South African
Computer
Journal
Number 6
March 1992

Suid-Afrikaanse
Rekenaar-
tydskrif
Nommer 6
Maart 1992

Computer Science and Information Systems

Rekenaarwetenskap en Inligtingstelsels

The South African Computer Journal

An official publication of the South African Computer Society and the South African Institute of Computer Scientists

Die Suid-Afrikaanse Rekenaartydskrif

'n Amptelike publikasie van die Suid-Afrikaanse Rekenaarvereniging en die Suid-Afrikaanse Instituut vir Rekenaarwetenskaplikes

Subscriptions

	Annual	Single copy
Southern Africa:	R45_00	R15_00
Elsewhere	$45_00	$15_00

to be sent to:
Computer Society of South Africa
Box 1714 Halfway House 1685

Computer Science Department, University of Pretoria, Pretoria 0002
Phone: (int)+(012)+420-2504 Fax: (int)+(012)+43-6454 email: dkourie@rkw-risc.cs.up.ac.za

Sigma Press

Guest Contribution

This guest contribution is a slightly edited report to the Foundation for Research and Development (FRD) drawn up by Ed Coffman. Ed was an FRD-sponsored guest at the 6th South African Computer Research Symposium. The report was not originally intended for general distribution. Rather, it was specifically compiled for the FRD and its staff. I am therefore grateful to both the FRD and the author for agreeing to its publication in SACJ. I believe that it contains several incisive observations that merit further thought and discussion amongst South African computer scientists. (Editor)

Impressions of Computer Science Research in South Africa

E. G. Coffman, Jr.
AT&T Bell Laboratories, USA

In commenting on the cross section of computer science research in South Africa, I will use the classification in the table of contents of the "Summary of Awards: Fiscal Year 1989," a document published recently by the US National Science Foundation. Of the 5 categories, I will treat Numeric and Symbolic Computation as inappropriate for the discussion below. In this category I noted no research in the computer science setting in South Africa. It is also common in the US and elsewhere to place this effort in other departments, e.g., departments of mathematics or applied mathematics.

Of the remaining categories I found South Africa to be strongest in software systems and engineering, to have a substantial investment in computer systems and architecture, and to be weakest in computer and computation theory.

The coverage in software systems and engineering (SSE) was broad, topical, and similar in scope to that in US universities. Technology transfer and the corresponding relations with industry seemed to be in place or developing along promising lines. I comment in passing that this was rather surprising to me. In the US the development of SSE within university departments has lagged behind almost all other disciplines of computer science. A primary problem has been the insatiable appetite of industry for all Ph.D. graduates in the SSE field.

The investment in parallel processing, computer networks, and distributed computing appears sound, although I expected to see a greater emphasis on mathematical foundations (see my remarks below), particularly in the parallel algorithms area. Given current resources, South African institutions are doing remarkably well in computer science research. But computer science is a fundamentally important course of study, beginning at an early age and extending through graduate Ph.D. research; I take this as sufficiently obvious that I need not dwell on justifications. With this in mind, and with the necessary resources in hand, South Africa should, in my opinion, expand and consolidate its computer science research effort, increase its visibility in the international arena, and correct the rather thin distribution of graduate research among universities.

I can see much of this proceeding along present lines, but I would strongly recommend a concerted development in computer and computation theory (CCT), education and research; this is mainstream computer science and forms the basis for virtually all other fields of study within computer science. It is by no means absent in South Africa curricula, but it appears to be under--represented in advanced studies and Ph.D. level research.

At the graduate level CCT is heavily mathematical. I understand that mathematical foundations are supplied by mathematics departments in certain cases. This is not ideal, but workable and it is justified by limited resources. However, it is important that mathematics departments not regard this as a mere service; faculty will have to make a major commitment to theoretical computer science, publishing in its leading journals (e.g. SIAM Journal of Computing, Journal of the ACM, Journal of Algorithms, Algorithmica, Journal of Computer and Systems Sciences, Theoretical Computer Science, etc.), and providing the supervision of theses sponsored by computer science departments and leading to degrees in computer science. I would also encourage active participation in the international computer science "theory" societies and their meetings; two highly prestigious examples of the latter are the annual Symposium on the Theory of Computing and the Foundations of Computer Science conference.

Returning to the thin distribution of computer science research, I would make the following point. If the current situation is only a stage of development - i.e., if further resources (both human and financial) can be counted on to bring at least a few of the departments to a critical mass - then little needs to be said beyond the earlier remarks. Critical mass is hard to define, but calls for adequate, expert coverage of mainstream computer science re- search. In view of the breadth of this research, 8-10 Ph.D. full- time-equivalent faculty would seem to be barely adequate; with the usual clumping of faculty in specific research areas, more would be expected. South Africa has a talent base such that there is little doubt that such departments would achieve a much wider international recognition.

