South African
Computer
Journal
Number 8
November 1992

Suid-Afrikaanse
Rekenaartydskrif
Nommer 8
November 1992

Computer Science and Information Systems

Rekenaarwetenskap en Inligtingstelsels

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

Die Suid-Afrikaanse Rekenaartydskrif
'n Amptelike publikasie van die Rekenaarverenging van Suid-Afrika 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: +27 (12) 420-2504 Fax: +27 (12) 43-6454

Sigma Press

Guest Contribution

The paper below was given as an invited address by Prof Roode at the July 1992 Conference of the South African Computer Lectures' Association. (Editor)

The Ideology, Struggle and Liberation of Information Systems

Dewald Roode
Department of Informatics, University of Pretoria

In 1989, Denning *et al* presented the final report of the Task Force on the Core of Computer Science in an article entitled "Computing as a Discipline" [3]. This was said to present a new intellectual framework for the discipline of computing and proposed a new basis for computing curricula.

In the words of the authors, "an image of a technology-based discipline is projected whose fundamentals are in mathematics and engineering." Algorithms are represented as the most basic objects of concern and programming and hardware design as the primary activities. Although there is wide consensus that computer science encompasses far more than programming, the persistent emphasis on programming "arises from the long-standing belief that programming languages are excellent vehicles for gaining access to the rest of the field" [3].

The new framework sets out to present the intellectual substance of the field in a new way, and uses three paradigms to provide a context for the discipline of computing. These paradigms are *theory*, rooted in mathematics; *abstraction*, rooted in the experimental scientific method and *design*, with its roots in engineering.

Programming, the report recommends, should still be a part of the core curriculum and programming languages should be seen and used as vehicles for gaining access to important aspects of computing.

The following short definition is offered of the discipline of computing [3]:

> The discipline of computing is the systematic study of algorithmic processes that describe and transform information: their theory, analysis, design, efficiency, implementation, and application. The fundamental question underlying all of computing is, *"What can be (efficiently) automated?"*

In the same issue of Communications, tucked away towards the end of the journal, an article by Banville and Landry asked the innocent question "Can the Field of MIS be disciplined?" [1]. It is not clear whether the use of the word "discipline" in both articles was purely coincidental – however, the implications were quite clear: computer science was able to talk about "computing as a discipline," and indeed, could present a report which, in a sense, was a culmination of more than twenty years' efforts. Yet, its sister discipline was still asking questions of a very introvertive nature about itself.

It has become quite clear that the fields (leaving aside for the moment the questions of "disciplines") of computer science and information systems (or MIS, informatics, or whatever other name we want to attach to it) have different aims and objectives, different problems that confront it, and, yes, if we want to be truly scientific, different paradigms. To support the latter statement, it is sufficient to contrast the three paradigms of computing with the four paradigms of information systems development described by Hirschheim and Klein [5]. It can be said that a central activity in information systems is the development of information systems, and that therefore, these paradigms have implications for the field of information systems. The four paradigms can be characterized briefly, as follows:

- The analyst as systems expert
- The analyst as facilitator
- The analyst as labour partisan
- The analyst as emancipator or social therapist.

In the same spirit, Lyytinen sees the "systems development process as an instrument in organizational change" [6] and remarks that analysts' principal problems are "in understanding the goals and contents of such change instead of solving technical problems." Already in 1987 Boland [2] observed that: "designing an information system is a moral problem because it puts one party, the designer, in the position of imposing an order on the world of another."

This is clearly a far cry from Denning *et al*'s statement that the fundamental question is "what can be automated?" At the same time, within the context of the field of computing, there is nothing wrong with this question, and it is probably the right question for practitioners of computing to continually ask themselves. But it is a disastrous question for a practitioner of informatics to ask. And it has taken us quite a long time to realise this – that the two disciplines have fundamentally different roles to play. These roles are complementary and supportive, and not destructively opposed.

The liberation of information systems lies in realising this elemental truth: that information systems are man-made objects designed to effect organisational change and that, as such, they can ill be studied using the paradigms of abstraction and engineering mentioned above.

What then is needed? Banville and Landry offer the consolation that we need not concern ourselves too much about the lack of discipline, and that we can indeed even pride ourselves in being a fragmented adhocracy. It is, in fact, even healthy to continue in all sorts of directions. During this process of finding itself, a discipline should be allowed a considerable degree of latitude, and many avenues should be explored. This obviously makes the field of information systems extremely exciting: it is in the process of discovering remarkable truths, discovering that there are in reality people out there using the systems which analysts design and build, and that the most intriguing problems centre around the role of people in all of this: the analyst, the user, their interaction, the impact of sytems on the work lives of workers on all levels, the impact on organizations. These are questions which have mostly been ignored or lightly treated over the years, but which have emerged as *the* problems to be solved. We do not have the tools to solve them – not yet; but a good starting point would certainly be to first understand more about our field and its research tools, for the empirical, positivist approach so often employed will not suffice to solve the above problems.

