CONCEPT LEARNING BY STRUCTURED EXAMPLES - AN ALGEBRAIC APPROACH

Fritz Wysotzki, Werner Kolbe, Ooachim Selbig

Dept. of Artificial Intelligence
Central Institute cf Cybernetics and
Information Frocesses of the Academy of Sciences
1080 Berlin, German Democratic Republic

ABSTRACT: A system learning concepts from training samples consisting of structured objects is described. It is based on descriptions invariant under isomorphism. In order to get a unified mathematical formalism recent graph theoretic results are used- The structures are transformed into feature vectors and after that a concept learning algorithm developing decision trees is applied which is an extension of algorithms found in psychological experiments. It corresponds to a general-to-specific depth-first search with reexamination of past events.
The generalization ability is demonstrated by means of the blocks world example and it is shown that the algorithm can successfully handle practical problems with samples of about one hundred of relatively complicated structures in a reasonable time. Additionally, the problem of representation and learning context dependent concepts is discussed in the paper.

1. Introduction

Concept learning on sets of structural descriptions has become one of the most challenging problems in AI-research in the last decade /1/, /3/, /7/, /9/, /10/, /13/,/14/.
There is increasing success of programs that make use of domain-specific knowledge like DENDRAL and METADENDRAL. But there remain areas, where a priori knowledge may be not or only partially available or the aquisition and application of knowledge would be difficult as in constructing rules from large sets of empirical data.
On the other hand in order to leave the empirical stage and to establish theoretical foundations AI has to discover general principles and to find appropriate formalizations for them. Therefore unified mathematical models should be applied as far as possible-
There are practical applications of concept or discrimination learning in which first the structures and training samples are large, second the concepts may be disjunctive sets of subconcepts and third matching is complicated because of the existence of many isomorphic descriptions (alphabetic variants).
It is felt that in such cases well elaborated general algorithms such as the inductive generalization in the predicate calculus and even the version space method /8/ would face problems of combinatorial explosion, (Refinements of the predicate calculus-based methods are developed in our laboratory /5/ which, it is hoped, will overcome some of the difficulties in dealing with practical problems.)

In this paper first a new method of the description of structures is represented which is based on recent graph theoretic results in the detection of isomorphism /4/.
From descriptions being almost invariant under isomorphism feature vectors are formed describing the structures unambiguously. This enables us to use well known techniques of concept learning on feature vectors to solve or at least to reduce the complexity of the concept learning task. In particular we adopt a method of sequentially constructing decision trees representing the hypotheses in terms of discriminant descriptions of a possible multiclass problem.
The algorithm corresponds to a general-to-specific depth-first search with reexamination of past instances.
As a first practical example a set of 89 chemical compounds with up to 15 nodes each is treated successfully.
In addition, our approach gives the possibility of learning concepts defined by relations to other concepts (context), a problem which until now has not yet been a subject of AI-research.

2. Description of structures

Definition 2.1.
A structured object (briefly structure) is a relational algebra

$$\Sigma = (V_\Sigma ; P_1, \dots, P_{s_1}, R_1, \dots, R_{s_2})$$

with the set of elementary objects V_Σ, the set of one-place relations P_i, $i = 1,\dots,s_1$ and the set of two-place relations R_j, $j = 1,\dots,s_2$ (Higher order relations may be transformed into sets of two-place ones).

It is supposed that Σ is represented in the memory by a labelled graph G_Σ with the set of nodes N corresponding to the set of elementary objects V_Σ by the bijection

$$\bar{d}: V_\Sigma \longleftrightarrow N = \{1, 2, \dots, n\}.$$

Different maps $\bar{d}$ lead to isomorphic descriptions (alphabetic variants). A node is labelled with all relations of which the elementary object corresponding to the node is an element. An arc is labelled by the corresponding two-place relations (fig. 1).