On the other hand, if resources remain fixed at current or even slightly retrenched levels, then I would recommend consolidation to achieve the same goals on a smaller scale. Within a university this can often be done by establishing interdisciplinary, degree-granting laboratories or institutes of computer science, which bring together the computer science efforts located in various departments other than computer science, such as electrical engineering, industrial engineering, business/-management science, mathematics, and operations research. The idea is to enjoy the advantages (opportunity, synergy, awareness, etc.) to both students and faculty of reasonably large computer science programs. There are many examples of such intramural laboratories in North America and Europe.

This approach could also be considered among universities within a confined geographical area, admittedly with greater difficulty perhaps. The Institute of Discrete Mathematics and Computer Science connecting Princeton University, Rutgers University, AT&T Bell Laboratories, and Bell Communications Research is a possible model. Examples in South Africa might consist of universities and research institutions on the Reef or those in the Western Cape (just to mention those with which I'm a little familiar).

As a final comment, I should note that my impressions have been based on limited information which may not give a representative picture. I am sure that my reactions will be appropriately discounted where I have been off target.

Editor's Notes

Prof John Schochot has graciously accepted to be SACJ's subeditor for papers relating to Information Systems. Authors wishing to submit papers in this general area should please contact him directly. I look forward working with John, and to a significant increase in IS contributions in future.

The hand of the new production editor, Riël Smit, will be clearly evident in this issue. Those papers not prepared in camera-ready format by the authors themselves were prepared by him in TEX. He will be announcing revised guidelines for camera-ready format in a future issue. If you use TEX or one of its variations, Riël would be happy to provide you with a styles document to SACJ format.

At last some Department of National Education committee has decided that SACJ should now be on the list of approved journals. This places it amongst the ranks of some 6800 other journals. These include not merely a number of ACM and IEEE Transactions but also such journals as *Ostrich*, *Trivium*, *Crane Bag*, *Koers*, *Mosquito News*, *Police Chief*, *Connoisseur*, *Lion and the Unicorn*, *About the House* and *Ohio Agricultural Research and Development Center Department Series ESS*. You will recall that in 1990 this same committee decided that, if judged on its own merits, SACJ did not deserve to be on the illustrious list. In the absence of other evidence, we must assume that the sole reason for its revised decision is that SACJ's predecessor, Quæstiones Informaticæ, was there. (I have a secret suspicion that the committee liked that name.)

It is my understanding that for official purposes, all journals on this list are regarded as *equally* meritorious, and all of them are more meritorious than *any* conference proceedings. What does all of this mean?

The momentous implication of the committee's deliberations is that the State will not give your institution a single cent for anything that you publish in SACJ. Instead, the State and your institution will scrupulously keep a score of the annual number of publications that count - but actually don't - because someday they might! And to encourage your enthusiastic participation in this Alice in Wonderland exercise, your institution might actually give you some of the standard subsidy funding that the State should have provided according to its own formulae, but didn't.

You will not be allowed to use this money to buy yourself a car - not even a casual meal. You may only use it to finance activities that are provably directed towards producing more papers in approved journals. The great consolation, of course, is that you will not be required to pay income tax on this money. The only tax involved will be the VAT component when you spend it in an approved manner. As a good computer scientist who enjoys recursion, my vote would be that all such revenue collected by the State should be earmarked to be placed in the pay packets of committee members who decided that SACJ should be approved.

If you publish in these approved journals with sufficient regularity and enthusiasm you will almost deserve to be regarded as a researcher. What you additionally need to do, is to ensure that you befriend and impress at least three overseas referees. You then apply to the FRD for official recognition as a researcher, and if they are sufficiently impressed, they will give you more of the non-taxable kind of money that you need to spend on research to publish in approved journals.

Derrick Kourie
Editor

Integrating Similarity-Based and Explanation-Based Learning

G. Deon Oosthuizen and Chrisna Avenant
Department of Computer Science, University of Pretoria, 0083 Hillcrest, South Africa
email: goosthui@rkw-risc.cs.up.ac.za

Abstract

Recently, there have been various attempts to combine the strengths of similarity-based learning (SBL) and explanation-based learning (EBL) in a single learning system. We describe a graph-based learning method called Graph Induction, which is based on the graphical representation of a formal lattice and supports both supervised and unsupervised learning. The method integrates SBL with a weak form of EBL in such a way that the two mechanisms become totally blended. The result is a unified algorithm with both SBL and EBL involved in each step. The domain theory is generated and/or extended as SBL proceeds and employed immediately, through EBL, to guard further learning and thus control the size of the lattice which otherwise has the potential for increasing exponentially.
Keywords: *Artificial Intelligence, Machine Learning*

Received January 1991, Accepted January 1991

1 Introduction

Two major Machine Learning paradigms are Empirical and Analytical learning. Empirical learning induces rules from a number of examples, referred to as the training set. Since many of the methods used are based on the similarities between samples, they have become known as Similarity-Based Learning (SBL) methods. Analytical learning employs domain theory to induce a class description from one sample only. Since this was viewed by many authors as a process of learning by explaining why a sample belongs to a given class, it has become known as Explanation-Based Learning (EBL).

