PROBABILISTIC REASONING AS ICMAUS (III): HYPOTHETICAL REASONING, GEOMETRIC ANALOGIES, DEFAULT VALUES, NONMONOTONIC REASONING, AND MODELLING 'EXPLAINING AWAY'

J Gerard Wolff

University of Wales, Bangor, UK
gerry@sees.bangor.ac.uk

Abstract: This article and its two accompanying articles introduce the idea that probabilistic reasoning (PrbRs) may be understood as information compression by multiple alignment, unification and search (ICMAUS).

This article presents further examples showing how kinds and aspects of PrbRs may be understood as ICMAUS. Examples include:

- Hypotheticql or 'what if' reasoning.
- 'Indirection' in information retrieval.
- Reasoning with default values.
- Nonmonotonic reasoning.
- Solving geometric analogy problems.
- ICMAUS as an alternative to Bayesian networks for modelling the phenomenon of 'explaining away'.

Key Words: Geometric analogy; default values; nonmonotonic reasoning; Bayesian networks; 'explaining away'.

Category: Computing Methodologies/Artificial Intelligence/Reasoning (uncertainty, 'fuzzy' and probabilistic). I.2.SD.1.2.3.

1 Introduction

This is the third of three articles introducing the idea that probabilistic reasoning (PrbRs) may be understood as information compression by multiple alignment, unification and search (ICMAUS). As for the second article, this article assumes that the first article [10] has been read and that the reader understands, at least in outline, the concept of multiple alignment as it has been developed in this research. At least in broad-brush terms, the reader should also understand how the information compression (IC) associated with an alignment may be calculated and how probabilistic inferences may be drawn from an alignment.

In a similar way, readers should have a general understanding of how probabilities of inferences may be calculated (Section 2 of [12]) and should have considered, at least briefly, the examples presented in [12].

As a preparation for the rest of this article, Section 1.1, below, reviews how concepts of 'variable', 'value', 'type' and 'type definition' may be modelled in the ICMAUS scheme. The remainder of the article considers a range of different kinds of

PrbRs and how they may be modelled in the ICMAUS framework. These include:

- Hypothetical or 'what if' reasoning (Section 2). This may be characterised as reasoning about possibilities where there is relatively little evidence to discriminate amongst possible conclusions.
- 'Indirection' in information retrieval (Section 2.1.1): retrieval of information via semantic associations or other kinds of association.
- How 'default' values may be modelled and how a default value may be overridden by some other value when the requisite evidence becomes available (Section 4).
- The closely-related subject of nonmonotonic reasoning (Section 5): how, in 'everyday' reasoning, probabilistic conclusions that we may have drawn when relatively little information was available may be overturned when more information comes to hand.
- How simple geometric analogy problems may be solved within the ICMAUS scheme (Section 6).
- The phenomenon of 'explaining away' and how the ICMAUS framework may provide an alternative to existing explanations in terms of Bayesian networks (Section 7).

As in [10] and [12], the majority of examples in this article show results produced by the SP61 model, which was described in Section 6 of [10].

This article ends with a brief review of the potential benefits and distinctive features of the approach to PrbRs which has been presented in these three articles.

1.1 'Variable', 'value, 'type' and 'type definition'

In Section 3.4 of [12], we saw how the definition of the class 'vertebrate' could contain an empty 'slot' like '`name #name`' which, in an alignment describing an individual animal could be filled with the name of the animal, thus: '`name Tibby #name`'. Similar slots appear elsewhere in the class hierarchy of animals in that section and there is a similar 'name' slot in the example in Section 4.1 of [12].

Any such slot is like a 'variable' in a conventional computing system: it is a place holder for information which may be empty or filled. The 'value' which fills one of these 'variables' may be a single symbol (as in the '`name #name`' example, above) or it may be a pattern of two or more symbols which may itself contain filled or empty 'variables'. Examples of the latter kind of value for a variable can be seen in the example of class hierarchy with inheritance presented in Section 3.4 of [12].

There is no process of 'assignment' in the ICMAUS scheme as there is in a procedural programming language (e.g., C++, Cobol etc), where the value of a variable may be overwritten and thus destroyed by another value. In ICMAUS, a variable acquires a value by alignment and unification of patterns, a process which is similar to the 'instantiation' of variables by 'unification' in a Prolog system (the concept of 'unification' in this context is a little more complicated than the concept of unification in ICMAUS).

Is it necessary to model 'assignment' of values to variables? Probably not. Although Prolog systems often contain an assignment mechanism, it seems that this is not strictly necessary and that instantiation is sufficient for most, possibly all, computational purposes.

As a working hypothesis, there is no need for an assignment mechanism in the ICMAUS scheme.

1.1.1 'Type' and 'type definition' of a variable

In a program written in a procedural language, every variable has a 'type' and every type has a 'type definition', either built into the language or defined by the programmer. The type of a variable, with its definition, specifies the range of possible values for the variable.

In Prolog, variables are not required to have an explicit type or type definition: the range of possible values for a variable is normally defined implicitly by the structure of the program. However, it is possible to imitate the effect of a type definition for a variable by the inclusion of suitable clauses in the program. For example, the variable '`X`' in the clause '`name(X)`' may be given a set of possible values by inclusion in the program of other clauses like '`name(john)`', '`name(susan)`', and so on.

The ICMAUS framework is similar: the range of possible values for a variable may be defined implicitly by the structure of Old but it is possible to imitate the effect of a type definition for a variable by including suitable patterns in Old. Figure 6 in Section 4.1 of [12] shows how a range of possible values for the variable '`name #name`' may be defined by the inclusion in Old of patterns like '`name Tweety #name`', '`name Tibby #name`', and so on.

Although concepts like 'variable', 'value', 'type' and 'type definition' have no formal status in the ICMAUS scheme, the terms will be used henceforth as 'shorthand' to refer to structures like those just described and, for convenience, quote marks will be dropped.

2 Hypothetical ('what if') reasoning

In Section 1.2 of [10], 'reasoning', including 'probabilistic reasoning', was characterised as a process of "going beyond the information given". All the examples of reasoning by ICMAUS that we have considered thus far have exhibited this feature in the form of symbols and sequences of symbols from Old which are not matched to anything in New. But in every case (except the one discussed in Section 4.4.4 of [12]), we have focused on the alignment or alignments which are best in terms of the compression of New that may be achieved by encoding it in terms of Old and we have ignored the other 'suboptimal' alignments which are normally formed alongside the best alignment. We have not considered the possible uses of reasoning in this domain.