In the spirit of contributing to the liberation movement of information systems, we have embarked on a study of research on research in Information Systems, and will report on the results more fully in the near future. We define Information Systems as follows [4]:

> Information Systems is an inter-disciplinary field of scholarly inquiry, where information, information systems and the integration thereof with the organisation is studied in order to increase the effectiveness and efficiency of the total system (of technology, people, organisation and society).

In Information Systems then, we see the fundamental question underlying the entire discipline, to be the problem of balancing the need to contribute, through information systems, to the achievement of the mission of the organisation with the moral responsibility to develop and implement socially accepted information systems.

Each of the fields, computer science and information systems, benefits enormously from the activities of the other. Nonetheless, we must recognize the different approaches used by the two disciplines and allow them to complement each other. It should not be our business to convince one another that the universal truth is that which we use in our discipline – whether that be computer science or information systems. Instead, we should seek out the opportunities for synergy, and for complementing each other. If we succeed in doing this at SACLA, then we could indeed do ourselves proud.

References

1. C Banville and M Landry. 'Can the field of MIS be disciplined?'. *Communications of the ACM*, **32**(1):48–60, (1989).
2. R J Boland and R A Hirschheim, eds. *Critical Issues in Information Systems Research.* John Wiley & Sons Ltd., 1987.
3. P J Denning, D E Comer, D Gries, M C Mulder, A Tucker, A J Turner, and P R Young. 'Computing as a discipline'. *Communications of the ACM*, **32**(1):9–23, (1989).
4. N F Du Plooy, L D Introna, and J D Roode. 'Notes on research in information systems'. Unpublished research report, Department of Informatics, University of Pretoria, (1992).
5. R Hirschheim and H K Klein. 'Four paradigms of information systems development'. *Communications of the ACM*, **32**(10):1199–1216, (1989).
6. K Lyytinen. 'New challenges of systems development: A vision of the 90's'. *Data Base*, pp. 1–12, (Fall 1989).

Editor's Notes: To Compete or Collaborate

Human interaction invariably brings with it a blend of competition and collaboration. Competition means that one enjoys the exhilaration of winning while the other endures the shame of loosing. Because of this reward/punishment mechanism, it is a widely assumed that competition enhances performance and efficiency. This dogma pervades not only commerce, sport and politics, but is found in practically all areas of human endeavour, including research.

The competitive spirit in research is found in the well-known saga of Watson and Crick racing to unravel the double helix structure of DNA. Not so well-known, though equally illustrative, is the intensity of Newton's stratagems to oust Leibnitz from receiving any credit for differentiation. Recently there have been reports of scientists who have either tolerated or manufactured fraudulent results in order to win some or other scientific race. The space race, the arms race, the race for an AIDS cure, the scurry for faster smaller hardware, the race for awards, the drive for publications, Nobel prizes: all of this attests to a profoundly competitive international research culture.

But while competition might be the handmaiden of commerce and sport, it is the harlot of research – an unfortunate concomitant of the silly side of human nature. The archetypal researcher not only rises above the incidentals of human accolades; he disdains them. By tradition, the definitive research qualification is a PhD – a Doctor of Philosophy – a lover of thought. Discovery and thought are not only by their very nature rewarding, they are also humbling. When the archetypal researcher moves outside his interior thought-world, it is to share his discoveries. If he is childish, it is not the little boy flexing his biceps and saying: "I'm stronger than you" but the child rushing to tell everyone: "Wow – look at this!" He is forgetful of self: Pythagoras, oblivious of the invading enemy and his impending death while he researches in the sand; Archimedes shouting "Eureka" without care for his nudity. The competitive spirit is a crass intrusion into this ancient legacy of innocence and selflessness.

By its nature, collaboration thrives in a climate of easy social intercourse. It may initially feel uncomfortable for researchers, who are inclined to be socially inept and are wont to bury themselves in work away from society. However, once the plunge to collaborate is taken there is ample evidence that it leads to successful research. In maximizing the use of available talent, it brings about a synergy in which two heads are better than one. All participants enjoy its rewards and no individual has to endure the full weight of its failures. In fact, the notion of collaboration is now so commonplace that significant research seems impossible without it. The tendency, however, is to encourage research collaboration within an organisation, but to emphasize competition in relation to outside organisations.