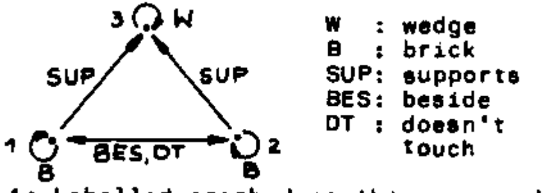


Fig. 1: Labelled graph describing an arch

Suppose a training sample $S = \{G_1, G_2, \dots, G_M\}$ of descriptions of structures is given. Each combination (vector) of relations found in the training sample a colour or new complex relation is ascribed (fig. 2).

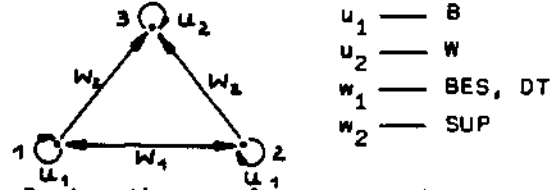


Fig. 2: Ascribing colours to nodes and arcs of fig. 1

Let U be the set of node colours and W the set of arc colours found in the training sample. We refer to U and W as the set of elementary properties (feature vectors) of nodes and arcs respectively. Formally, we have the following definitions:

Definition 2.2.
A description of a structure Σ is a coloured graph $G_\Sigma = (N, Q, U_\Sigma, W_\Sigma, f, g)$ with the set of nodes N, the set of arcs $Q \subseteq N \times N$ and the colouring functions

$$f : N \rightarrow U_\Sigma \subseteq U, \quad g: Q \rightarrow W_\Sigma \subseteq W.$$

Definition 2.3.
Two descriptions G and G' are isomorphic if there exist a bijective map $d : N \longleftrightarrow N'$ with

$$(d(i), d(j)) \in Q' \leftrightarrow (i, j) \in Q$$
$$f(i) = f(d(i)) \text{ and } g(i, j) = g(d(i), d(j)).$$

An automorphism of G is an isomorphism onto itself.

Isomorphic graphs in the training sample are supposed to describe the same real structure EW and to arise from using different coding strategies. (Therefore they must belong to the same class in the case of consistent training samples.) Identifying isomorphic descriptions is especially in practical applications a very important task. As enovel approach to this problem in our system descriptions invariant under isomorphism are used in order to get efficient concept descriptions, matching and generalization possibilities. An invariant characterization of a structure would be the set of all n! descriptions generated from a description given in the training sample (or as test item to be matched) by permutations of the nodes. But this set is, in general, much too large.
(A normalization to a standard description by alphabetic ordering would hide similarities of structures which are used in the construction of efficient concept descriptions, see below.)
To reduce the number of descriptions generated a method used in algorithms detecting isomorphism of graphs /4/ was adopted*"*")

Definition 2.4.
Two nodes i, j of a description G are (elementary) distinguishable iff $f(i) \neq f(j)$, i. e. they have different colours.

By this definition a partition of the set of nodes into equivalence classes of equally coloured nodes

$$S_u = \{j \,/\, j \in N, f(j) = u\}, \quad u \in U_\Sigma,$$

is induced, this partition being invariant under isomorphism. The number of descriptions is therefore reduced from n! to $\prod_{u \in U_\Sigma} \text{card } S_u!$. For example in fig. 2 there are the equivalence classes S_{u_1} and S_{u_2} with card $S_{u_1} = 2$ and card $S_{u_2} = 1$, i. e. $3! > 2! \cdot 1!$.

To reduce the number of descriptions further context dependent features of nodes and arcs are introduced.

Definition 2.5.
Let $i \in N$, $(i, j) \in Q$, then

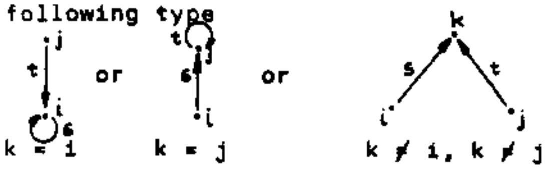
$$n_w(i) = \text{card } \{j \,/\, g(i, j) = w\}, \quad w \in W,$$

is the number of neighbours of node i which can be reached by a w - coloured arc. For arcs similarly

$$n_{st}(i,j) = \text{card } \{k / f(i,k) = s \text{ and } f(j,k) = t\},$$
$$(s,t) \in U \times W \text{ or } (s,t) \in W \times U \text{ or } (s,t) \in W \times W \text{ is defined.}$$

+) The authors would like to thank Dr. D. Pötschke for many helpful theoretical discussions.