What happens if there is little or no information in New or if the process of 'reasoning' goes so far beyond the information in New that alternative lines of reasoning lose their support? Something like this seems to happen in ordinary thinking when, in considering alternative scenarios in the future, we think and sometimes worry about possibilities which are very unlikely to occur (winning the lottery and what we might do with the 'loot', falling under the proverbial bus, etc). Hence the patronising advice, "Don't worry, it may never happen!" or, a little more helpfully, "Let's climb that mountain [or 'cross that bridge'] when we get to it" - the problem may never arise.

This kind of reasoning is sometimes called 'what if' reasoning or hypothetical reasoning.

Figure 1 shows the first seven alignments formed by SP61 with the patterns for fault diagnosis in car engines from Figure 9 in Section 4.3 of [12] in Old and only the symbol '`Start`' in New. If it is allowed to proceed without any check that the alignments formed are actually or potentially useful, the program would carry on creating alignments corresponding to the large number possible alignments which are implicit in the patterns in Old.

Figure 1 about here

Figure 1

Alignments formed by SP61 with the patterns from Figure 9 in Section 4.3 of [12] in Old and only the symbol '`Start`' in New. SP61 was set to stop searching when if two 'unsupported' inferential steps had been made. Without this check, many more alignments would be formed.

Out of all these many alignments, the first one (at the top of Figure 1) has the highest CD because it encodes New completely and contains the fewest patterns from Old.[1] Amongst the other alignments, CD values decrease as the number of patterns from Old in each alignment increases.

2.1 Other manifestations of 'suboptimal' reasoning

The kind of 'suboptimal' reasoning discussed in this section, where inferences are drawn which have relatively little support from the available evidence, seems to have uses other than the kind of 'hypothetical' ('what if') reasoning considered thus far:

- As is discussed briefly in Section 2.1.1, below, this kind of reasoning seems often to be required as a part of ordinary reasoning as a means of escaping from 'local peaks' in the search space.
- In the next section (Section 2.1.1), it is argued that, when information is retrieved indirectly via semantic associations or other kinds of association, inferences of a similar kind are required.
- In Section 4, it is argued that 'default values' for variables can be modelled in the ICMAUS scheme and that, in this context, they can be understood as suboptimal inferences which are similar to inferences formed in hypothetical reasoning.

---

1. As was noted in [10], the IC associated with any alignment is not realised until New has been encoded in terms of the patterns from Old which appear in the alignment. However, it is convenient at times to say that a given alignment encodes New or compresses New. Wherever this form of words is used it should be understood to mean that a code which expresses New in a compressed form may be derived from the alignment under consideration.

2.1.1 Escaping from 'local peaks' in the search space

As mentioned above, 'suboptimal' reasoning of the kind which has been considered in this section seems to have a role to play in normal reasoning where there are 'local peaks' in the search space. To escape from a local peak, the inference process must be able to explore regions of the search space which are suboptimal from a local perspective but which may provide the means of finding a result elsewhere which is better from a broader perspective (a higher 'peak').

The example described in Section 4.3.6 of [12] (Figure 12) illustrates this kind of reasoning. Compared with Figure 11 in [12], the example contains a gap in New because the substring '`yes no no no`' is missing. Working left-to-right in sequential mode, SP61 is able to bridge this gap in New because, after it has found a good alignment for the first part of New, it is able to continue forming relatively poor alignments until it finds one which bridges the gap so that the right-hand section of New can be included. The alignment shown in Figure 12 in Section 4.3.6 of [12] has a higher CD value than any of the earlier alignments, including the best of the alignments for the left-hand section of New.

3 Indirection in information retrieval

A common problem with the simpler kinds of system for information retrieval is that you will fail to find relevant material if, in the query which you apply to initiate the search, you use a term (or terms) which does not appear in the index, title or body of that relevant material. For example, you may be interested in 'epistemology', but this term will not retrieve material which is relevant to your interests if that material is indexed by or uses the term 'theory of knowledge' instead of 'epistemology'.

This kind of information retrieval may seem far removed from hypothetical or 'what if' reasoning but, as noted above, it seems that, at least in the ICMAUS scheme, similar principles are involved.

Figure 2 shows a small knowledge base in which the first four patterns represent bodies of information about 'sleep', 'theory of knowledge', 'luggage' and 'concurrency'. At the bottom is shown four patterns from a thesaurus where each pattern contains one of the four terms just mentioned, together with one or more synonyms for that term.

```
sleep ... info_about_sleep ... #sl                        500
theory of knowledge ... info_about_TOK ... #tk           800
luggage ... info_about_luggage ... #lg                    200
concurrency ... info_about_concurrency ... #cy           400
sleep #sl nap #np kip #kp                                   1
epistemology #ey theory of knowledge #tk                    1
luggage #lg baggage #bg                                     1
concurrency #cy parallel processing #pp                     1
```

Figure 2
A small knowledge base, including patterns representing entries in a thesaurus.

The best alignment formed with this knowledge base in Old and the pattern 'parallel processing' in New is shown at the top of Figure 3. Below it is the one other alignment in the reference set of alignments for the best alignment. The relative probabilities of the two alignments are 0.9878 and 0.0122 respectively.

Figure 3 about here

Figure 3

The best alignment and the one other alignment in its reference set formed by SP61 with the patterns from Figure 2 in Old and the pattern 'parallel processing' in New.

The best alignment shows that 'concurrency' is associated with 'parallel processing'. This is not entirely without interest or usefulness but it does not tell us anything about the technical or other aspects of parallel processing. However, the second alignment in Figure 3 contains a pattern with 'info_about_concurrency' which, we may suppose, is much closer to what we are looking for.

Although the relative probability of the second alignment is only 0.0122 (and likewise for the pattern 'concurrency ... info_about_concurrency ... #cy' and the symbol 'info_about_concurrency', this is the best of the 'suboptimal' inferences made by the system. Thus, information retrieval with 'indirection' seems to require inferences which are similar in some ways to the inferences formed in hypothetical or 'what if' reasoning.

4 'Default value' for a variable and over-riding a default value

In many knowledge-based systems, it is common practice to provide a mechanism by which any variable can be given a 'default value' - a value which is assigned to the variable if no value is supplied by the user. For example, in a system for booking guests into a hotel, there may be a variable ('field') in the record for each guest which describes their dietary needs. For a variable like this, the most frequent value is likely to be 'standard'. In that case, it saves effort by the booking clerk if that value is supplied as a default which, when the need arises for any particular guest, can be over-ridden by 'vegetarian', 'gluten free' or some other value.

