity of objects. For more details about the algorithm and its application to case-based reasoning, see [33, 32]. The approach decribed in this paper has been employed in two classification algorithms. The first one is a new *similarity-based inductive graphtheoretic* learning algorithm called SIG-Learning which reduces the set of given instances to a smaller set of prototypes used for classification. The second algorithm performs a weighted nearest neighbor classification. The results of the two algorithms are compared with those produced by some recent logic, graphtheoretic and feature vector based algorithms, applied to the discovery of cause-effect relationships of some organic compounds (mutagenesis data), a typical data mining problem.

2 The Concept of Structural Similarity

In contrast to objects represented by feature vectors, no appropriate mapping of graph representations into the Euclidian vector space of real numbers exists. Thus the similarity of graphs cannot be determined using the Euclidian or a general Minkowski metric. Caused by the growing interest in relational descriptions, different measures of similarity of relational descriptions have been proposed in the last years. Only a few of them have metric properties. In general the computation of the similarity of relational objects requires a lot of effort because every measure is based on the (NP-complete) search of some best mapping between the objects (see [4, 3, 5, 25, 27]). Subgraphs are used in the similarity detection of cases, for instance in [37, 6, 36], where the notion of strucutral similarity is defined as the largest isomorphic subgraph of two graphs. Shapiro and Haralick defined in [8, 35] a structural difference of relational descriptions with metric properties for graphs of the same size. The well-known Dice and the Tanimoto-Coefficient for feature vector representations can also be adapted for graph representations. In [10], Emde and Wettschereck propose a distance-based learning algorithm using a recursive similarity measure for relational structures described by predicate logic.

The similarity measure used in this paper is based on these concepts and the ZELINKA-Metric [42] and its derivates [22], where the common parts of the structures are related to the number of nodes and edges of the smaller of the two graphs. Colored graphs are described in this paper by $G(N, V, l, e, L, E)$, where N and $V = N \times N$ [1] are the nodes and edges of the graph, respectively, L and E some arbitrary sets of colors or labels and $l : N \rightarrow L$ and $e : V \rightarrow E$ the coloring functions for the nodes and edges of the graph. Depending on the task,

[1] That means all graphs are considered complete graphs.

a similarity measure based on the following definition is chosen:

$$sim_a(x,y) = \frac{|(n_i,n_j) \in V_x : e_x(n_i,n_j) = e_y(\varphi(n_i),\varphi(n_j))|}{|N_x| * (|N_x| - 1)} \quad (1)$$

where φ is the chosen mapping between nodes of the graphs
$x = G(N_x, V_x, l_x, e_x, L_x, E_x), N_x = \{n_1, n_2, ..., n_{|N_x|}\}$ and
$y = G(N_y, V_y, l_y, e_y, L_y = L_x, E_y = E_x)$. It is assumed that φ is a graph morphism that maps a node only onto a node with the same label and provides a bijective partial mapping between the nodes of the graphs. So the measure relates the number of corresponding relations (matching edges) between corresponding nodes of the two graphs to the number of these relations when graph x is mapped onto itself, or in short: The value of $sim_a(x,y)$ measures the fraction of edges of x contained in y. In general, φ is chosen in such a manner that it gives the best mapping between y and x with respect to the similarity in Eq.(1). As it can be seen, the measure is an asymmetrical one. $sim(x,y)$ can be defined as $sim_a(x,y)$ or $sim_a(y,x)$ depending on the task and the properties of the graphs involved. In nearest neighbor classification as described in Section 5 the symmetric definition $sim(x,y) = \left\{ \begin{array}{c} sim_a(x,y) \text{ if } max(|N_x|,|N_y|) = |N_x| \\ sim_a(y,x) \text{ otherwise} \end{array} \right\}$ turned out to be the best choice. The similarity measure used in the prototype classification in Section 4 reflects the asymmetry of the generalisation/specialisation relation between the generalised description P of a set of examples and the single example x. It is given by the formula

$$sim_l(P,x) = \frac{|(n_i,n_j) \in V_x : e_x(n_i,n_j) = e_P(\varphi(n_i),\varphi(n_j))|}{|N_x| * (|N_x| - 1)} \in [0,1] \quad (2)$$