The n_{st} are the numbers of graphs of the following type



For example in fig. 2 one has

$$n_{u_1w_1}(1,2) = 1,\ n_{w_1u_1}(1,2) = 1,$$
$$n_{w_2w_2}(1,2) = 1$$

resulting by reconstruction in the partially labelled graph of fig. 3.

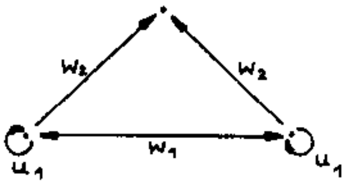


Fig. 3: Context dependent arc features

The $n_w(i)$, $w \in W$ can be ordered to get together with the original colour f(i) a new (context dependent) feature vector $(f(i), n_1, n_2, ..., n_r)$ for every node i. Similarly one gets new feature vectors for arcs (i, j).

These feature vectors can be used <u>in learning concepts which are defined by context</u> (as most human concepts are!), i. e. when the membership of an object to a class (concept) is defined by relations to other objects. In this case a training sample consists of a set of <u>distinguished elementary objects (nodes)</u> together with structures in which they are embedded and the class of the nodes (being positive or negative instances) depends on the existence of certain subgraphs in the embedding structures. The theory is outlined in /12/.

The step described in Def. 1.5. is done for all descriptions in the training sample and leads to new sets $U^{(1)}$ and $W^{(1)}$ of node and arc colours respectively with new colouring functions $f^{(1)}$ and $g^{(1)}$ for each graph. (In a sequential learning task the sets $U^{(1)}$, $W^{(1)}$ may be sequentially updated.) Nodes i,j which are not elementary distinguishable (definition 2.4) may now become distinguishable by context i. e. $f^{(1)}(i) \neq f^{(1)}(j)$.

Example: The nodes 1 and 2 in fig. 2 not elementary distinguishable become distinguishable by context if the asymmetric relation LEFTOF is added to arc (1,2). Additionally, the new features $\in U^{(1)}$, $W^{(1)}$ are easily computable graph characteristics, which can be used for the concept learning task (see below).

<u>Proposition 2.1.</u>
The new partition of N according to $U^{(1)}$ is equal or finer than the partition according to U and the number of permutations may be further reduced. Proof: see /12/.

The step of def. 2.5. may be repeated and leads in that case after a finite number of iterations to stable $f^{(i)}$'s and $g^{(i)}$'s, thereby <u>getting possibly exactly one description of</u> Σ (automorphism partition of the nodes /4/).

In practical cases and sequential learning tasks one would stop after a fixed number i of steps depending on the problem. Let the set of descriptions of Σ generated by the remaining permutations by $D_\Sigma^{(i)}$

3. <u>Transformation of graphs into feature vectors</u>

Now we want to transform the graphs $G_\Sigma^{(i)} \in D_\Sigma^{(i)}$ into linear representations (feature vectors) which can be easily handled by well elaborated algorithms of concept learning on sets of feature vectors. First we represent all edges (j,k) of $G_\Sigma^{(i)}$ together with their adjacent nodes j,k by triples (w_{jk}, u_j, u_k) of their colours. If $u_j = u_k$ then j and k belong to the same equivalence class and have to be distin guished by an additional number in order to save the full graph information (for instance the pair (1,2) in fig. 2 has to be described by $(w_1, (u_1, 1), (u_1, 2))$). All different triples occurring in the training sample are constructed (or sequentially updated) and enumerated* These triples correspond to elementary propositions with the labels u,w indicating complex semantic features of case arguments and relations respectively.