Figure 4 shows part of a knowledge base for booking in hotel guests, including three patterns which identify some of the possible values for the variable 'diet'. As usual with patterns in the ICMAUS scheme, each pattern has an associated frequency of occurrence. The relative values of these frequencies are intended to be reasonably realistic although, like other examples in these articles, the figures are fictional. The main point here is that the type of diet which is identified as 'standard' is served very much more frequently than the 'vegetarian' or 'gluten free' diets.

guest name #name	
addr #addr	
tel #tel	
room #room	
diet #diet #guest	10000
diet standard #diet	9000
diet vegetarian #diet	500
diet gluten free #diet	80

Figure 4

Part of a knowledge base for booking in hotel guests, with some of the possible alternative values for dietary needs.

Figure 5 shows results from a run of SP61 with the patterns from Figure 4 in Old and the pattern '`guest name Mr Smith #name addr London #addr room 2 #room #guest`' in New. The alignments shown are, in descending order of their CDs, the best alignment formed and the other three alignments in its reference set. CDs and relative probabilities are shown in the caption to the figure.

In this pattern in New, the name, address and assigned room number of the guest are specified but there is no entry for his home telephone number or dietary requirements. The first of the four alignments in Figure 5 does not add anything to the information in New, except that there is an empty variable for telephone number and an empty variable for diet. For this reason, we can ignore this alignment, despite its high relative probability (shown in the caption to Figure 5). However, the remaining three alignments, with their relative probabilities (shown in the caption to Figure 5), tell us the three alternative kinds of diet specified in the knowledge base and their relative probabilities in this context. In effect, the first of these diets ('`standard`') is a default value for the 'diet' variable.

Figure 5 about here

Figure 5

Results from SP61 with the patterns from Figure 4 in Old and the pattern '`guest name Mr Smith #name addr London #addr room 2 #room #guest`' in New. The alignments shown are, in descending order of CD, the best alignment found and the other three alignments in its reference set. Reading from the top, the CD values for the four alignments are: 249.73, 239.30, 235.13 and 229.87. In the same order, the relative probabilities are: 0.9916, 0.0079, 0.0004 and 7.0122e-5.

4.1 Default values and reasoning as hypothetical or 'what if' reasoning

This account of how default values may be modelled in the ICMAUS scheme relies on the use of hypothetical or 'what if' reasoning as described in Section 2. If we always focus narrowly on the best alignment for any combination of New and Old then, in a case like the one just considered, we would learn only that there were empty variables for the guest's telephone number and dietary requirements. To see the possible values for 'diet' and their relative probabilities (in terms of the system's current knowledge), we need to look at alignments with lower relative probabilities than the best one.

4.2 Over-riding a default value

What happens if a particular guest requires a diet which is in the knowledge base for the system but is not '`standard`'? Figure 6 shows the best alignment formed by SP61 with the patterns from Figure 4 in Old and the pattern '`guest name Mr Smith #name addr London #addr room 2 #room diet gluten free #diet #guest`' in New.

Figure 6 about here

Figure 6

The best alignment formed by SP61 (CD = 520.85) with the patterns from Figure 4 in Old and the pattern '`guest name Mr Smith #name addr London #addr room 2 #room diet gluten free #diet #guest`' in New. There is no other alignment in the best alignment's reference set. Its relative probability is 1.0.

In this case, the default value ('`standard`') is irrelevant because it does not match the value for the 'diet' variable which is now specified in New. There is no other alignment apart from the one shown that encodes all and only the same symbols from New - the reference set of alignments for the best alignment contains only one member, the best alignment itself. Consequently, the relative probability of this alignment (as we defined it in Section 2.5 of [12]) is 1.0.

4.3 Introducing a value which is new to the system

What happens if the guest specifies a type of diet (e.g., '`macrobiotic`' or '`anti-candida`') which is not recorded in the booking system's knowledge base? Assuming that the chef is exceptionally versatile and that the hotel is willing to oblige, the results for one such specification are shown in Figure 7. These results - the best alignment found by SP61 and the other alignments in its reference set - are the same as in Figure 5, except that, in all four cases, the symbols '`diet anti-candida #diet`' appears in row 0 (representing New) where previously there was nothing.

Figure 7 about here

Figure 7

Results from SP61 with the patterns from Figure 4 in Old and the pattern '`guest name Mr Smith #name addr London #addr room 2 #room diet anti-candida #diet #guest`' in New. The alignments shown are, in descending order of CD, the best alignment found and the other three alignments in its reference set. The values for CD and relative probability are the same as in Figure 5.

In the first alignment shown in Figure 7, the symbol '`anti-candida`' in New occupies a vacant 'slot' in the alignment. In the second, third and fourth alignments, there is a 'clash' between the symbol '`anti-candida`' and, respectively, the symbols '`standard`', '`vegetarian`' and '`gluten free`'. In all four alignments shown in Figure 7, the symbol '`anti-candida`' is not matched with or encoded by anything in Old.

As it stands, SP61 is not designed to do anything with unmatched symbols from New except to show them as and when they appear in alignments formed by the program. However, as was noted in Section 1.3 of [10], SP61 is seen as a stepping stone on a path which leads to a model which demonstrates unsupervised learning and the integration of such learning with PrbRs, parsing, pattern recognition and other kinds of computation. When the model has been developed to include learning, it is anticipated that, for each pattern in New, the model will first try to encode it as fully as possible in terms of patterns already stored in Old. If, when this is done, there are any symbols from New which have not been encoded, they will be added to Old in such a way that their context can be retrieved.

When the model has been developed in this way, it is anticipated that the occurrence of a symbol like '`anti-candida`' in New in the context '`diet ... #diet`' should result in the addition to Old of a pattern like '`diet anti-candida #diet`' which, in effect, augments the range of dietary types which the booking system knows about. Assuming that the knowledge base has been augmented in this way, the system may generate a new alignment like the one shown in Figure 6 but with the symbol '`anti-candida`' in place of the symbols '`gluten free`'. And, like the alignment shown in Figure 6, the relative probability of this alignment should be 1.0.

In short, when the SP system has been developed to allow unsupervised learning, it should be able to accept new knowledge as and when it arises and incorporate that knowledge into its knowledge base. As usual, the principle of "rubbish in means rubbish out" still applies: like a human learner, the system will assimilate false information unless there is some kind of 'filter' to separate 'good' information from 'bad'.