during learning where the instance covered best by the current prototype has to be determined and

$$sim_t(P,x) = \frac{|(n_i,n_j) \in V_x : e_x(n_i,n_j) = e_P(\varphi(n_i),\varphi(n_j))|}{|N_P| * (|N_P| - 1)} \in [0,1] \quad (3)$$

in overgeneralisation testing and classification where the question is asked whether the instance x belongs to the set of instances containing P as a subgraph or not. The similarity measure (3) is used here - similar to testin θ-subsumption restricted to injective mappings - as a asymmetric measure of the quality of the covering of x by P, i.e. in terms of graph matching, the containment of P as a subgraph that generalises the set of all graphs containing P. In generalising two examples x_1 and x_2 or an example x and a generalised description P, a new graph Q is constructed that contains the nodes and edges of the common subgraph of x_1 and x_2 or of x and P, respectively, defined by the mapping φ.The nodes which are not mapped onto nodes of the other graph are omitted. The edges where $e_{x_1}(n_i,n_j) \neq e_{x_2}(\varphi(n_i),\varphi(n_j))$ or $e_x(n_i,n_j) \neq e_P(\varphi(n_i),\varphi(n_j))$ are generalised by edges with the color "don't know" (or, in more sophisticated versions of the algorithm, some generalisation of the colors $e_{x_1}(n_i,n_j)$ and $e_{x_2}(\varphi(n_i),\varphi(n_j))$ or $e_x(n_i,n_j)$ and $e_P(\varphi(n_i),\varphi(n_j))$, respectively) in Q.

These generalised edges are treated in two different ways during the computation of the similarity between Q and another example y. If the best mapping of Q into the example y is computed, for instance in the classification of y or

in overgeneralisation testing, the "don't know"-color is considered equal to all other colors in E_y. If a new generalisation of Q and y is to be constructed, the "dont-know"-edges are considered not equal to edges of y with another color or at least less similar than edges of Q with the same color like the edge in y under consideration.

3 The Computation of Graph Similarity

If the similarity of graphs with respect to Eq. (1) is to be determined, a mapping φ must be found that maximizes this similarity between the graphs. Replacing φ by a corresponding relation $\rho \subseteq N_x \times N_y$ between nodes of the graphs with the same or similar labels and allowing ambiguous mappings between the graphs, the graph matching problem can be transformed into an optimization task where matches between pairs of nodes $(n_i^x, n_j^y) \in \rho$ with $e_x(n_i^x, n_j^x) = e_y(n_k^y, n_l^y)$ for related nodes $(n_i^x, n_k^y) \in \rho, (n_j^x, n_l^y) \in \rho$ are rewarded by a positive weight w in order to maximize the numerator in Eq. (1) where the denominator remains constant. On the other hand a penalty $-w_I$ is introduced to mappings of the same node of one graph to more than one node of the other graph. The graph matching problem can then be reformulated as follows:

$$\rho^*(x,y) = max_{\rho \subseteq N_x \times N_y} \{w * c1_\rho - w_I * c2_\rho\} \tag{4}$$

$$c1_\rho = |\{(\rho(n_i^x, n_k^y), \rho(n_j^x, n_l^y)) | e_x(n_i^x, n_j^x) = e_y(n_k^y, n_l^y)\}| \tag{5}$$

$$c2_\rho = |\{(\rho(n_i^x, n_k^y), \rho(n_j^x, n_l^y)) | (n_i^x = n_j^x \wedge n_k^y \neq n_l^y) \vee (n_i^x \neq n_j^x \wedge n_k^y = n_l^y)\}| \tag{6}$$