During a forum discussion at the July South African Computer Lecturers' Association (SACLA) conference, an appeal was made for greater collaboration between universities. Not surprisingly, the information technology disciplines at local universities have always had both a competitive and a collaborative relationship. The competitiveness usually takes the form of friendly rivalry, while the very existence of SACLA bears testimony to a rather unique collaborative relationship. In latter years the competitiveness seems to have intensified, while electronic mail and other developments have improved the prospects for collaboration. At issue, then, is whether there is an imbalance between these dual forces. The appeal at the SACLA forum implied that there is, and I would strongly agree. It is my view (my prejudice, if you will) that competition between universities is a self-indulgent and wasteful dissipation of energy.

Those who are inclined to compete should seriously examine what is to be gained. It is unconvincing to argue that winning makes a significant impact on the way in which students select universities: in the main, this is a matter of geography and language preference. To some extent, the same might be said about staff, although research reputation perhaps plays a more important role here. Neither are research funding agencies (e.g. the FRD) influenced by whether X is "better" in some or other sense than Y. On the contrary, it has wisely been decided to fund on the basis of criteria that are believed to be objective, without any reference whatsoever to the performance of competitors. True enough, funds are limited, but it is precisely for this reason that it is wasteful to divide the little there is between divergent research efforts.

It seems to me that there is a wealth of research talent out there, but that each researcher selects an area of interest almost as a matter of whim. There is an urgent need for well-coordinated collaboration on focussed research areas that have been carefully selected as directly relevant to the country. It is especially incumbent on those who finance, manage and lead research to identify such areas and to encourage collaboration in every possible way.

I look forward to the manifestation of such collaboration in SACJ publications authored by researchers from different university departments. To date there have been none of consequence. If we fail to collaborate, we are in danger of becoming little Don Quixotes who spend our lives attacking windmills and defending castles of xenophobia and irrelevance.

Beam Search In Attribute-based Concept Induction

Hendrik Theron Ian Cloete
Department of Computer Science, University of Stellenbosch, Stellenbosch 7600

Abstract

This paper investigates the issues of specializing only a single best conjunction to employing a beam search when learning attribute-based concept descriptions using the GCA algorithm. We describe GCA, a recently introduced generic learning algorithm which generalizes a number of well-known learning algorithms like CN2 and AQ. It is shown, using ten test domains, that concept descriptions found by a beam search are seldom more accurate than those found by specializing only a single best conjunction. In addition, the former descriptions are usually more complex than the latter and in some cases even considerably more so. This result holds even when more stringent pruning is applied during a beam search. Since specializing only one conjunction is computationally much less demanding than specializing a set of alternative best conjunctions, the result is that GCA need not employ a beam search in order to find good descriptions.

Keywords: *Learning from examples, beam search, pruning.*
Computing Review Categories: *I.2.5, I.2.6*

Received April 1992, Accepted June 1992.

1 Introduction

Concept learning programs aim to induce from a given set of training instances concept descriptions that will have the highest prediction accuracy on unseen instances. Algorithms like CN2 [3], AQ [4] and GCA (Generic Covering Algorithm) [7] learn concept descriptions following a general-to-specific search. Conjunctions describing concepts are constructed by specializing a conjunction until it becomes consistent (i.e. covers only instances belonging to the concept being learned) or until some pruning criterion terminates further specialization. The specialization process is essentially a best-first hill-climbing search. To reduce the chance of being mislead by one bad specialization step, the above mentioned algorithms employ a *beam search*. In this approach a user specified number k of current best conjunctions are specialized instead of only the single best conjunction (i.e. $k = 1$). Only the k best conjunctions are kept for the next specialization step.

Employing a beam search is considerably more time consuming than specializing only a single best conjunction. Thus to be useful a beam search should increase the accuracy of concept descriptions, or if the accuracy is very similar, at least significantly reduce the complexity of the descriptions.

This paper presents empirical investigations in ten test domains which show that employing a beam search rarely improves the accuracy or reduces the complexity of concept descriptions. In fact, a beam search often increases the description complexity. It is shown that even when more stringent pruning is applied during beam search the generated descriptions are rarely significantly less complex than those found without a beam search. The vehicle for our experiments is the recently introduced GCA algorithm. GCA was chosen because it generalizes both CN2 and AQ and obtained similar or higher accuracy than both these algorithms [7].

The paper has the following outline: GCA is described in the next section. The experimental method and test domains are described in section 3. In section 4 experiments are described where the best descriptions found with $k = 1$ are compared with those found for $k = 10$ and $k = 20$. Section 5 and 6 compare the descriptions found with $k = 1$ with those found with a beam search when more stringent pruning is performed. We close with a summary of the results and a conclusion.