5 Nonmonotonic reasoning

The concepts of *monotonic* and *nonmonotonic* reasoning are well explained by Ginsberg [4] (words are emphasised as in the original):

> *... Conventional deductive inference has a property known as **monotonicity**: as your set of beliefs grows, so does the set of conclusions that can be drawn from those beliefs.*
>
> *Consider the* [syllogistic deduction] *that Socrates was mortal from the facts that he was a man and that all men are mortal. If we add some new facts to our premises, the conclusion that Socrates was mortal will remain valid; short of **removing** one of the premises that underlies it, there is nothing we can do to invalidate the argument. Even if we were to add the fact that Socrates could leap the Parthenon in a single bound, we would still conclude that he was mortal: physical prowess notwithstanding, he was still a man.*
>
> *Our conclusion about Socrates' mortality remains valid because the rule, 'All men are mortal', is universally quantified. There are no exceptions, not even implicit ones. **All** men are mortal, including those capable of jumping over the Parthenon or those possessing other extraordinary qualities.*
>
> *This is to be contrasted with, for example, a statement such as, 'Birds fly'. It is not the case that **all** birds fly, but only that **typical** birds fly. Now if I tell you that Tweety is a bird, it is reasonable for you to conclude that Tweety can fly, since you have no evidence that Tweety is atypical. But there is a crucial difference here: you based your conclusion that Tweety could fly on an **absence** of information about Tweety's atypicality: on learning something new about Tweety (such as that he has a foot set in concrete), you will need to flexibly revise your conclusion that he can fly.*
>
> *The inference here is **non**monotonic: On learning a new fact (that Tweety has a foot set in concrete), you were forced to retract your conclusion that he could fly. This original conclusion did not follow logically from the facts that birds typically fly and that Tweety is a bird; it follows from these facts together with the **assumption** that Tweety is a typical bird. When we learn more about Tweety, we may discover that this assumption is unjustified.*

This section presents some simple examples which suggest that the ICMAUS framework may provide a 'home' for nonmonotonic reasoning. However, no attempt is made to address the several problems associated with nonmonotonic reasoning which are described in [4].

5.1 Typically, birds fly

Figure 8 shows a set of patterns like the patterns describing animals that we saw in Figure 4 of [12] but adapted to illustrate nonmonotonic reasoning. The main points to notice in this connection are that the set of patterns includes one for the class '`bird`' and one each for '`swallow`', '`eagle`' and '`penguin`'. Also, the set of patterns in Figure 8 contains the pattern '`bird canfly yes #canfly #bird`'.

This last-mentioned pattern provides a 'default value' for the variable '`canfly #canfly`' in the class '`bird`' in much the same way as '`diet standard #diet`' provided a default value for the variable 'diet #diet' in the hotel booking example in Section 4. In the present case, there are no alternative values (with lower frequencies) provided in the knowledge base, and the context 'bird ... #bird' is included in the pattern. This context ensures that the default value applies only to birds and allows for the possibility that a different default value for the same variable might apply in the case of, say, insects.

The pattern 'bird canfly yes #canfly #bird' may be interpreted as a statement that "Typically, birds fly". It should not be interpreted as the universally-quantified statement that "All birds fly" because, as will be seen below, it can be over-ridden in much the same way as 'diet standard #diet' was in our hotel booking example (Section 4). How one might express a universal truth is considered briefly later.

5.2 Tweety is a bird so, probably, Tweety can fly

Figure 9 shows, at the top, the best alignment found by SP61 with the pattern 'bird Tweety' in New and the patterns from Figure 8 in Old. Below this alignment, in descending order of CDs, are the other alignments in the reference set. Relative probabilities of these alignments are shown in the caption to the figure.

The first alignment, which has by far the highest relative probability, tells us that Tweety almost certainly has wings, feathers, beak and a crop. In itself, this alignment does not tell us whether or not Tweety can fly. This accords with a 'strict' interpretation of the statement "Tweety is a bird": because of the existence of flightless birds as well as birds that fly, this statement, in itself, does not tell us whether or not Tweety can fly.

However, the second alignment tells us, without supposing that Tweety is any particular type of bird, that it is more likely than anything else that Tweety can fly. The default value for the variable 'canfly #canfly' manifests itself in much the same way as the default value in the hotel booking example.

```
vertebrate description
     name #name
     structure #structure
     function #function
     #description #vertebrate                                  23000
vertebrate reptile description
     structure dry_skin cold_blood #structure
     function
          lays_eggs_in_shells_on_land
     #function
     #description #reptile #vertebrate                          5000
vertebrate bird description
     structure
          wings #wings
          feathers beak crop
     #structure
     function canfly #canfly lays_eggs
     #function
     #description #bird #vertebrate                            10000
vertebrate mammal description
     structure hairy warm_blooded #structure
     function
          young_born_alive feeds_young_on_milk
     #function
     #description #mammal #vertebrate                           8000
bird swallow description
     wings pointed #wings
     function canfly yes #canfly #function
     #description #swallow #bird                                1200
bird eagle description
     wings large #wings
     function canfly yes #canfly #function
     #description #eagle #bird                                   300
bird penguin description
     wings stubby #wings
     function canfly no #canfly #function
     #description #penguin #bird                                 400
bird canfly yes #canfly #bird                                   9000
name Tweety #name                                                300
name John #name                                                  500
name Tibby #name                                                 400
```

Figure 8

A set of patterns to illustrate nonmonotonic reasoning.

The last three alignments in Figure 9 tell us that, in order of probability and within the limits imposed by the system's store of knowledge, Tweety could be a swallow, a penguin, or an eagle. The alignments predict that, in the first and last cases, Tweety would be able to fly, but not if he/she were a penguin.

Figure 9 about here

Figure 9

At the top, the best alignment found by SP61 for the pattern '`bird Tweety`' in New with the patterns from Figure 8 in Old. Below that alignment, in descending order of CD values, are the other alignments in the reference set of alignments. ***Key:*** `bd = bird, bk = beak, cp = crop, cy = canfly, dn = description, ee = eagle, fn = function, fs = feathers, ne = name, ls = lays_eggs, no = no, pn = penguin, se = structure, sw = swallow, Twy = Tweety, ve = vertebrate, ws = wings, ys = yes`. In order from the top, the relative probabilities of these alignments are: 0.9989, 0.0009, 5.7132e-5, 1.9044e-5, and 1.1711e-6.

5.3 A redundant pattern in old?

To a large extent, the information in '`bird canfly yes #canfly #bird`' duplicates the information in Old about specific types of bird: knowing that swallows and eagles fly and that, in comparison with the total number of swallows and eagles, a relatively small number of penguins do not fly, tells us that a specific bird of undefined type will probably be able to fly. This redundancy is reflected in the alignments shown in Figure 9.