The mapping ρ can be represented by a twodimensional array $o_{[N_x, N_y]}$ where $o_{ik} = 1$ iff $\rho(n_i^x, n_k^y)$ and $o_{ik} = 0$ otherwise. So, the objective function of the optimization tasks is

$$f(o^*) = max_{o \in \{0,1\}^{|N_x|} \times \{0,1\}^{|N_y|}} \sum_{i,j=1}^{N_x} \sum_{k,l=1}^{N_y} w_{ij,kl} o_{ik} o_{jl} \tag{7}$$

where $w_{ij,kl} = \left\{ \begin{array}{c} w \\ -w_I \\ 0 \end{array} \right\}$ according to Eq.(4). It has been shown ([20, 19, 2, 21, 28, 7, 14, 39]), that such quadratic optimization tasks can be solved by Hopfield-like Artificial Neural Nets, i.e. bi-directional associative memories. In contrast to the most other implementations which use Hopfield Nets with binary or continuous sigmoidal output function, in this work an approach described in [11, 12, 41] is used where the output of the neurons is restricted by the non-differentiable ramp function $r(x) = \left\{ \begin{array}{c} 0 \text{ if } x < 0 \\ x \text{ if } 0 \leq x \geq 1 \\ 1 \text{ if } x > 1 \end{array} \right\}$. In addition the units of the net receive a part of their own output as an input, setting the diagonal of the connection matrix to a weight $1 > w_d > 0$. As usually in the domain of artificial neural nets, the net's results depend strongly on the chosen parameters. In [34], for the first time an extensive analysis of the approach of Feldman [11, 12] and Wysotzki [41] is given that describes the algorithm and provides its theoretical foundations, delivering the parameter settings which guarantee that the net reaches a stable

state that represents a good solution of the problem. Experiments have shown that the algorithm approximates the optimum solution within $O((|N_x| * |N_y|)^2)$ time which can be accelerated considerably using parallel hardware. Based on these results, the approach has been incorporated into two different classification algorithms.

4 The SIG-Learner and the Prototype Classifier

4.1 Relational Learning Algorithms

Learning tasks for structured objects often aim at the discovery of certain substructures of a set of objects, for instance the part of the structure that causes a specific common property of the objects. The main approach to relational learning models objects and relations in a logical framework, for instance in the field of Inductive Logic Programming (ILP). In this paper, elementary objects that constitute a structured object and the (binary) relations between them are described by the nodes and edges of graphs representing the structured objects, respectively. Colors (labels) of nodes denote one-place relations, i.e. properties of elementary objects, and colors of edges represent the names of the binary relations between objects. A graph is a more general description than another graph if it has less nodes or relations, i.e. it is a part of the latter. A data mining or classification learning task is to find a general description of objects showing or not showing a common property or behavior. Using a graph representation of structured objects, this generalisation is a set of common subgraphs that are believed to cause or to prevent the property or behavior in question. An example is a certain substructure in a chemical structural formula the presence or absence of which causes some biological activity.

In relational learning, bottom-up and top-down approaches can be distinguished. Bottom-up approaches try to find general descriptions of a set of objects starting with the most specific description, i.e. the description of a given example. This description is generalised passing through several stages until a description is found that covers a subset of examples belonging to a class or subclass, respectively. The problem in this approach consists in choosing the "right" generalisation in every step of the algorithm, i.e. a generalisation that covers the parts of the objects' description which are relevant for the property to explain. Usually, algorithms of inductive logic choose the *least general generalisation (lgg)* trying to preserve as much information as possible. Unfortunately, this may result - for instance in the case of the Plotkin-lgg (see [26] for an overview) - in a growing complexity of the description in every step, because no unique (injective) mapping from one structure into the other is used.

Using a graph representation, the search for a general description of a set of objects corresponds to a search for a subgraph or subgraphs which are relevant for the property to explain[2]. Common subgraphs provide an injective mapping between the nodes and relations of two structures to be generalised, i.e. one node of one graph is mapped exactly onto one node of the other (see Section 3, Eq.(4)). As well as in ILP, the selection of the appropriate subgraph is the key problem

[2] For a comparison of graph theoretic methods with ILP see [23], Chapter 6.

in every step of generalisation. With the aim to reduce rather than expand the description length in every step, usually a largest common subgraph between the current hypothesis and the new example is selected as a new hypothesis. Another approach is to choose a subgraph which is the best with respect to the aim of the learning, for instance the Minimum Description Length criterium used in [9] which results in very large search spaces.