Recently various attempts have been made to combine the strengths of SBL and EBL [20]. We describe a representation model which allows the two paradigms to be integrated, complementing each other during each learning step. Langley points to the fact that the two paradigms are more similar than the literature suggests [6]. The current paper harbours the same sentiment. The representation model described forms the basis of an incremental inductive learning method, called Graph Induction, implemented in a system called GRAND (GRAph inDuction) [13]. Conceptual clustering forms the basis of generalization in GRAND. Thus, it is primarily a concept formation method [16], extending the ideas embodied in the UNIMEM [8] and COBWEB [2] systems. However, GRAND also supports concept learning as a special case of conceptual clustering by regarding the class of a sample – i.e. the concept to be learnt - as just another feature. This places GRAND among systems like AQ15 [10] and ID5 [21]. GRAND is also closely related to the CHARADE system [3].

In this paper, we focus on the interplay between similarity-based and explanation-based learning taking place in GRAND. In this regard, GRAND has much in common with work done by others [7], [18] and [22].

We first describe the representation model and then explain how SBL and EBL are supported and integrated. We conclude by comparing GRAND with related systems.

2 Graph Induction

Graph induction derives its name from the fact that it is based on the explicit graphical representation of a formal lattice. We now explain how the lattice is constructed and how it supports inductive learning.

2.1 Construction of the lattice

A lattice is an acyclic directed graph in which every pair of nodes has a least common superior (the join) and a greatest common subordinate (the meet) which are necessarily unique. A lattice is constructed by first creating a node for each value of each attribute to be recorded in the system. These nodes can be considered as the upper layer of a network to be expanded below them. Training samples are read in the form of arrays of features – normally attribute-value pairs. For the first sample (array) read, a single new node is created below the initial layer of features and connected to each of the attribute values (in the initial layer) constituting the sample (see figure 1a). Thus, the sample is represented by a node linked to each of the attribute values composing the sample.

The same is done for the second sample read, with the difference that if the second sample has any features in common with the first sample, the graph is transformed such that the lattice properties – mentioned above – are preserved. I.e. each subset of attribute nodes would have a unique meet (if it does have a meet) and each group of samples would have a unique join (if it does have a join). In the process intermediate nodes are – created between the attribute-nodes at the top and the sample-nodes at the bottom (see figure 1b). The intermediate nodes are referred to as *concepts*. The nodes above a given node are said to be

Figure 1

spanned by the node and the ones below are *covered* by it. For each set of samples there exists a unique corresponding lattice[1]. (Since the focus in this paper is on the application of lattices to learning, the ideas are discussed with reference to informal examples only. A comprehensive description of the transformation algorithm can be found in [15] and another application is described in [14].)

Let us consider the following example. A number of elephants (see Table 1) are classified on the basis of 4 features each. The attributes are ear size, colour, temperament and love for candy. Figure 2 shows the concept nodes (indicated by '*'-nodes) created during transformation. Notice that this is only a partial graph. Some arcs have been deliberately omitted for the sake of simplicity.

2.2 Similarity-Based Learning

We now explain how the lattice supports generalization and clustering.

Each intermediate node in the lattice is associated with a cluster of features at the top and a cluster of samples at the bottom. Thus, each intermediate node denotes a concept characterized by the attribute values spanned by the node (its intension). Below each intermediate node is a cluster of samples exemplifying the concept (its extension).

The lattice captures all similarities between samples in a series of tangled hierarchies. Stepping bottom-up, each concept covers a larger set of samples. Thus the hierarchies form sequences of concepts of increasing generality (upward). If a specified set of samples has a mutual join, then this join is unique and constitutes a *least generalization* (maximally specific concept) of the set.

2.2.1 Concept Descriptions

Each node can be regarded as an identity (label) for a *class* – the set of samples below it. The *characteristic description* of the class is the set of features spanned by the node. In that sense, the concept nodes represent induced class descriptions or 'rules'. Although the features spanned by a node can be read as a list, the specific tree structure involved reflects dependencies between features. Consequently, each node denotes a rule [17]: if the (unique) meet of a number of features spans any additional features, apart from the given features themselves, then such features are inferred. For example, in figure 2 the meet of PINK, FIERCE and CANDY_LOVER[2] is *10, and *10 spans AFRICAN in addition to the named features, therefore:

PINK and FIERCE and CANDY_LOVER → AFRICAN

Thus, the lattice itself can be used as basis for inference, i.e. as knowledge base. This has the additional effect that training and prediction are integrated, i.e. the system learns incrementally while it operates and there is no distinction between training and test instances. Each instance confirms certain patterns and contradicts other patterns associated with nodes stored in the lattice.

A large number of generalizations are kept in the lattice and evaluated in parallel, treated as embryonic concepts with the potential to be confirmed, or to be contradicted and eliminated (see next section). The lattice structure changes continuously and forms the basis of SBL.

As we mentioned above, GRAND provides *maximally specific* generalizations. When a positive instance of the concept is seen but the current concept definition would classify it as a negative instance, the concept definition is redefined to be the intersection of the current concept definition and the instance [19]. This corresponds to the Wholist strategy [1], also called the One-Sided Algorithm for Pure Conjunctive Concepts [4]. As in Bruner's work, the initial hypothesis is a conjunction of all features in the first positive example. Sarret & Pazzani, however, initialize the hypothesis to be the conjunction of all features in the example description language.