The pattern '`bird canfly yes #canfly #bird`' is not totally redundant because its (notional) frequency is higher than the sum of the frequencies for the patterns for 'swallow' and 'eagle'. This implies that there are types of bird that can fly which are not swallows or eagles and which are not described in the patterns in Figure 8.

5.4 Tweety is a penguin, so tweety cannot fly

What happens if, when we have learned that Tweety is a bird and inferred that he/she can probably fly, we are then told that Tweety is a penguin? How, in the ICMAUS scheme, can we reason nonmonotonically, replacing our first inference with the new inference that Tweety cannot fly?

We must suppose that the information that Tweety is a bird ('`bird Tweety`') has been added to the patterns in Old which are shown in Figure 8. Now, in New, we present the pattern: '`penguin Tweety`' and search for good alignments. Figure 10 shows the reference set of two alignments found by SP61 with '`penguin Tweety`' in New and with Old augmented as just described.

With regard to Tweety's ability to fly, both these alignments tell us the same thing: since Tweety is a penguin, he/she cannot fly. Since there is no other alternative, the relative probability of this inference (in terms of the information which has been provided in New and Old) is 1.0. In the manner of nonmonotonic reasoning, the system has replaced one inference with another.

Figure 10 about here

Figure 10

The best alignment found by SP61 with the pattern ‘penguin Tweety’ in New and with the patterns from Figure 8 in Old, augmented with the pattern ‘bird Tweety’. In descending order, the relative probabilities of the two alignments are 0.9986 and 0.0014.

5.5 All birds fly but tweety is a bird that cannot fly

The example just described suggests that probabilistic statements like “Typically, birds fly” can be expressed and manipulated in an appropriate manner in the ICMAUS scheme. Can the same be said about non-probabilistic, universal statements like “All birds fly”?

It seems that the required strengthening of the statement can be achieved by defining the class ‘bird’ with a pattern something like this: ‘vertebrate bird description structure wings #wings feathers beak crop #structure function canfly yes #canfly lays_eggs #function #description #bird #vertebrate’ instead of the corresponding pattern in Figure 8.

The key difference between this pattern and the corresponding pattern in Figure 8 is that, in the previous case, the variable ‘canfly #canfly’ is empty, whereas, in this case, the variable has been given the value ‘yes’. The empty variable allows for alternatives, while the filled variable does not. Thus, it seems reasonable to interpret the pattern ‘vertebrate bird ... canfly yes #canfly ... #bird #vertebrate’ as a definition of the class bird which includes the assertion that *all* birds can fly.

In logic, the assertion that “All birds fly” is in direct conflict with a statement like “Tweety is a bird that cannot fly”. How might this conflict manifest itself in the ICMAUS scheme? The first statement is, in effect, contained in the revised definition of the class ‘bird’ described in the previous paragraph, while the second statement is, in effect, contained in a pattern ‘bird Tweety canfly no #canfly’.

SP61 has been run with the pattern ‘bird Tweety canfly no #canfly’ in New and the patterns from Figure 8 in Old but with the definition of the class ‘bird’ modified as described above so that, in effect, it contains the assertion that “All birds fly”. Also, the pattern defining the class ‘penguin’ is removed altogether (because it conflicts with the revised definition of the class ‘bird’) and the pattern defining ‘swallow’ is modified so that the substring ‘function canfly yes #canfly #function’ is removed, and likewise for ‘eagle’. It is not necessary to say that a swallow or eagle can fly because this is now recorded in the pattern for ‘bird’.

The best alignment found by SP61 and the other two alignments in its reference set are shown in Figure 11. In all three alignments, the value ‘no’ for the ‘canfly #canfly’ variable in New does not form a hit with the value ‘yes’ for the same variable in the patterns for ‘bird’. This mis-match expresses the contradiction between “All birds fly” and “Tweety is a bird that cannot fly”. We might speculate that there was an error either in the assertion that “Tweety is a bird that cannot fly” or in the supplied definition of ‘bird’.

Figure 11 about here

Figure 11

In descending order of relative probability (and CD value), the three alignments in the reference set of best alignments found by SP61 with ‘`bird Tweety canfly no #canfly`’ in New and the patterns from Figure 8 in Old, modified as described in the text. The relative probabilities are: 0.9999, 3.6591e-5 and 9.1477e-6. ***Key:*** as in Figure 9 plus: `pd = pointed, le = large`

If the truth of the first assertion is accepted then, when SP61 has been developed so that it can achieve unsupervised ‘learning’ (Section 1.3 of [10]), it may change Old so that, in the pattern defining the class ‘bird’, the value ‘`yes`’ is removed from the variable ‘`canfly #canfly`’, the substring ‘`canfly yes #canfly`’ is added to the patterns for ‘swallow’ and ‘eagle’, and a pattern is added to Old representing the newly-learned fact that a bird has been observed that cannot fly (something like ‘`bird name Tweety #name canfly no #canfly #bird`’). There may be a case for putting ‘swallow’ and ‘eagle’ into a newly-established subclass of birds all of which can fly.

6 Geometric analogies

Figure 12 shows an example of a well-known type of simple puzzle - a geometric analogy problem. The task is to complete the relationship “A is to B as C is to ?” using one of the figures ‘D’, ‘E’, ‘F’ or ‘G’ in the position marked with ‘?’. For this example, the ‘correct’ answer is clearly ‘E’.

What has this got to do with PrbRs? This kind of problem may be seen as an example of reasoning because it requires a process of “going beyond the information given”. Is it probabilistic? In the example shown in Figure 12, there seems to be only one ‘right’ answer which does not seem to leave much room for probabilities less than 1.0. But, more generally, in some problems of this type, a case can be made for two or even more altermative answers with a corresponding uncertainty about which answer is ‘correct’.

Computer-based methods for solving this kind of problem have existed for some time (e.g., Evans’s well-known heuristic algorithm [1]). In recent work [3, 2], MLE principles have been applied to good effect. The proposal here is that, within the general framework of MLE, this kind of problem may be understood in terms of ICMAUS.

Figure 12 about here

Figure 12

A geometric analogy problem.

In common with some previous work [3, 2], the proposed solution assumes that some mechanism is available which can translate the geometric forms in each problem into patterns of alpha-numeric symbols like the patterns in other examples in this article. For example, item 'A' in Figure 12 may be described as '`small circle inside large triangle`'.