In generalisation based learning the aim is to produce some general descriptions of objects of the same class. A new object is believed to belong to class K if a generalisation of objects of class K exists which covers the new object. In similarity based learning, an object is assigned class K if it is more similar to some stored instances of class K than to instances of another class. Generalised Prototype algorithms [40] can be considered as a kind of mixture between generalisation and similarity based classification. Prototypes are descriptions of generalised objects which can be interpreted as abstract, typical objects of some class. Prototypes are constructed by using some similarity-based notion of generalisation whereas the classification procedure can use similarity or generalisation alone or a combination of both. The algorithm presented in this paper on the one hand produces general descriptions of objects in the form of common subgraphs which can be interpreted as typical partial structures of objects of a class. On the other hand the size of the common subgraph of two descriptions gives *simultaneously* a similarity measure (see section 3). During the construction process, the prototypes are built by generalising similar objects of the same class. Thus, the classes are divided into partly overlapping subclasses with high intra-class similarity represented by prototypes. The class of a new object is the class of the prototype which covers the object better than all other prototypes, i.e. the class of the most similar prototype. This will be described in the following.

4.2 The Similarity-Based Learning Algorithm

Although the fast approximate neural net algorithm decribed in section 3 is used, the computation of the appropriate best mapping between two graphs is still of high complexity. Thus the learning algorithm *is combined with some similarity based method of subclass or cluster formation* during the generalisation.

A simple algorithm for the generalisation of a set of examples of the same class by constructing prototypes P would be:

1. Take a set of examples S' of class K,
   mark all examples in S' as not_processed
2. WHILE examples with mark not_processed in S' exist
   (a) S:=S', choose an example y marked not_processed from S, remove y from S,P:=y
   (b) WHILE S not empty
      i. choose the x from S most similar to P, remove x from S
      ii. IF the generalisation of P and x does not cover any example from another class than K THEN P := the generalisation of P and x ,mark x as processed FI
   (c) save the new prototype P

So every example is generalised by one of the prototypes, and some examples can be generalised by more than one prototype. Thus the algorithms finds prototypes which are generalised descriptions of disjunct as well as overlapping subclasses.

The result of this generalisation process depends on the order of the examples chosen in the steps (2a) and (2(b)i). Assume that every prototype describes a subclass and that a subclass contains preferably a set of similar examples. Then a promising heuristic would be to choose the example most similar to the current prototype for the next generalisation step. This results in a heuristic where the most specific generalisation is chosen, similar for instance to the minimum inductive-leap heuristic in [23] in ILP. So the generalisation is combined with the method of agglomerative single linkage clustering of the instances around the seed y chosen in step (2a) by subsequently adding the most similar instances to a cluster described by the current prototype and containing all structures generalised by the prototype including the seed and the instances added in step (2(b)ii). This is similar to the method of conceptual clustering (see for example [13] and [18]). A similar procedure can be used to find clusters of structured objects with some common structural pattern in a database by unsupervised learning, i.e. where no classification is given a priori.

This similarity-based method requires the computation of the similarity of all remaining examples to the current prototype which is reduced in the actual computation by dividing S into two subsets which are processed in sequence. The first subset contains all examples not yet covered by a prototype. The second subset of S consists of all other examples, so in the second step the overlapping of subclasses is detected.

The overgeneralisation test in step (2(b)ii) is the most time-consuming part of the algorithm. In order to detect overgeneralisation as fast as possible, a heuristic is introduced in this step, too. Since the overgeneralisation test provides a measure of similarity between the current prototype and the tested counter-example, the counter-examples are sorted in every step by their similarity to the current prototype.

Another property of the generalisation by subgraph detection can be utilized to reduce the number of matches in the overgeneralisation test. The prototype is reduced in size in every generalisation step. So, if we are looking for a counter-example that is covered by the current prototype, it is useful to test the counter-examples most similar to the old, more specific prototype first and to omit all counter-examples where the old match contains less nodes and edges of the old prototype than the new prototype consists of.

Using these heuristics and pruning conditions, the number of graph matches in the learning algorithm can be reduced to a manageable extent.

As a result, the set of examples can be reduced to K sets of class prototypes which can be used for the distance-based classification of new examples.