As already intimated, the intermediate nodes represent concepts with associated features. The coincidence of features associated with a concept gives rise to dependencies between the features. The coincidences and dependencies between features in turn give rise to predictability, which forms the basis of inference, and in the case of 'strong' concepts (see next section), to inference rules. Because of this ambiguity in the nature of the intermediate nodes, we will sometimes refer to them as concepts and sometimes as rules, depending on their role in the particular context.

2.2.2 Confidence Factors

Incidental similarities between samples give rise to the creation of a multitude intermediate nodes representing insignificant clusters. To keep the size of the lattice within reasonable limits, a *pruning* strategy is applied [15] whereby insignificant nodes (nodes denoting patterns of features which are not confirmed) are removed on a regular basis. However, if the number of features per sample is large (15+) pruning alone becomes inadequate to curb the

[1] To construct a proper lattice, the attribute nodes have to be connected to a single common superior, and the sample notes to a common subordinate. We omit them since they have no role to play in the conceptual modelling exercise.

[2] For the sake of brevity we write PINK, instead of COLOUR=PINK, etc.

Sample no.	Ear size	Colour	Temperament	Loves_Candy	Class
S1	small	pink	fierce	yes	African
S2	big	pink	friendly	no	Indian
S3	big	grey	fierce	yes	African
S4	small	black	fierce	no	Indian
S5	big	black	fierce	yes	Indian
S6	big	pink	fierce	yes	African
S7	big	black	friendly	no	Indian
S8	small	pink	friendly	no	Indian
S9	big	pink	friendly	yes	Indian

Table 1

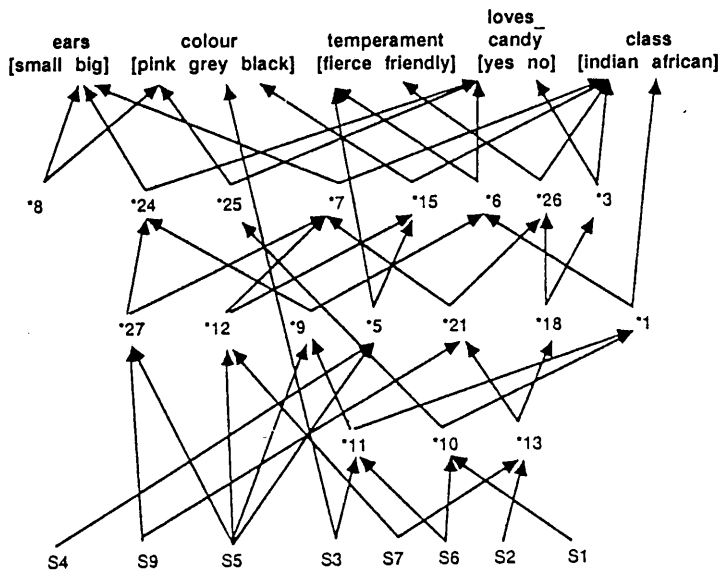

Figure 2

lattice. The lattice size is much better contained if the SBL is directed by EBL (see next section). Pruning is based on a *confidence factor* (CF) associated with every node. The CF of a node is an integer value equal to the number of samples read containing the pattern of attribute values spanned by the node, i.e. the number of sample nodes below it; we refer to it as the *strength* of the node.

By specifying a certain *threshold value*, nodes can be divided into a set denoting established rules and a set representing weaker dependencies between features which may or may not become rules depending on whether they are confirmed or not. (Nodes which are not confirmed, are removed eventually.) This is similar to Lebowitz' idea of *freezing* a feature in conceptual clustering [8]. Lebowitz distinguishes between the confirmation of a feature and of a generalization. The CF of a feature is used to determine whether the feature forms part of a generalization description. The strength of a generalization is used as a measure of interest (or "relevance") to decide when EBL must be applied to a generalization. In other words, the number of instances a generalization represents is an indication of its relevance or "interestingness" (after Lebowitz). Both these notions of confirmation of features and of generalizations are incorporated in the CFs in GRAND. E.g. *15 represents a generalization identified by

BLACK, INDIAN

covering instances S4, S5 and S7. Although *5 represents the generalization

FIERCE, BLACK, INDIAN

the CF of *5 really indicates the number of times FIERCE occurred in conjunction with BLACK and INDIAN. If this happens often enough, *5 would be frozen, thus identifying FIERCE, BLACK, INDIAN as a useful or permanent generalization with

BLACK, INDIAN

as a more general generalization. Otherwise, *5 would be deleted, leaving the latter generalization only. The same would happen in UNIMEM by changing the CF of the feature FIERCE.