How this kind of translation may be done is not part of the present proposals (one such translation mechanism is described in [1]). As has been pointed out previously [2], successful solution of this kind of problem depends on consistency in the way the translation is done. For this example, it would be unhelpful if item 'A' in Figure 12 were described as '`large triangle outside small circle`' while item 'C' were described as '`small square inside large ellipse`'. For any one puzzle, the description needs to stick to one or other of "X outside Y" or "Y inside X" - and likewise for 'above/below' and 'left-of/right-of'.

Given that the diagrammatic form of the problem has been translated into patterns as just described, this kind of problem can be cast as a problem of partial matching, well within the scope of SP61. To do this, symbolic representations of item A and item B in Figure 12 are treated as a single pattern, thus:

```
A small circle inside large triangle ;
     B large circle above small triangle #,
```

and this pattern is placed in New. Four other patterns are constructed by pairing a symbolic representation of item C (on the left) with symbolic representations of each of D, E, F and G (on the right), thus:

```
C1 small square inside large ellipse ;
     D small square inside large circle #
C2 small square inside large ellipse ;
     E large square above small ellipse #
C3 small square inside large ellipse ;
     F small circle left-of large square #
C4 small square inside large ellipse ;
     G small ellipse above large triangle #
```

These four patterns are placed in Old.

Figure 13 shows the best alignment found by SP61 with the patterns in New and Old as just described. The alignment is a partial match between the pattern in New and the second of the four patterns in Old. This corresponds with the 'correct' result (item E) as noted above.

Figure 13 about here

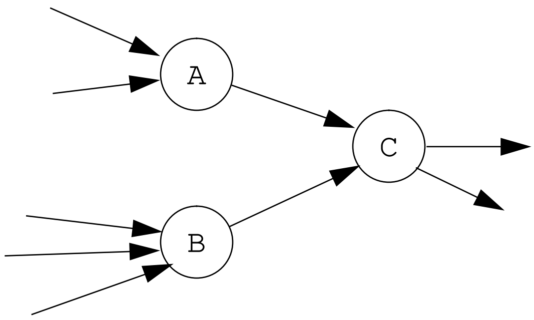
Figure 13

Multiple alignment in a geometric analogy problem.

7 Explaining away ‘explaining away’: ICMAUS as an alternative to Bayesian networks

In recent years, *Bayesian networks* (otherwise known as *causal nets*, *influence diagrams*, *probabilistic networks* and other names) have become popular as a means of representing probabilistic knowledge and for probabilistic reasoning.

A Bayesian network is a directed, acyclic graph comprising nodes like these:



where each node has one or more ‘inputs’ (connections with nodes that can influence the given node) and one or more ‘outputs’ (connections to other nodes that the given node can influence).

Each node represents a variable with a set of possible values (‘outputs’) and each node contains a set of conditional probabilities relating the probability of each of these values to values of the variables in the ‘input’ nodes. With this information, conditional probabilities of alternative ‘outputs’ for any node may be computed for any given combination of ‘inputs’. Using this kind of calculation in sequences of nodes, probabilities may be propagated through the network from ‘start’ nodes to ‘finishing’ nodes.

No attempt will be made in this article to discuss in detail how Bayesian networks may be modelled in the ICMAUS framework or to compare the two approaches to probabilistic inference. However, an example is presented below showing how ICMAUS may provide an alternative to a Bayesian network explanation of the phenomenon of “explaining away”.

7.1 A bayesian network explanation of “explaining away”

In the words of Judea Pearl [5, p. 7], the phenomenon of ‘explaining away’ may be characterised as: “If A implies B, C implies B, and B is true, then finding that C is true makes A *less* credible. In other words, finding a second explanation for an item of data makes the first explanation less credible.” (his italics). Here is an example described by Pearl [5, pp. 8-9]:

> *Normally an alarm sound alerts us to the possibility of a burglary. If somebody calls you at the office and tells you that your alarm went off, you will surely rush home in a hurry, even though there could be other causes for the alarm sound. If you hear a radio announcement that there was an earthquake nearby, and if the last false alarm you recall was triggered by an earthquake, then your certainty of a burglary will diminish.*

Although it is not normally presented as an example of nonmonotonic reasoning, this kind of effect in the way we react to new information is similar to the example we considered in Section 5 because new information has an impact on inferences that we formed on the basis of information that was available earlier.

The causal relationships in the example just given may be captured in a Bayesian network like the one shown in Figure 14.:

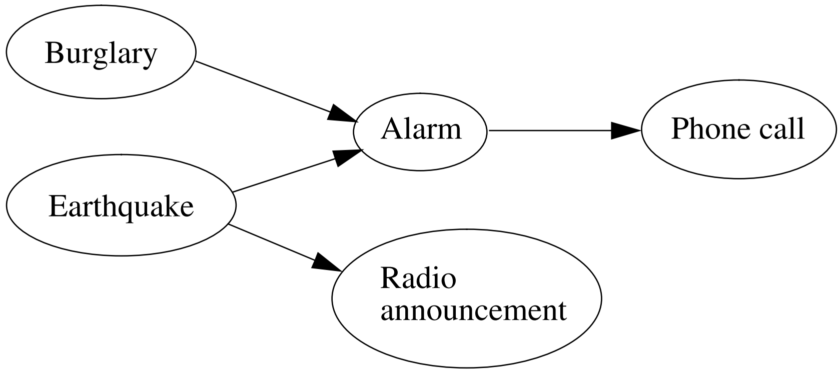


Figure 14

A Bayesian network representing causal relationships discussed in the text. In this diagram, "Phone call" means "a phone call about the alarm going off" and "Radio announcement" means "a radio announcement about an earthquake".

In [5], Pearl argues that, with appropriate values for conditional probability, the phenomenon of "explaining away" can be reproduced by this network, comparing the case where there is no radio announcement with the.case where there is one.

7.2 Representing contingencies with patterns and frequencies

To see how this phenomenon may be understood in terms of ICMAUS, consider, first, the set of patterns shown in Figure 15, which are to be stored in Old. The first four patterns in the figure show events which occur together in some notional sample of the 'World' together with their frequencies of occurrence in the sample.

Like other knowledge-based systems, an ICMAUS system would normally be used with a 'closed-world' assumption that, for some particular domain, the knowledge stored in the knowledge base is comprehensive. Thus, for example, a travel booking clerk using a database of all flights between cities will assume that, if no flight is shown between, say, Edinburgh and Paris, then no such flight exists. Of course, the domain may be only 'flights provided by one particular airline', in which case databases for other airlines would need to be checked. In systems like Prolog, the closed-world assumption is the basis of 'negation as failure': if a proposition cannot be proved with the clauses provided in a Prolog program then, in terms of that store of knowledge, the proposition is assumed to be false.