Another possibility of defining features for the graph classification task consists in choosing subgraphs of several orders (gestalt features). A method of selecting subgraphs and using them as tests to be built into decision tree-classificators has been developed in our laboratory successfully and applied to practical problems /II/-

A feature vector $v^{(i)}$ corresponding to a graph $G_\Sigma^{(i)} \in D_\Sigma^{(i)}$ is now defined as follows

$$v^{(i)} = (x_1, ..., x_r;\ y_1, ..., y_s;\ z_1, ..., z_t)$$

x_j = l iff the node colour $u_j \in U^{(i)}$ occurs l times in

$G_{\Sigma}^{(i)}$, l = 0, 1, ...

y_j = l iff the arc colour $w_j \in W^{(i)}$

occurs l times in $G_{\Sigma}^{(i)}$

$$z_j = \begin{cases} 1 \text{ iff triple } j \text{ occurs in } G_{\Sigma}^{(i)} \\ 0 \text{ otherwise} \end{cases}$$

Let $\overline{V}_{\Sigma}^{(i)}$ be the set of all vectors generated from all graphs $G_{\Sigma}^{(i)}$ in $D_{\Sigma}^{(i)}$.

Theorem: The map $\Sigma \longleftrightarrow \nabla^{(i)}$ is a bijection, i. e. the set $\overline{V}_{\Sigma}^{(i)}$ describes unambiguously the structure Σ . The proof is given in /12/.

4. Concept learning

As an algorithm for concept learning a method of sequentially constructing hypo*-theses in form of decision trees is used, which is more appropriate to problems with large feature vectors# It is a generalization of human behaviour observed in learning concepts represented by propositional functions /12/ and it is similar to the algorithm of learning discrimination nets for syllables described in /2/. It corresponds to a general-to-specific depth-first search with reexamination of past events. (Due to limited STM capacity human beings are keeping only one hypothesis per step in memory. This method is preferred to a possible breath-first search since in case of large structures, large training samples, and disjunctive concepts the number of hypotheses to be pursued simultaneously would be probably to large even in the case of the lattice-theoretic approach used in /8/. In large samples it is also not easy to get initial clusters for disjunctive concepts.) Since the decision trees are built up serially by adding no more than one new test attribute (test node) per step to the tree, the set of attributes A = {v ,...,v } constituting the feature vectors is a priori reordered by some elementary discrimination measure known in pattern recognition. (In a more sophisticated approach this procedure is applied to each branch of the tree representing the current hypothesis.)
The algorithm can be defined in terms of simple data driven production rules. In the case of finite consistent training samples it can be proved /12/ that it results in a final hypothesis which matches all training instances correctly. In /12/ extensions to the case of inconsistent training samples, probabilistic decisions, and continuously varying attributes (algorithmic construction of feature intervals during learning) can also be found.

Generalization

First, one gets a simple generalization by means of the fact that the construction algorithm, in general, stops before all attributes are exhausted on each path. Second, irrelevant attributes in the tree (or conditionally irrelevant attributes in subtrees) may be contained leading to symmetric branches which can be detected. An equivalence transformation of the tree is then performed using a calculus first described by MCCARTHY /6/ for optimizing propositiona1 functions in tree-form. Attributes being dependent on others already used on a certain path are also detected and eliminated thus getting additional rules describing feature dependencies and the structure of the domain.
Now let us consider the tree in fig. 4 which represents a hypothesis in a concept learning problem on structures. A test node is labelled by a triple z , i.e.

If z exists in the structure to be matched, the structure passes the 1-branch, if not it passes the 0-branch. Each path in the tree represents a subclass (subconcept) by a conjunction of the z 's (1-labelled branches) or their negations (0-labelled branches), the whole concept being the disjunction of all paths with terminal nodes labelled by the concept. (The extension of the formalism to multiclass discrimination problemsis obvious.) Since the z 's correspond to triples describing an aVc together with its adjacent nodes, unambiguously identified "by their colours (possibly with additional numbers), the positive z 's on a path can be combined into a subgraph characterizing the concept (fig. 4). The negated z 's on this path correspond to the "must not" conditions of /14/ i. e. they are forbidden in the substructure.