2.3 Explanation-Based Learning

In Section 2.2 we explained how SBL is supported by the lattice. We now consider EBL, which is manifested as follows. Confirmed rules are regarded as forming part of the domain theory which has evolved. Most of these rules express dependencies between features. In other words, they are not part of the description of the concept to be learnt. E.g. in figure 2, the node *9 denotes the rule

BIG and FIERCE → CANDY_LOVER

which does not form part of the descriptions for the classes INDIAN or AFRICAN. Since there are many nodes in the lattice denoting such rules, it implies that the lattice contains a rich theory of the domain. (Notice that because of the existence of *24 and *6, the following inferences are inhibited:

FIERCE and CANDY_LOVER → BIG

BIG and CANDY_LOVER → FIERCE)

The intermediate nodes can also be viewed in a slightly different way. An intermediate node which spans a pattern which occurred many times, can be viewed as an eminent concept, identified by the system, but which is not listed among the recorded features. The detection of such concepts has been studied under the heading of *feature construction* [9]. If questioned, a domain expert (or *oracle* [12]) might be able to associate a lexical term with it. In graph terms: such a node may be viewed as being linked to an additional feature, not present among the given features. E.g. let us again consider *9 in figure 2. Considering the fact that these elephants are fierce, big-ear, candy loving elephants, *9 could easily be linked to a *new* feature SPOILT_ELEPHANT. *9 would thus denote the rule

BIG and FIERCE and CANDY_LOVER →
SPOILT_ELEPHANT.

This interpretation of the graph differs from the one above in the sense that here *9 suggests the existence of an autonomous concept, defined by the features spanned by the node *9. In other words, the node *9 denotes a concept which constitutes the right-hand side of a (domain theory) rule which has as its left-hand side the features spanned by the node itself. This is the reason why the intermediate nodes were referred to as concepts in the first place (see section 2.1). Thus, each (confirmed) concept node can be visualized as having an arc to a corresponding lexical label (feature) at the top.

If a subset of features of a sample is spanned by a particular concept, then it means that the sample satisfies the rule describing the concept. And if this concept in turn implies the target concept (i.e. the concept being learnt – INDIAN in this case), then these rules explain why the sample is an instance of the target concept. Thus, when a new sample is read, it is first determined whether it contains one or more subpatterns of features (which may overlap) which are spanned by eminent concepts in the lattice[3]. This constitutes the first step of explanation-based learning, i.e. explaining why the sample is an instance of the concept being learnt.

E.g. let us say the following sample is added to the lattice in figure 2

S10: BIG, GREY, FRIENDLY, NON-CANDY_LOVER, INDIAN

Let us say *13 denotes the concept GOOD_ELEPHANT, described by the features[4]

BIG, FRIENDLY, NON-CANDY_LOVER.

The fact that *13 is below (i.e. a subset of) INDIAN, implies that all GOOD ELEPHANTS are INDIAN elephants. Consequently, the fact that S10 satisfies the conditions for GOOD_ELEPHANT explains its being an instance of INDIAN.

The second step involves using the domain rules to identify important and irrelevant features of the sample with the aim of generalizing it. If there are any of the sample's features that are not spanned by eminent concepts, they are likely to be irrelevant – especially if we are talking of

[3] This happens implicitly as part of the transformation algorithm

[4] INDIAN is omitted from this list since in the example the concept is being learned.

lattices containing several hundreds of concepts. Thus the sample is not connected to such features, which effectively removes a part of the sample description and thereby makes it more general (see Section 2.5.2). In other words, we accomplished explanation-based generalization.

Since GREY, in the above example, is not spanned by eminent concepts, it is dropped, and the sample is generalized by connecting it to *13 only.

This happens implicitly as part of the transformation algorithm.

INDIAN is omitted from this list since in the example it is the concept being learnt.

Thus, while SBL is in progress, the domain theory evolves – in the form of implicit rules in the lattice. As soon as these are confirmed adequately, they begin to play a role in EBL. The benefit of this type of learning is that knowledge necessary to perform EBL can be acquired by the learning system and the learner gets better at learning [18].

2.4 External Rules

In the previous section we considered a domain theory that was developed internally from scratch. Domain theory can be added to the lattice in the form of rules supplied by an external source. Rules are treated just like samples – ignoring the separation between antecedent and consequent parts but associating extraordinary high confidence factors with the rule nodes. Such rules are then automatically incorporated in the learning process, just as the internally generated ones are. Thus:

- samples read are generalized by means of both SBL and EBL [18], but simultaneously
- the domain theory is generalized or specialized through SBL and
- new domain theory is developed through SBL.

Pazzani citepazzani refers to specially selected examples inserted with the specific aim of establishing the domain theory as "foundational examples".

2.5 Effect of EBL on the lattice size

SBL has the effect of generating intermediate nodes and EBL has the effect of removing them (referred to as *operationalization*) [23]. Although the above weak form of EBL does not remove nodes from the graph, it does prevent nodes from being created and thus enables us to contain the lattice. E.g. Let us say a concept *1 represents an adequately confirmed pattern $< A, B, C, D >$ (see figure 3a) and a pattern $< A, B, D, F >$ is read as part of a sample S_n, where C and F are different values of the same attribute. If normal transformation takes place, a new node *4 is created to represent the new pattern of features (see figure 3b). However, if EBL takes place, *1 is identified as the meet of $< A, B, D >$. Since *1 denotes a strong rule, we accept that

$$A \text{ and } B \text{ and } D \rightarrow C.$$

Thus, C is inferred and F ignored and no new nodes are created (see figure 3a).