2 A Generic Covering Algorithm

This section describes the version of GCA that was used for the experiments reported below. The ability of GCA to efficiently generate (almost) most general concept descriptions by avoiding all useless specializations of a conjunction is a major improvement of both AQ and CN2. It is also the only covering algorithm to employ a stop-growth criterion (described below) which enables it to generate very simple descriptions in noisy domains.

Table 1 contains an example of the input to GCA and the concept descriptions generated by it. Table 2 contains the GCA algorithm. The top-level loop generates conjunctions describing each concept until all instances belonging to it are covered or until a null conjunction is returned (due to pruning). The FindBestConj procedure implements a beam search (note by setting k to 1 only a single best conjunction is specialized). Pruning takes place in three stages. The significance test (step (2)) compares the distribution of instances covered by a conjunction with that of the complete training set. It eliminates specializations whose information do not differ significantly from that of the training set. We employed the log likelihood ratio test as significance test [2]. Only significant conjunctions are

Table 1. An example of the input and output of GCA

Description of training instances

Attributes:

attribute	type	domain
outlook	nominal	{sunny,overcast,rain}
autumn	nominal	{yes,no}
temp	linear	{15..35}

Concepts to learn:

stop-raining-tomorrow = yes
stop-raining-tomorrow = no

Instances:

#	outlook	autumn	temp	stop-raining
1	sunny	yes	17	no
2	overcast	no	18	no
3	rain	yes	16	no
4	sunny	yes	22	no
5	sunny	no	29	no
6	overcast	yes	30	no
7	overcast	no	35	no
8	rain	yes	23	no
9	rain	no	27	no
10	sunny	yes	28	yes
11	overcast	no	23	yes
12	sunny	no	27	yes
13	rain	no	23	yes

Rules generated by GCA

[outlook ∈ {overcast,sunny}][22<temp≤28]	⇒	yes
[autumn = no][temp = 23]	⇒	yes
[temp ≤ 22]	⇒	no
[temp > 28]	⇒	no
[autumn = yes][temp ≤ 27]	⇒	no
[outlook = {overcast,rain}][temp > 23]	⇒	no

Default: stop-raining-tomorrow = **no**

Table 2. GCA

```
PROCEDURE GCA ($T$ : training_set,$k$ : beam_width);
  rule_set := $\emptyset$;
  FOR each concept $C_i$ DO
    $P$ := instances belonging to $C_i$ and $N := T - P$;
    FOR each attribute value or interval $a_i$ DO
      determine $X_P(a_i)$ and $X_N(a_i)$;                              (1)
    REPEAT
      new_conj := FindBestConj($P$,$N$,$k$);
      IF new_conj = NULL THEN
        $P := \emptyset$
      ELSE
        $P := P - X_P$(new_conj);
        rule_set := rule_set ∪ {new_conj}
      END {IF}
    UNTIL $P = \emptyset$;
  END; {FOR}
  RETURN rule_set
END GCA;

PROCEDURE FindBestConj ($P$,$N$,$k$);
  best_conj := NULL;
  alternatives := {most general conjunction with
                   $X_P = P$ and $X_N = N$};
  WHILE alternatives $\neq \emptyset$ DO
    alternatives := GenerateSpecializations(alternatives);
    FOR each conjunction $c \in$ alternatives DO
      IF Significant($c$) THEN                                          (2)
        IF Better($c$,best_conj) THEN best_conj := $c$;                 (3)
      IF StopGrowth($c$) THEN
        alternatives := alternatives - {$c$}                            (4)
    END; {FOR}
    Retain only $k$ best conjunctions in alternative_set
  END; {WHILE}
  IF Uninformative(best_conj) THEN
    best_conj := NULL;                                                  (5)
  RETURN best_conj
END FindBestConj;

PROCEDURE GenerateSpecializations(alternatives);
  new_alternatives := $\emptyset$;
  FOR each $c \in$ alternatives DO
    RemoveUselessValues($c$.usable,$c$.excluded);                       (6)
    FOR each value $a_i$ in $c$.usable DO
      $c'$ := $c$ specialized by removing $a_i$ from it;
      $X_P(c') := X_P(c) - X_P(a_i)$;
      $X_N(c') := X_N(c) - X_N(a_i)$;
      $c'$.usable := $c$.usable - {$a_i$};
      $c'$.excluded := $c$.excluded ∪ {$a_i$};
      new_alternatives := new_alternatives ∪ $c'$
    END
  END;
  Remove duplicate expressions form new_alternatives;
  RETURN new_alternatives
END GenerateSpecializations;
```

compared with the current best specialization (step (3)). The best conjunction is selected as the one with the highest value for the Laplace error estimate [2]. A form of pruning novel to GCA is the stop-growth criterion (step (4)). This function compares the distribution of instances covered by a conjunction with that of its immediate predecessor. It thus measures whether there is a significant difference between a conjunction and its specialization. If not, all further specialization of the conjunction is terminated. The log likelihood ratio test was also used as stop-growth criterion. The last pruning step (step (5)) terminates conjunction generation when the Laplace error estimate for the instances covered by a conjunction is lower than that for the concept in the training set. This prevents rules with too low accuracy.