5 The Variable Kernel Nearest Neighbor Classification

The success of a k-nearest neighbor classifier depends heavily on the choice of an appropriate distance measure. In order to check the usefulness of the chosen

similarity concept as well as the goodness of the approximation of this similarity function by the neural net, a weighted nearest neighbor classfication, the variable-kernel similarity metric (see [38]) and David G. Lowe's extension described in [29] has been implemented. This algorithm takes into account the distribution of the distances $d(x,y) = 1 - sim(x,y)$ in the set of given instances. The influence of any of the k neighbors on the classification is weighted by a number proportional to a Gaussian function of its distance to the instance to be classified. The parameters of the Gaussian depend on the average distance of the M (M<k) nearest neighbors. The evidence of belonging to class c for an instance x with the set of k nearest neighbors $\{x_1, x_2, ..., x_k\}$ is given by the formulae:

$$e_c(x) = \frac{\sum_{j=1, x_j \in c}^{k} w_j}{\sum_{j=1}^{k} w_j}, w_j = exp\left(-\frac{1}{2}\frac{d(x,x_j)^2}{\sigma^2}\right) \quad (8)$$

If the evidence for a class c exceeds a given threshold, for instance 0.5, x is assumed to belong to class c.

As is proposed in [29], the memory requirements and the classification effort was reduced by removing instances which are not important for classification. This is done by deleting all instances whose k nearest neighbors all belong to the same class. In our tests, the algorithm reduces the number of instances in the data set to about 60% of the original number of examples which cuts down the number of necessary matches for classification considerably.

The results presented in Section 6 show that the chosen distance measure as well as the approximation algorithm perform very well.

6 Results and Conclusions

The algorithms above have been applied to two datasets of chemical compounds provided by Dr. R. King. The sets contain nitro aromatic compounds and their mutagenicity. The aim is to produce a classifier that predicts the mutagenicity of such compounds. The first set contains 188 compounds which could be classified successfully using regression while the second set of 42 compounds caused some more difficulties. The compounds are described by graphs of atoms and bonds, providing information about the chemical elements, the kind of bonds between them, the information about the mutagenicity ("active" or "not active") and some additional data which were not used. The algorithms have been tested using tenfold crossvalidation for the first data set and leave-one-out for the smaller second one. The other data in Table (1) is taken from [16] and [24] where the dataset is described in more detail.

The SIG-Learning reduced the datasets to sets of ca. 26 prototypes and achieved the classification rates shown in the Table (1). It is outperformed by the variabel kernel k-nearest neighbor classifier introduced in this paper and some of the former classifiers but shows the advantage of producing very few prototypes which in addition contain new knowledge about substructures causing the mutagenetic activity of the substances.

In the SIG-NN algorithm as well as in the variable kernel k-NN classifier less information than in the former algorithms has been used. The authors assumed

	188	42		188	42
Linear Regression	0.85	0.67	SIG-NN	0.80	0.81
Neural Net (Backprop)	0.86	0.64	Variable Kernel k-NN, k=3	0.89	0.83
CART	0.83	0.83	Variable Kernel k-NN		
Progol	0.81	0.86	reduced set of instances		
Progol-S2	0.88	0.83	k=6	0.85	0.86
INDIGO	0.86	0.89	k=9	0.88	0.86

Table 1. Results for the mutagenesis data

that the structure of the compounds alone causes all other chemical and physical properties of the compounds except the variances caused by stereo-chemical effects. Thus, the classifier should be able to predict the mutagenicity of the compounds *using only the structural information.* So in contrast to other algorithms the classifiers described in this paper did not use additional data like atom charges or chemical or physical properties of the compound as a whole. The assumption turned out to be true. In PROGOL-S2 [24] and INDIGO [16] structural information was added to the features of the atoms by using external information (PROGOL-S2) or context information generated by the algorithm (INDIGO). In both cases this results in classification errors near the ones obtained by the algorithms described in this paper, so the differences are not significant for the given dataset where about $\frac{2}{3}$ of the examples belong to one class and the rest to the other.