3 Related Work

Although various related systems have been referenced already, we now focus on some of them specifically.

3.1 UNIMEM

As we mentioned before, GRAND extends the ideas implemented in UNIMEM [8] – the main difference being that UNIMEM uses a hierarchy where GRAND uses a lattice as framework for knowledge representation. Lebowitz [7] explains how external rules can be used to apply EBL to concepts, in particular to establish causal relationships between features. To apply EBL methods to UNIMEM, rules have to be supplied that capture the initial understanding of the domain. In UNIMEM this is done with implications that capture hypothesized low-level causal connections among features. Rules (external from the hierarchy and hand coded) can be used to explain the presence of one feature from the presence of another feature. With the initial rules UNIMEM can engage in EBL with the purpose of analyzing a generalization done by SBL. The relevance of a generalization determine whether EBL will be applied to it [7]).

In GRAND, such low-level rules can be added to the concept-base (lattice) itself and the explanations can be derived in a similar way. However, the kind of low-level dependencies added by Lebowitz as domain theory, are normally the first kind of dependencies which quickly evolve in the lattice. In other words, most of them would be there already. Like OCCAM [18], (but unlike UNIMEM) GRAND can thus be regarded as a "closed-loop" learning system where the same memory is used for SBL and EBL.

Secondly, UNIMEM stores features as 'flat' lists associated with generalizations. In GRAND, each feature is involved in a complex graph structure reflecting its dependence on other features or groups of features, as well as features with which it jointly determines other features. In other words, whereas UNIMEM has to apply an algorithm to derive explanations, in GRAND they can be read from the graph. Since the lattice stores all the low level dependencies between predictive and predictable features, the explanation can be derived by a straightforward interpretation of the graph structure. In summary, UNIMEM requires a separate EBL procedure to identify causal relationships from features contained in generalizations, while in GRAND the EBL takes place implicitly and the causal relationships are there to be read. UNIMEM uses the confidence factor of a generalization as a measure of its interestingness and uses this to determine which generalizations to explain. In GRAND every generalization can be explained.

3.2 OCCAM

OCCAM [18] also integrates SBL and EBL in one system. The main differences between GRAND and OCCAM are

- Pazzani's emphasis on knowledge-intensive strategies,
- the knowledge representation used, and
- the differentiation between Theory-Driven Learning and Explanation-Based Learning – a result of his use of Conceptual Dependency Theory.

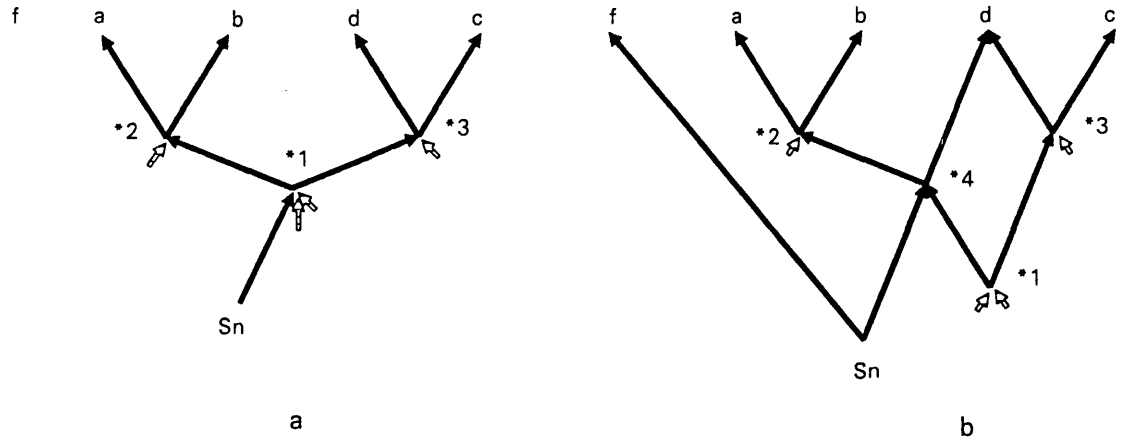

Figure 3

OCCAM starts with an initial hierarchy of schemata which represents the conceptual dependency actions, goal and states; as OCCAM learns, the hierarchy is extended by creating specializations of the existing schemata. Thus, OCCAM incrementally forms a concept hierarchy that explains and organizes previous experiences [18].

Pazzani puts much emphasis on the fact that the more knowledge-intensive strategies have stronger justifications and should receive priority in learning. (Empirical techniques justify the inclusion of a feature in a schema on the simple basis that it has appeared in previous events.) Since EBL is more knowledge-based whereas SBL is more statistics- based, there is merit in the argument that EBL should be given priority. However, although the argument about the superiority of knowledge intensive learning is a valid one, it has to be remembered that, ultimately, knowledge has to come from somewhere – this role of extracting knowledge from data is fulfilled by SBL.