The last and most distinguishing part of GCA is its specialization procedure. For each numerical attribute value in the training set intervals of the form $(A \leq a_i)$ and $(a_i < A)$ are created. Let P denote the (*positive*) instances belonging to the concept being learned and let N denote the remaining (*negative*) instances. For each attribute value and interval a_i its extension in P and N ($X_P(a_i)$ and $X_N(a_i)$) is the subset of P or N which take that value or satisfy that interval (step (1)). Similarly the extension of a conjunction in a set is those instances which match the conjunction. At the start, the most general conjunction is simply that conjunction where each attribute takes all its possible values. At each specialization step, a conjunction is specialized by removing a single value or interval from it. The specialization process is implemented by maintaining for each conjunction its own `usable` and `excluded` sets. Initially, `usable` contains all the attribute values and

Table 3. The ten test domains

#Instances	#Attributes	#Classes	References
	Lymphography		
148	18	4	[5]
	Breast cancer		
286	9	2	[5]
	Primary tumor		
339	17	22	[5]
	Hepatitis		
157	19	2	[2]
	Soybean		
630	35	19	[4]
	Iris		
150	4	3	[6]
	Digit		
500	7	10	[6]
	Voting		
435	16	2	[1]
	Babs		
186	7	4	[6]
	Soccer		
346	5	3	[6]

intervals, and `excluded` is empty. After each specialization step the value removed from the conjunction is added to its `excluded` set. Before a conjunction c is specialized, all values in its `usable` set that will lead to useless specializations are discarded (step (6)). These are values which will cause $X_P(c)$ to become empty, leave $X_N(c)$ unchanged, or cause the set $\{X_N(a_i) \mid a_i \in \texttt{excluded}\}$ to become a redundant set cover of N. The latter test ensures that GCA generates (almost) most general conjunctions. In addition, it prevents the exclusion of values which would not have changed the X_N of a conjunction had they not been removed from it.

GCA thus generates internally disjunctive concept descriptions that are (almost) most general and employs both a stop-growth and significance criterion for pruning.

3 Experimental method and test domains

Comparisons between beam search and non-beam search versions of GCA were performed in 10 test domains. These domains were used by other authors and some occur in many comparisons. We thus give in Table 3 only a short description together with references where the interested reader may obtain more detailed information for each domain. The large variation in accuracy from one domain to another (see next 3 sections) is indicative of inherent noise in some domains. Similar results are reported in the references listed in Table 3.

The following experimental method was employed in all cases: Ten random training and test sets were selected for each domain. In each case the training set comprised 70% of the training instances and the remaining 30% the test set. All the experiments in a particular domain used the same 10 training and test sets. The average results over the ten training and test sets are reported.

Table 4. Results obtained with different beam widths

BW	ST	TT	Time	#Tests	%Correct
			Lymphography		
1	99	0	7 σ 0.7	32 σ 4.7	83.1 σ 5.2
10			110 σ 13.0	33 σ 6.4	83.8 σ 4.5
20			308 σ 34.8	34 σ 6.1	84.4 σ 4.9
			Primary tumor		
1	90	90	23 σ 0.5	29 σ 2.3	44.0 σ 4.0
10			139 σ 11.7	65 σ 7.0	44.2 σ 4.4
20			182 σ 16.0	66 σ 6.4	44.1 σ 4.3
			Breast cancer		
1	99.9	0	6 σ 1.1	10 σ 3.1	72.1 σ 4.7
10			203 σ 25.0	29 σ 5.0	72.3 σ 4.1
20			533 σ 71.9	31 σ 4.9	72.6 σ 4.2
			Hepatitis		
1	99	99	7 σ 1.2	3 σ 0.9	80.6 σ 2.7
10			206 σ 35.9	9 σ 1.7	82.6 σ 4.7
20			352 σ 74.1	9 σ 2.2	80.4 σ 4.0
			Soybean		
1	99	0	85 σ 2.8	160 σ 12.5	87.2 σ 3.1
10			1074 σ 62.5	175 σ 13.4	87.6 σ 3.2
20			2757 σ 146.0	172 σ 12.2	88.4 σ 2.2
			Iris		
1	90	99.9	3 σ 0.0	4 σ 1.1	93.5 σ 2.9
10			26 σ 3.3	8 σ 1.6	92.4 σ 2.8
20			43 σ 2.8	8 σ 1.6	92.4 σ 2.8
			Digit		
1	99	90	10 σ 0.4	37 σ 3.7	72.6 σ 4.0
10			82 σ 5.1	74 σ 7.0	72.6 σ 4.1
20			124 σ 7.8	78 σ 8.1	72.1 σ 4.4
			Voting		
1	0	99.9	3 σ 0.5	3 σ 1.0	95.3 σ 1.3
10			33 σ 4.8	21 σ 2.6	94.7 σ 2.0
20			59 σ 9.4	22 σ 3.4	94.8 σ 2.1
			Babs		
1	99.9	90	2 σ 0.5	3 σ 1.9	63.3 σ 5.1
10			20 σ 7.5	6 σ 2.6	63.1 σ 5.2
20			24 σ 10.3	6 σ 2.8	63.1 σ 5.2
			Soccer		
1	99.9	99	10 σ 0.8	2 σ 0.8	51.3 σ 3.4
10			293 σ 43.1	14 σ 2.8	52.9 σ 2.7
20			538 σ 126.7	16 σ 4.3	53.3 σ 2.7