The results in Table (1) show, that the chosen similarity measure as well as the connectionist algorithm for the approximation of the graph similarity are feasible for processing data represented by labeled graphs. The intrinsic complexity of graph algorithms could be reduced far enough to be able to process a data base like the mutagenesis data set. The authors are optimistic about its use for larger databases because the algorithm bears a lot of possibilities to further reduce the complexity, for instance by using parallel hardware, reducing the number of processed instances or some preprocessing of the data, including the use of domain similarity knowledge and transforming the instances into smaller graphs by producing more abstract descriptions of them. All these improvements will be implemented in the future.

References

1. D.W. Aha. *Inductive Logic Programming*, chapter Relating Relational Learning Algorithms. Academic Press, London, 1992.
2. Y. Akiyama, A. Yamashita, M. Kajiura, and H. Aiso. Combinatorial optimization with gaussian machines. In *IEEE Int. Conf. on Neural Networks*, volume I, pages 533–540. 1989.
3. J.E. Ash, W.A. Warr, and P. Willett. *Chemical Structure Systems. Computational Techniques for Representation, Searching and Processing of Structural Information.* Ellis Horwood, 1991.

4. J.M. Barnard. Substructure Searching Methods: Old and new. *J. Chem. Inf. Comp. Sci.*, 33:532–538, 1993.
5. D. Bawden. *Chemical Information Systems. Beyond the Structure Diagram.* Ellis Horwood, 1990.
6. K. Börner, K.P. Jantke, S. Schönherr, and E. Tammer. Lernszenarien im fallbasierten Schließen. FABEL Report 14, GMD, Sankt Augustin, 1993.
7. L.I. Burke and J.P. Ignizio. Neural networks and operations research: An overview. *Computers Ops.Res.*, 19(3/4):179–189, 1992.
8. L. Cinque, D. Yasuda, L.G. Shapiro, S. Tanimoto, and B. Allen. An improved algorithm for relational distance graph matching. *Pattern Recognition*, 29(2):349–359, feb 1996.
9. D.J. Cook and L.B. Holder. Substructure Dcovery Using Minimum Description Length and Background Knowledge. *Journal of Artificial Intelligence Research*, 1:231–255, 1994.
10. W. Emde and D. Wettschereck. Relational instance-based learning. In L. Saitta, editor, *Proc. of the 13th International Conference on Machine Learning*, pages 122–130. Morgan Kaufmann, 1996.
11. J. A. Feldman and D. H. Ballard. Computing with Connections. TR 72, Computer Science Department, University of Rochester, April 1981.
12. J. A. Feldman, M. A. Fanty, N. Goddard, and K. Lynne. Computing with Structured Connectionist Networks. TR 213, Department of Computer Science, University of Rochester, April 1987.
13. D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2):139–172, 1987.
14. N. Funabiki, Y. Takefuji, and K.-C. Lee. A neural network model for finding a near-maximal clique. *Journal of Parallel and Distributed Computing*, 14(3):340–344, March 1992.
15. P. Geibel, K. Schädler, and F. Wysotzki. Begriffslernen für strukturierte Objekte (Concept Learning for Relational Structures). In *Proceedings of FGML-95, Dortmund, Germany*, 1995.
16. P. Geibel and F. Wysotzki. Learning relational concepts with decision trees. In L. Saitta, editor, *Machine Learning: Proceedings of the 13th International Conference*, pages 166–174. Morgan Kaufmann Publishers, San Fransisco, CA, 1996.
17. P. Geibel and F. Wysotzki. Relational learning with decision trees. In W. Wahlster, editor, *Proceedings of the 12th European Conference on Artificial Intelligence*. John Wiley and Sons, Ltd., 1996. to appear.
18. J. H. Gennari, P. Langley, and D. Fisher. Models of Incremental Concept Formation. *Artificial Intelligence*, 40:11 – 61, 1989.
19. J. Hopfield and D. Tank. Neural computations of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1986.
20. J.J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. In *Proceedings of the National Academy of Sciences USA 81*, pages 3088–3092. 1984.
21. A. Jagota. Efficiently approximating MAX-CLIQUE in a hopfield-style network. In *Proceedings of International Joint Conference on Neural Networks '92 Volume II*, pages 248–253, 1992.
22. F. Kaden. Graphmetriken und Distanzgraphen. In *Beiträge zur angewandten Graphentheorie*, ZKI-Information. Berlin, Juni 1982.
23. J. Kietz. *Induktive Analyse Relationaler Daten.* PhD thesis, TU Berlin, FB 13, 1996.