3.3 Analytical and Similarity-based Classification

Vilain et al [22] present an exposition on the role of classification (both analytical and similarity-based) in Knowledge Representation and Machine Learning. They contend that pre-inserted domain knowledge is essential "in providing an inductive bias to the learning procedure, thereby shortening the required training phase, and reducing the brittleness of induced generalizations". Since KL-ONE, the knowledge representation scheme used by them, is also lattice based, it is no surprise that there is a high degree of correspondence with GRAND. Vilain et al divide features characterizing generalizations ('generalization frames') into *definitions*, and *norms* which are interpreted as defaults. There is a direct correspondence between their definitions and the features spanned by a node in GRAND. Similarly, their norms correspond to the features spanned by nodes immediately below a given node in GRAND. E.g. the definition of *15 (see figure 2) would be

GREY and NEGATIVE

and its norms would be the nodes spanned by *5 and *12 not spanned by *15, namely

BLUE and TALL.

In GRAND the norms are either inferred probabilistically or treated as proper defaults as in other knowledge representation paradigms. Depending on which interpretation is adopted, additional features spanned by S4, S5 and S7 (which are not immediate children of *15) might also be regarded as norms.

Since concepts in GRAND are (implicitly) not subject to the canonical form imposed on the analytical language of Vilain et al, GRAND is more expressive. GRAND also utilizes a different classification method. But for the rest there is a remarkable correspondence in functionality between the two approaches.

3.4 Incremental Version Space Merging

Hirsch [5] uses the *version space* paradigm [11] to combine empirical and analytical learning. The central idea is to apply explanation-based generalization to training data, and then to do empirical learning on the generalized data. Thus, rather than updating the version space (doing empirical learning) with single instances, each instance has the effect of multiple instances. Since the version space is also incorporated in the lattice, Incremental Version Space Merging and GRAND exhibits similar behaviour along a spectrum from knowledge-poor to knowledge-rich learning.

4 Conclusion

"A central activity of science is the search for unifying principles that account for apparently diverse phenomena within a single framework" [6] . The lattice model goes some way toward achieving that goal. It has been shown elsewhere [16] that the lattice supports supervised and unsupervised learning, rule-based and case-based, incremental and non-incremental learning. In this paper we showed that both SBL and a weak form of EBL are supported.

We explained that Graph Induction does not involve separate SBL and EBL procedures. SBL and EBL are integrated as follows: intermediate SBL results – tentative domain theory – are used as part of an EBL process which is

in turn used to guide SBL Furthermore, SBL and EBL continuously re-validate and extends the already constructed theory. Instead of separate SBL and EBL phases the two processes operate in parallel and in complete synergism. The result is that each kind of learning gains the maximal effect from the intermediate results of the other. Also EBL help control the size of the lattice, expanded as a result of SBL.

References

[1] J.S. Bruner, J.J. Gutnow, and G.A. Austin. *A Study of Thinking*. Wiley, New York, 1956.

[2] D. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2), September 1987.

[3] J-G. Ganascia. CHARADE: A rule system learning system. In *Tenth International Joint Conference on Artificial Intelligence, Volume 1*, 1987.

[4] D. Hausler. Applying Valiant's learning framework to AI concept learning problems. Technical report no. ucsc-crl-87-11, Department of Computer Science, University of California at Santa Cruz., 1987.

[5] H. Hirsch. Combining empirical an analytical learning with version spaces. In A.M. Segre, editor, *Sixth International Workshop on Machine Learning*, pages 29–33, New York, 1989. Morgan Kaufmann.

[6] P. Langley. Unifying themes in empirical and explanation-based learning. In A.M. Segre, editor, *Sixth International Workshop on Machine Learning*, pages 2–4, New York, 1989. Morgan Kaufmann.

[7] M. Lebowitz. Integrated learning: Controlling explanation. *Cognitive Science*, 10(2):219–240, 1986.

[8] M. Lebowitz. UNIMEM, a general learning system: An overview. In *European Conference on Artificial Intelligence*, pages 32–41. Brighton, 1986.

[9] J.C. Matheus. Adding domain knowledge to SBL through feature construction. In *AAAI-90*, Boston, 1990.

[10] R.S. Michalski, I. Mozetic, J. Hong, and N. Lavrac. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In *AAAI 86*, Philadelphia, USA, 1986.

[11] T.M. Mitchell. Generalization as search. In B.L. Webber and N.J. Nilsson, editors, *Readings in Artificial Intelligence*. Morgan Kaufman, Los Altos, 1985.

[12] S. Muggleton. Duce, an oracle based approach to constructive induction. In *IJCAI-87*, pages 287–292. Kaufmann, 1987.

[13] G.D. Oosthuizen. Graph induction. In *First European Knowledge Acquisition for Knowledge-based Systems Workshop*, Reading, UK, 1987.

[14] G.D. Oosthuizen. The use of a lattice for fast pattern matching. *South African Computer Journal*, (1), 1990.

[15] G.D. Oosthuizen. Lattice-based knowledge discovery. In *AAAI-91 Workshop on Knowledge Discovery in Databases*, Anaheim, USA, 1991.