4 Trying to improve description quality with a beam search

The purpose of experiments described in this section is to determine whether an increase in the beam width results in simpler and/or more accurate concept descriptions. Various combinations of significance and stop-growth thresholds were tried with k set to 1. In this way the best results (highest accuracy, lowest complexity) were obtained in each domain for $k = 1$. Then only the beam width was increased to 10 and 20. The results are given in Table 4. The standard deviations over the ten experiments are given for all the results. Run times are given in seconds, BW denotes the beam width, ST the significance threshold (%) and TT the stop-growth threshold (%).

For $k = 10$ the induction time ranged from 6 (primary tumor) to 34 (breast cancer) times that obtained for $k = 1$. For $k = 20$ the learning time ranged from 8 to 89 times that obtained for $k = 1$ (in the same two domains). These large increases are not unexpected since the selection of the k best complexes and testing for duplicate expressions may cause the time requirements of a beam search to be more than k times that for $k = 1$.

Only in 2 cases did an increase in the beam width increase the classification accuracy by 2% or more, namely in the soccer ($k = 20, +2\%$) and hepatitis ($k = 10, +2\%$) domains. From the remaining cases it is clear that increasing the beam width rarely leads to any significant improvements in description accuracy.

The most upsetting result is the large increase in description complexity when increasing the beam width from 1 to 10. Apart from the lymphography and soybean domains this increase in beam width more than doubled description complexity. Analysis of the generated descriptions revealed two reasons for this increase in complexity: Firstly, more specific rules (containing more tests) tend to be selected during a beam search because these rules cover fewer negative instances and hence have a larger value for the Laplace function than rules found with $k = 1$. These more specific rules often cover fewer instances than those found when $k = 1$, thus more conjunctions are needed to cover "enough" positive instances of a concept. Secondly, the specific rules found during a beam search tend to be more significant than those found with $k = 1$. Hence more conjunctions pass the significance test during a beam search, thus increasing the overall complexity. The small increases in complexity when increasing the beam width from 10 to 20 indicates that a ceiling for description complexity is soon reached after the initial jump in complexity.

From these results it is clear that employing a beam search can usually be avoided due to the large increases in complexity and rare but small increases in accuracy.

The only mechanisms available to GCA (and CN2) to combat the increase in description complexity is to perform more severe pruning. In the next two sections we describe experiments where the significance and stop-growth thresholds are increased during beam search.

5 Increasing the significance threshold during beam search

Table 5 contains the results where the previous beam search experiments were repeated with higher significance thresholds. In each domain the thresholds were increased until the accuracy of descriptions decreased by 1% or more or until the highest threshold (99.9%) was reached. The simplest descriptions thus obtained are reported in Table 5. The results obtained with $k = 1$ are repeated for easy comparison. The results for the breast cancer, babs and soccer domains were omitted since the significance threshold was already at 99.9% in these domains. Results for the digit and voting domains were omitted as well since all increases in the significance threshold decreased accuracy by 1% or more. In the remaining 5 test domains a higher significance threshold reduced description complexity from 8% (soybean, $k = 20$) to 44% (iris, $k = 20$) without an unacceptable drop in accuracy. However, when compared with results obtained for $k = 1$, a beam search gave slightly lower complexity in only the soybean and lymphography domains. Furthermore, the higher significance thresholds left description accuracy mostly unchanged. Thus increasing the significance threshold during beam search do not lead to descriptions with either significantly higher accuracy or significantly lower complexity than those found with $k = 1$. Hence, when compared with $k = 1$, a beam search is not justified even when using more stringent significant thresholds. The results also point to a general rule of thumb, namely to perform more stringent pruning when increasing the beam width.