24. R. D. King, M. J. E. Sternberg, A. Srinivasan, and S. H. Muggleton. Knowledge Discovery in a Database of Mutagenic Chemicals. In *Proceedings of the Workshop "Statistics, Machine Learning and Discovery in Databases" at the ECML-95*, 1995.
25. W. Knödel. Ein Verfahren zur Feststellung der Isomorphie von endlichen, zusammenhängenden Graphen. *Computing*, 8, 1971.
26. N. Lavrac and S. Dzeroski. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, New York, 1994.
27. K.B. Lipkowitz and D.B. Boyd. *Reviews in Computational Chemistry*, volume I-IV. VCH, Weinheim, 1990-1993.
28. C. Looi. Neural network methods in combinatorial optimization. *Computers and Operations Research*, 19(3/4):191–208, 1992.
29. D. G. Lowe. Similarity metric learning for a variable-kernel classifier. Technical Report UBC-TR-93-43, Computer Science Dept., University of British Columbia, Vancouver, B.C., V6T 1Z4, Canada, November 1993.
30. H. Mannila. Aspects of data mining. In *Proceedings of the Workshop "Statistics, Machine Learning and Discovery in Databases" at the ECML-95*, 1995.
31. M. Moulet and Y. Kondratoff. From machine learning towards knowledge discovery in databases. In *Proceedings of the Workshop "Statistics, Machine Learning and Discovery in Databases" at the ECML-95*, 1995.
32. K. Schädler, U. Schmid, B. Machenschalk, and H. Lübben. A neural net for determining structural similarity of recursive programs. In *Proc. of the German Workshop of Case-Based Reasoning*, Bad Honnef, March 1997.
33. K. Schädler and F. Wysotzki. Klassifizierungslernen mit Hilfe spezieller Hopfield-Netze. In Werner Dilger, Michael Schlosser, Jens Zeidler, and Andreas Ittner, editors, *Beiträge zum 9.Fachgruppentreffen "Maschinelles Lernen" der GI-Fachgruppe 1.1.3.*, number CSR-96-06 in Chemnitzer Informatik-Berichte, pages 96–100. TU Chemnitz-Zwickau, August 1996.
34. K. Schädler and F. Wysotzki. Theoretical foundations of a special neural net approach for graphmatching. Technical Report 96-26, TU Berlin, CS Dept., 1996.
35. L.G. Shapiro and R.M. Haralick. A metric for computing relational description. *IEEE Trans.Pattern Anal. Mach.Intell.*, 7(1):90–94, 1985.
36. E. Tammer, K. Steinhöfel, S. Schönherr, and D. Matuschek. Anwendung des Konzepts der Strukturellen ähnlichkeit zum Fallvergleich mittels Term- und Graph-Repräsentationen. Technical Report FABEL-Report No.38, HTWK Leipzig, FB Informatik, Mathematik und Naturwissenschaften, August 1995.
37. A. Voß. Similarity concepts and retrieval methods. FABEL Report 13, GMD, Sankt Augustin, 1994.
38. M. P. Wand and M. C. Jones. *Kernel Smoothing*. Chapman and Hall, London, 1995.
39. J. Wang. *Progress in Neural Networks*, volume 3, chapter 11: Deterministic Neural Networks for Combinatorial Optimization, pages 319–340. Ablex Publishing Corporation, Norwood, New Jersey, 1995.
40. Ch. Wisozki and F. Wysotzki. Prototype, nearest neighbor and hybrid algorithms for time series classification. In N. Lavrac and S.Wrobel, editors, *Machine Learning: ECML-95*, number 912 in LNAI, pages 364–367. Springer, 1995.
41. F. Wysotzki. Artificial Intelligence and Artificial Neural Nets. In *Proc. 1st Workshop on AI*, Shanghai, September 1990. TU Berlin / Jiao Tong University Shanghai.
42. B. Zelinka. On a certain distance between isomorphism classes of graphs. *Časopis pro pěstováni matematiky*, 100:371–373, 1975.