In the present case, we shall assume that the closed-world assumption applies so that the absence of any pattern msy be taken to mean that the corresponding pattern of events did not occur, at least not with a frequency greater than one would expect by chance.

The fourth pattern shows that there were 1000 occasions when there was a burglary and the alarm went off and the second pattern shows just 20 occasions when there was an earthquake and the alarm went off (presumably triggered by the earthquake). Thus we have assumed that burglaries are much more common than earthquakes. Since there is no pattern showing the simultaneous occurrence of an earthquake, burglary and alarm, we shall infer from the closed-world assumption that this constellation of events was absent or rare.

The first pattern shows that, out of the 1020 cases when the alarm went off, there were 980 cases a telephone call about the alarm was made. Since there is no pattern showing telephone calls (about the alarm) in any other context, the closed-world assumption allows us to assume that there were no false positives (including hoaxes) - telephone calls about the alarm when no alarm had sounded.

alarm phone_alarm_call	980
earthquake alarm	20
earthquake radio_earthquake_announcement	40
burglary alarm	1000
e1 earthquake e2	40

Figure 15

A set of patterns to be stored in Old in an example of 'explaining away'. The symbol '`phone_alarm_call`' is intended to represent a phone call conveying news that the alarm sounded; '`radio_earthquake_announcement`' represents an announcement on the radio that there has been an earthquake. The symbols '`e1`' and '`e2`' represent context for '`earthquake`' other than '`alarm`' or '`radio_earthquake_announcement`'.

Some of the frequencies shown in Figure 15 are intended to reflect the two probabilities suggested for this example in [5, p. 49]: "... the [alarm] is sensitive to earthquakes and can be accidentally (P = 0.20) triggered by one. ... if an earthquake had occurred, it surely (P = 0.40) would be on the [radio] news."

In our example, the frequency of '`earthquake alarm`' is 20, the frequency of '`earthquake radio_earthquake_announcement`' is 40 and the frequency of '`earthquake`' in other contexts is 40. Since there is no pattern like '`earthquake alarm radio_earthquake_announcement`' or '`earthquake radio_earthquake_announcement alarm`' representing cases where an earthquake triggers the alarm and also leads to a radio announcement, we may assume that cases of that kind have not occurred. As before, this assumption is based on the closed-world assumption that the set of patterns is a reasonably comprehensive representation of non-random associations in this small world.

The pattern at the bottom, with its frequency, shows that an earthquake has occurred on 40 occasions in contexts where the alarm did not ring and there was no radio announcement.

7.3 Approximating the temporal order of events

In these patterns and in the alignments shown below, the left-to-right order of symbols may be regarded as an approximation to the order of events in time. Thus, in the first pattern, '`phone_alarm_call`' (a phone call to say the alarm has gone off) follows '`alarm`' (the alarm itself); in the second pattern, '`alarm`' follows '`earthquake`' (the earthquake which, we may guess, triggered the alarm); and so on. A single dimension can only approximate the order of events in time because it cannot represent events which overlap in time or which occur simultaneously. However, this kind of approximation has little or no bearing on the points to be illustrated here.

7.4 Other considerations

Other points relating to the patterns shown in Figure 15 include:

- No attempt has been made to represent the idea that "the last false alarm you recall was triggered by an earthquake" [5, p. 9]. At some stage in the development of the SP system, an attempt may be made to take account of recency.
- In these imaginary figures, it has been assumed that burglaries (with a total frequency of occurrence of 1160) are much more common than earthquakes (with a total frequency of 100). As we shall see, this difference reinforces the belief that there has been a burglary when it is known that the alarm has gone off (but without additional knowledge of an earthquake).
- In accordance with Pearl's example [5, p. 49] (but contrary to the phenomenon of looting during earthquakes), it has been assumed that earthquakes and burglaries are independent. If there was some association between them, then, in accordance of the closed-world assumption, there would be a pattern in Figure 15 containing a symbol for each of those two kinds of event.

7.5 Formation of alignments: the burglar alarm has sounded

Receiving a phone call to say that one's house alarm has gone off may be represented by placing the symbol '`phone_alarm_call`' in New. Figure 16 shows, at the top, the best alignment formed by SP61 in this case with the patterns from Figure 15 in Old. The other two alignments in the reference set are shown below the best alignment, in order of CD value and relative probability. The actual values for CD and relative probability are given in the caption to Figure 15.

Figure 16 about here

Figure 16

The best alignment (at the top) and the other three alignments in its reference set formed by SP61 with the pattern '`phone_alarm_call`' in New and the patterns from Figure 15 in Old. In order from the top, the values for CD and relative probability are: 19.91, 0.6563; 18.91, 0.3281; 14.52, 0.0156.

The unmatched symbols in these alignments represent inferences made by the system. The probabilities for these symbols which are calculated by SP61 (using the method described in Section 2 of [12]) are shown in Table 1. These probabilities do not add up to 1 and we should not expect them to because any given alignment can contain two or more of these symbols.

Symbol	*Probability*
`alarm`	1.0
`burglary`	0.3281
`earthquake`	0.0156

Table 1

The probabilities, calculated by SP61, of unmatched symbols for the four alignments shown in Figure 16. The probability of '`phone_alarm_call`' is 1.0 because it is supplied as a 'fact' in New.

The most probable inference is the rather trivial inference that the alarm has indeed sounded. This reflects the fact that there is no pattern in Figure 15 representing false positives for telephone calls about the alarm. Apart from the inference that the alarm has sounded, the most probable inference (p = 0.3281) is that there has been a burglary. However, there is a distinct possibility that there has been an earthquake. But the probability in this case (p = 0.0156) is much lower than the probability of a burglary.

These inferences and their relative probabilities seem to accord quite well with what one would naturally think following a telephone call to say that the burglar alarm at one's house has gone off (given that one was living in a part of the world where earthquakes were not vanishingly rare).

7.6 Formation of alignments: the burglar alarm has sounded and there is a radio announcement of an earthquake

In this example, the phenomenon of 'explaining away' occurs when you learn not only that the burglar alarm has sounded but that there has been an announcement on the radio that there has been an earthquake. In terms of the ICMAUS model, the two events (the phone call about the alarm and the announcement about the earthquake) can be represented in New by a pattern like this:

'`phone_alarm_call radio_earthquake_announcement`'

or '`radio_earthquake_announcement phone_alarm_call`'. The order of the two symbols does not matter because it makes no difference to the result, except for the order in which columns appear in the best alignment.