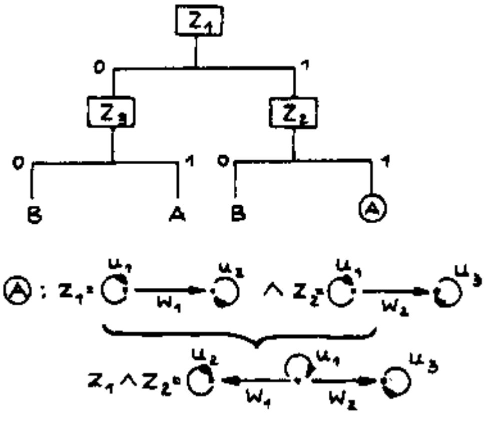


Fig. 4: Decision tree and description of a subconcept

Generalization on node and edge features

Generalization on node and edge features may depend on the place in the structure where the elementary object or relation

is embedded, therefore a priori given generalization hierarchies on elementary features may not be helpful. For instance, it may be irrelevant for the concept "arch" (fig. 1), whether the top object (and only this) is a brick or wedge. Therefore, generally, the system has to learn context dependent generalization hierarchies.
Irrelevant features of such kind lead to substructures in the trees constructed by our algorithm having an "analogous" composition in the sense that the substructures can be made identical and merged by introducing variables for corresponding constituents in corresponding triples. This procedure is the tree-analogue of VERE's algorithm of inductive generalization in the predicate calculus /13/.

5* Experiments

To illustrate general issues and techniques of our approach one particular concept learning problem, originally discussed by WINSTON /14/ will be described. It involves the learning of how to identify simple classes of structures built of children's blocks. The task is to learn the concept of an arch, with a series of block structures being given, each labelled as either an arch or a non-arch. In comparision with WINSTON^s world our sample is supplemented by additional structures. It consists of 10 or 11 elements (see fig. 5).

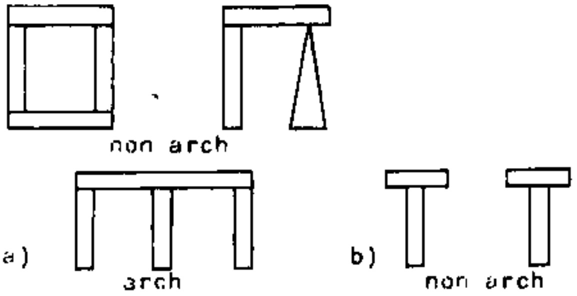


Fig. 5: Some of the additional training examples

The described algorithm constructs complex features to characterize all elements of the training sample (see 2.). The decision tree approach selects only those tests from the set of the a priori reordered attributes that it needs to identify all training examples-
The results are illustrated in fig. 6. The decision tree resulting from the sample that does not contain the structure shown in fig. 5 b) is illustrated in fig. 6 a). It contains only one node feature. The label '2' in the test of the decision tree means/ that there must be at least two nodes 'of that type' in a structure. The algorithm produces the decision tree shown in fig* 6 b), if the training sample contains additionally the structure shown in fig. 5 b). The selected arc feature (fig. 3) discriminates all arches from the non-arches. (Note the generalization on the form of the top object.)

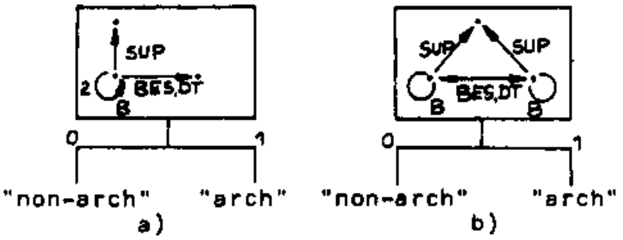


Fig. 6: Decision trees of a blocks world example (for reading the tests in the decision trees se fig. 1).

Our program is written entirely in FORTRAN and consists of about 3 000 executable statements. It runs in 150 K bytes of memory on the BESM-6 computer.