Table 5. Results for higher significance thresholds during beam search

BW	ST	TT	Time	#Tests	%Correct
Lymphography					
1	99	0	7 σ 0.7	32 σ 4.7	83.1 σ 5.2
10	99.9		98 σ 12.2	27 σ 4.7	83.3 σ 5.2
20	99.9		267 σ 26.8	27 σ 3.9	84.2 σ 5.4
Primary tumor					
1	90	90	23 σ 0.5	29 σ 2.3	44.0 σ 4.0
10	99.9		129 σ 8.1	46 σ 3.4	44.4 σ 5.1
20	99.9		170 σ 12.9	46 σ 3.3	44.3 σ 5.1
Hepatitis					
1	99	99	7 σ 1.2	3 σ 0.9	80.6 σ 2.7
10	99.9		192 σ 35.1	8 σ 1.3	81.3 σ 6.0
20	99.9		342 σ 75.2	8 σ 1.5	81.1 σ 4.3
Soybean					
1	99	0	85 σ 2.8	160 σ 12.5	87.2 σ 3.1
10	99.9		1015 σ 67.8	159 σ 11.9	87.3 σ 3.5
20	99.9		2640 σ 167.7	159 σ 12.9	88.3 σ 2.4
Iris					
1	90	99.9	3 σ 0.0	4 σ 1.1	93.5 σ 2.9
10	99.9		25 σ 3.2	5 σ 1.2	92.4 σ 2.8
20	99.9		42 σ 2.6	5 σ 1.2	92.4 σ 2.8

6 Increasing the termination threshold during beam search

The last group of experiments attempted to further decrease description complexity during beam search by increasing the stop-growth threshold. The significance thresholds which gave the best results were used (see previous two sections), and the stop-growth threshold was increased while this did not decrease description complexity by 1% or more. The results are given in Table 6. Again the results obtained with a beam width of 1 are repeated for easy comparison. No results are given for the iris and voting domains since the maximum termination threshold (99.9%) was already used in these domains. No results are also given for the lymphography, primary tumor and soccer domains since any increase in the stop-growth thresholds during beam search decreased accuracy (usually considerably) more than 1%. This "overpruning" is to be expected since increasing the stop-growth threshold is a more severe form of pruning than increasing the significance threshold. The reason is that a higher threshold requires each specialization step to make a big improvement to prevent the termination of all

Table 6. Results when increasing the stop-growth thresholds during beam search

BW	ST	TT	Time	#Tests	%Correct
			Breast cancer		
1	99.9	0	6 σ 1.1	10 σ 3.1	72.1 σ 4.7
10	99.9	90	250 σ 26.7	26 σ 4.2	73.3 σ 4.2
20	99.9	99	227 σ 51.4	13 σ 1.9	72.0 σ 6.5
			Hepatitis		
1	99	99	7 σ 1.2	3 σ 0.9	80.6 σ 2.7
10	99.9	99.9	141 σ 22.7	7 σ 0.9	82.1 σ 5.6
20	99.9	99.9	219 σ 44.3	7 σ 0.9	82.1 σ 5.6
			Soybean		
1	99	0	85 σ 2.8	160 σ 12.5	87.2 σ 3.1
10	99.9	99	491 σ 9.1	81 σ 2.9	86.6 σ 3.2
20	99.9	90	1223 σ 46.5	106 σ 4.1	87.4 σ 2.9
			Digit		
1	99	90	10 σ 0.4	37 σ 3.7	72.6 σ 4.0
10	99	99	52 σ 2.1	51 σ 2.6	72.6 σ 3.4
20	99	99	66 σ 3.6	53 σ 3.3	72.8 σ 3.9
			Babs		
1	99.9	90	2 σ 0.5	3 σ 1.9	63.3 σ 5.1
10	99.9	99.9	4 σ 0.3	0 σ 0.0	63.5 σ 4.7
20	99.9	99.9	4 σ 0.3	0 σ 0.0	63.5 σ 4.7

further specialization of a conjunction.

Increasing the stop-growth threshold halved description complexity in the breast cancer ($k = 20$) and soybean ($k = 10$) domains. In the digit domain slightly less spectacular improvements were recorded. In the babs domain all rules were pruned away leaving the default rule which gave an accuracy of 63.3%. When compared with results obtained for $k = 1$, it is clear that a beam search gave significantly simpler descriptions in only the soybean domain. Since description accuracy stayed more or less the same, this is the only domain where more stringent pruning during beam search gave significantly better results than those obtained for $k = 1$.