[16] G.D. Oosthuizen and C. Avenant. Lattice-based conceptual clustering. In preparation, 1991.

[17] G.D. Oosthuizen and D.R. McGregor. Induction through knowledge base normalisation. In *European Conference on AI*. Pitman Publishing, 1988.

[18] M.J. Pazzani. Integrating explanation-based and empirical learning methods in OCCAM. In *Third European Working Session on Learning*. Pitman Publishing, 1988.

[19] Sarrett and Pazzani. One-sided algorithms for integrating empirical and explanation-based learning. In A.M Segre, editor, *Sixth International Workshop on Machine Learning*, pages 26–28, New York, 1989. Morgan Kaufmann.

[20] A.M. Segre. *Procs of the Sixth International Workshop on Machine Learning*. Morgan Kaufmann, New York, 1989.

[21] P.E. Utgoff. ID5: an incremental ID3. In *Fifth International Conference on Machine Learning*, pages 107–120. Morgan Kaufmann, 1988.

[22] M. Vilain, P. Koton, and M.P. Chase. On analytical and similarity-based classification. In *AAAI-90*, Boston, 1990.

[23] J. Wogulis and P. Langley. Improving efficiency by learning intermediate concepts. In *Proceedings of Eleventh International Joint Conference on Artificial Intelligence*, Detroit, 1989.

Acknowledgements: We want to thank the referees for their comments. We are indebted to Prof Roelf van den Heever for providing us with the opportunity to conduct the current research, and to Jesus Christ the Origin of Knowledge.

Notes for Contributors

The prime purpose of the journal is to publish original research papers in the fields of Computer Science and Information Systems, as well as shorter technical research papers. However, non-refereed review and exploratory articles of interest to the journal's readers will be considered for publication under sections marked as a Communications or Viewpoints. While English is the preferred language of the journal papers in Afrikaans will also be accepted. Typed manuscripts for review should be submitted **in triplicate** to the editor.

Form of Manuscript

Manuscripts for review should be prepared according to the following guidelines.

- Use double-space typing on one side only of A4 paper, and provide wide margins.
- The first page should include:
  - title (as brief as possible);
  - author's initials and surname;
  - author's affiliation and address;
  - an abstract of less than 200 words;
  - an appropriate keyword list;
  - a list of relevant Computing Review Categories.
- Tables and figures should be on separate sheets of A4 paper, and should be numbered and titled. Figures should be submitted as original line drawings, and not photocopies.
- Mathematical and other symbols may be either handwritten or typed. Greek letters and unusual symbols should be identified in the margin, if they are not clear in the text.
- References should be listed at the end of the text **in alphabetic order** of the (first) author's surname, and should be cited in the text in square brackets. References should thus take the following form:

  [1] E Ashcroft and Z Manna, [1972], The translation of 'GOTO' programs to 'WHILE' programs, *Proceedings of IFIP Congress 71*, North-Holland, Amsterdam, 250-255.

  [2] C Bohm and G Jacopini, [1966], Flow diagrams, Turing machines and languages with only two formation rules, *Comm. ACM*, **9**, 366-371.

  [3] S Ginsburg, [1966], *Mathematical theory of context free languages*, McGraw Hill, New York.

Manuscripts *accepted* for publication should comply with the above guidelines, and may be provided in one of the following formats:

- in a **typed form** (i.e. suitable for scanning);
- as an **ASCII file** on diskette; or
- as a **WordPerfect**, **TEX** or **LATEX** or file; or
- in **camera-ready** format.

A page specification is available on request from the editor, for authors wishing to provide camera-ready copies. A styles file is available from the editor for Wordperfect, **TEX** or **LATEX** documents.

Charges

Charges per final page will be levied on papers accepted for publication. They will be scaled to reflect scanning, typesetting, reproduction and other costs. Currently, the minimum rate is R20-00 per final page for camera-ready contributions and the maximum is R100-00 per page for contributions in typed format.

These charges may be waived upon request of the author and at the discretion of the editor.

Proofs

Proofs of accepted papers will be sent to the author to ensure that typesetting is correct, and not for addition of new material or major amendments to the text. Corrected proofs should be returned to the production editor within three days.

Note that, in the case of camera-ready submissions, it is the author's responsibility to ensure that such submissions are error-free. However, the editor may recommend minor typesetting changes to be made before publication.

Letters and Communications

Letters to the editor are welcomed. They should be signed, and should be limited to about 500 words.

Announcements and communications of interest to the readership will be considered for publication in a separate section of the journal. Communications may also reflect minor research contributions. However, such communications will not be refereed and will not be deemed as fully-fledged publications for state subsidy purposes.

Book reviews

Contributions in this regard will be welcomed. Views and opinions expressed in such reviews should, however, be regarded as those of the reviewer alone.

Advertisement

Placement of advertisements at R1000-00 per full page per issue and R500-00 per half page per issue will be considered. These charges exclude specialized production costs which will be borne by the advertiser. Enquiries should be directed to the editor.

South African Computer Journal

Number 6, March 1992
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif

Nommer 6, Maart 1992
ISSN 1015-7999

Contents