6. Application

In pharmacology there is the problem to find atoms or groups of atoms (substructures) in chemical compounds necessary to evoke a specific biological activity. The knowledge of such substructures may be a starting point for the synthesis of new drugs.
As an example a training sample of 89 carboxamides the structural formulas of which were known was investigated. These compounds scaled by their biological activity were divided into four classes. The structural formulas of the compounds were transformed into coloured graphs- Each atom or group of atoms is considered as a node in the graph, each type of bond as an edge*
Our algorithm described above was applied to this four-class problem* From 89 graphs (objects) the computer generated by the permutation of some of them 103 descriptions with altogether about 300 complex features(triples) including the context after the second iteration (see 2*).
A disjunction of characteristic substructures was obtained for each class. Our results were confirmed by the empirical knowledge of chemists.
The central processor time needed on the BESM-6 was about 20 minutes.

7. Summary and conclusions

A concept learning system has been described which is based on descriptions of structures invariant under isomorphism and transformation into feature vectors. The generalization abilities are demonstrated in case of the blocks world example.
The algorithm can treat training samples of about one hundred of relatively com-

plicated structures in a reasonable time.

Since the use of all possible features (graph properties) characterizing a structure would lead to combinatorial explosion, subsets of feature are extracted by a unified principle. Global characteristics (subgraphs) relevant to the concept *are* subsequently synthesized in the form of Boolean functions constituting the final concept description. Complex or elementary features irrelevant to the concept and yet included in the final hypothesis can be eliminated by a secondary generalization and optimization procedure.

Acknowledgements

The authors would like to thank Dr. D. Potschke and Dr. G. Stanke for many helpful discussions. S- Dbring and R. Miiller *are* gratefully acknowledged for programming parts of the system.

Reference

/ l / DIETTERICH, T. G.; MICHALSKY, R. S.: Learning and generalization of characteristic descriptions: evaluation criteria and comparative review of selected methods; In Proc. IJCAI-6, 1979.
/2/ FEIGENBAUM, E.: The simulation of verbal learning behaviour; In Computers and Thought (E. FEI GENBAUM and 0. FELDMAN, Eds.), New York 1963.
/3/ HAYES-ROTH, F.: Collected papers on the learning and recognition of structured patterns; Carnegie-Mellon Univ., 1975.
/4/ HINTEREGGER, 0.: Ein verbessertes Verfahren zur Feststellung der Isomorphic endlicher Graphen; Dissertation, Innsbruck 1976.
/5/ KADEN, F.: Zur Formalisierung induktiver Schlusse uber strukturierten Objekten; 2KI-Informationen 3/80, Akademie der Wissenschaften der DDR, 1980
/6/ MCCARTHY, J.: A basis for a mathematical theory of computation; In Computer Programming and Formal Systems (P. BRAFFORD, D. HIRSHBERG, Eds.), Amsterdam 1963.
/!/ MICHALSKY, R. S.: Toward computer-aided induction: a brief review of currently implemented AQVAL programs; In Proc. IJCAI-5, 1977.
/8/ MITCHELL, T. M.: Version spaces: an approach to concept learning; Ph. U. thesis, STAN-CS-711, Stanford 1978.
/9/ MITCHELL, T. M.: Analysis of generalization as a search problem; In Proc. I3CAI-6, 1979.
/10/ PLOTKIN, G. D.: A note on inductive eneralization; In Machine Intelligence 5 B. MELTZER and D. MICHIE, Eds.), Edinburgh 1970.
/ l l / SOBIK, F,; SOMMERFELD, E.: The program CALG for classification of structured objects; Proc. 2 Int. Meeting on Artificial Intelligence, Leningrad-Repino, 1980 (to be published).
/12/ UNGER, S.; WYSOTZKI, F.: Lernfahige Klassifizierungssysteme; Akademie-Verlag Berlin 1981.
/13/ VERE, S. A.: Induction of concepts in the predicate calculus; In Proc. IJCAI-4, 1975.
/14/ WINSTON, P. H.: Learning structural descriptions from examples; Ph. D. thesis, MAC TR-76, Cambridge 1970.