To summarize, only in the soybean domain did a beam search lead to significantly simpler descriptions than those found with $k = 1$. In contrast, a beam width of 1 gave considerably simpler descriptions in the primary tumor, digit and hepatitis domains. In the remaining domains description complexity was similar when comparing the best results obtained in each domain. Only in the soccer and hepatitis domains did a beam search increase accuracy by 2%. In the remaining instances all the descriptions had similar accuracy. By virtue of the large savings in computation time when keeping only a single best conjunction and the fact that this approach is rarely outperformed with respect to either accuracy or complexity, beam search seems to be of little use.

7 Summary and conclusion

This paper described experiments that investigated the utility of following a beam search when learning concept descriptions from examples. It was shown that employing a beam search in GCA rarely improves description accuracy or complexity, and often leads to large increases in description complexity. Similar beam search results for CN2 (which generates more restricted "pure" descriptions) and for AQ (which considers many fewer specializations than GCA) are not available. However, results reported in [7] for the primary tumor, lymphography and breast cancer domains suggest a similar tendency for these two algorithms.

References

1. D W Aha, D Kibler, and M K Albert. 'Instance-based learning algorithms'. *Machine Learning*, **6**:37–66, (1991).
2. P Clark and R Boswell. 'Rule induction with CN2: Some recent improvements'. In Y Kodratoff, ed., *Machine Learning – European Working Session on Learning EWSL–91*, pp. 151–163, Berlin, (1991). Springer-Verlag.
3. P Clark and T Niblett. 'The CN2 induction algorithm'. *Machine Learning*, **3**:261–283, (1989).
4. R S Michalski and R L Chilauski. 'Learning by being told and learning from examples: An experimental comparison of the two methods of knowledge acquisition in the context of developing and expert system for soybean disease diagnosis'. *International Journal of Policy Analysis and Information Systems*, **4**:125–161, (1980).
5. 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 *Proceedings of the American Association of Artificial Intelligence*, pp. 1041–1045, Los Altos, CA, (1986). Morgan Kaufmann.
6. J Mingers. 'An empirical comparison of pruning methods for decision tree induction'. *Machine Learning*, **4**:227–243, (1989).
7. H Theron and I Cloete. 'GCA: A generic covering algorithm for learning attribute-based concept descriptions'. Submitted for publication, 1991.

Acknowledgements: We thank John Mingers for the breast cancer, babs, soccer, digit and iris data sets. The remaining data sets were obtained from the UCI Repository of machine learning databases and domain theories. We thank Peter Clark and Ross Quinlan who informed us of the repository. Peter Clark also sent us copies of his CN2 articles for which we are grateful. Patrick Murphy at the UCI Repository was extremely helpful in providing us with the data and promptly replying to all our requests. The lymphography, primary tumor and breast cancer data originally came from Institute of Oncology, University Medical Center, Ljubljana, Yugoslavia. We thank M. Zwitter and M. Soklic for providing this valuable data to the machine learning community.

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 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 wide margins and $1\frac{1}{2}$ or double spacing.
- 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 numbered and titled. Figures should be submitted as original line drawings/printouts, and not photocopies.
- 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 [1, 2, 3]. References should take the form shown at the end of these notes.

Manuscripts accepted for publication should comply with the above guidelines (except for the spacing requirements), and may be provided in one of the following formats (listed in order of preference):

1. As (a) LaTeX file(s), either on a diskette, or via e-mail/ftp – a LaTeX style file is available from the production editor;
2. As an ASCII file accompanied by a hard-copy showing formatting intentions:
   - Tables and figures should be on separate sheets of paper, clearly numbered on the back and ready for cutting and pasting. Figure titles should appear in the text where the figures are to be placed.
   - 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.

   Further instructions on how to reduce page charges can be obtained from the production editor.
3. In camera-ready format – a detailed page specification is available from the production editor;
4. In a typed form, suitable for scanning.

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 LaTeX or 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 in categories 2 and 4 above 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 less than 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.

References

1. E Ashcroft and Z Manna. 'The translation of 'goto' programs to 'while' programs'. In *Proceedings of IFIP Congress 71*, pp. 250–255, Amsterdam, (1972). North-Holland.
2. C Bohm and G Jacopini. 'Flow diagrams, turing machines and languages with only two formation rules'. *Communications of the ACM*, **9**:366–371, (1966).
3. S Ginsburg. *Mathematical theory of context free languages.* McGraw Hill, New York, 1966.

South African Computer Journal

Number 8, November 1992
ISSN 1015-7999

Suid-Afrikaanse Rekenaar-tydskrif

Nommer 8, November 1992
ISSN 1015-7999

Contents

GUEST CONTRIBUTION

RESEARCH ARTICLES

COMMUNICATIONS AND REPORTS

Sigma Press