Figure 17 about here

Figure 17

At the top, the best alignment formed by SP61 with the pattern '`phone_alarm_call radio_earthquake_announcement`' in New and the patterns from Figure 15 in Old. Other alignments formed by SP61 are shown below. From the top, the CD values are: 74.64, 54.72, 19.92, 18.92, and 14.52.

In this case, there is only one alignment (shown at the top of Figure 17) that can 'explain' all the information in New. Since there is only this one alignment in the reference set for the best alignment, the associated probabilities of the inferences that can be read from the alignment ('`alarm`' and '`earthquake`') are 1.0.

These results show broadly how 'explaining away' may be explained in terms of ICMAUS. The main point is that the alignment or alignments that provide the best 'explanation' of a telephone call to say that one's burglar alarm has sounded is different from the alignment or alignments that best explain the same telephone call coupled with an announcement on the radio that there has been an earthquake. In the latter case, the best explanation is that the earthquake triggered the alarm. For this reason, other possible explanations are pushed out of the picture.

The foregoing account of 'explaining away' in terms of ICMAUS is not entirely satisfactory because it does not say enough about alternative explanations of what has been observed. The next subsection tries to plug this gap.

7.7 Other possible alignments

What is missing from the foregoing account of 'explaining away' is any consideration of such other possibilities as, for example, a burglary (which triggered the alarm) and an earthquake (which led to a radio announcement) at the same time, or an earthquake that triggered the alarm and led to a radio announcement and, at the same time, a burglary that did not trigger the alarm. And so on. Alternatives of this kind may be created by combining alignments shown in Figure 17 with other alignments of that kind or with patterns or symbols from Old, or both these things. The two examples just mentioned are shown in Figure 18.

Any alignment created as just described may be evaluated in exactly the same way as any other alignment. CDs and absolute probabilities for the two example alignments are shown in the caption to Figure 18.

Figure 18 about here

Figure 18

Two alignments discussed in the text. (a) An alignment created by combining the second and fourth alignment from Figure 17. CD = 73.64, Absolute P = 5.5391e-5. (b) An alignment created from the first alignment in Figure 17 and the symbol '`burglary`'. CD = 72.57, Absolute P = 2.6384e-5.

Given the existence of alignments like those shown in Figure 18, values for relative probabilities of alignments will change. The best alignment from Figure 17 and the two alignments from Figure 18 constitute a reference set because they all 'encode' the same symbols from New. However, there are probably several other alignments that one could construct that would belong in the same reference set and the values of relative probabilities would change for each additional alignment. This is a shortcoming of relative probability as a comparative measure. Values for absolute probabilities are better in this regard.

Alignment	*Absolute probability*	*Relative probability*
First in Figure 17	1.1052e-4	0.5775
First in Figure 18	5.5391e-5	0.2881
Second in Figure 18	2.6384e-5	0.1372

Table 2

Values for absolute and relative probability for the best alignment in Figure 17 and the two alignments in Figure 18.

Given a reference set containing the first alignment in Figure 17 and the two alignments in Figure 18, values for relative probabilities are shown in Table 2, together with the absolute probabilities from which they were derived. Whichever measure is used, the alignment which was originally judged to represent the best interpretation of the available facts has not been dislodged from this position.

8 Evaluation

These three articles have concentrated on describing the ICMAUS framework and its potential range of applications in PrbRs. No serious attempt has yet been made to evaluate these proposals in relation to alternative systems. This section outlines some tentative ideas concerning distinctive features and potential benefits of these proposals.

8.1 Potential scope of ICMAUS proposals

As was described in Section 1.5 of [10], the primary goal of this research is to develop a framework which may integrate a wide range of observations and concepts in computing and cognition, not PrbRs alone.

If the scope of these proposals turns out to be as broad as is suggested by work to date (summarised in Section 1.3 of [10]), it will be much broader than any alternative. The potential advantages of this kind of broad scope were noted in Section 1.3.1 of [10].

8.2 Representation of knowledge using ‘patterns’

A key idea in the ICMAUS scheme is that diverse kinds of knowledge may be represented as patterns, as defined in Appendix A1 of [10], in one, two or more dimensions. This is a working hypothesis which may, of course, turn out to be wrong. The following subsections consider some of the reasons why this idea has been adopted and some of the questions that will, at some stage, need to be answered.

8.2.1 The working hypothesis that all kinds of empirical knowledge is derived ultimately from ‘raw’ data by unsupervised inductive learning

As was noted in Section 1.3 of [10], this programme of research is founded on an earlier programme of research on language learning [13, 14] and related ideas about unsupervised inductive learning [9, 10, 6]. Although the SP model has not yet been developed specifically for unsupervised inductive learning, this goal has been clealy in view throughout this work and it is anticipated that, with relatively small changes, the current version of the model can be adapted to accommodate this kind of learning.

As a working hypothesis, it is assumed that all kinds of (empirical) knowledge are derived ultimately from 'raw' data via inductive learning.

That being the case, it has seemed necessary to adopt a format for knowledge which has potential to accommodate a wide range of kinds of knowledge from 'raw' data of various kinds through syntax rules (for natural language) and various kinds of 'semantic' knowledge. A simple, uniform format for all kinds of knowledge would facilitate the development of simple, uniform processes for learning knowledge at all levels of complexity and abstraction.

8.2.2 Modularity of knowledge

One of the apparent benefits of the present proposals (compared with, say, Bayesian networks), is the modularity of knowledge stored as 'patterns' (as defined in Appendix A1 of [10]). This kind of modularity is similar to the modularity of 'rules' in expert systems and the advantages are much the same. Edward Shortliffe wrote:

> *"... an inference model that depends on a complex decision tree is apt to be difficult to augment without a complete diagram of the tree so that all the implications of additions can be observed. A modular system, on the other hand, permits knowledge to be acquired as isolated facts and allows the ... program itself to decide under what conditions the new information is relevant."* [8, p. 70].

The argument is clearly relevant to the supposition (Section 8.2.1) that all empirical knowledge has its origins ultimately in unsupervised inductive learning. If knowledge is modular in the sense described above, this should facilitate the augmentation and modification of knowledge structures which is an integral part of such learning.

With respect to these arguments, patterns in the ICMAUS scheme are modular in the sense used by Shortliffe and, like rules in an expert systems, they have the advantage that they can be added or deleted from a knowledge base without the need to examine a whole tree or network.

The apparent advantage of modular knowledge compared with alternatives has not gone unchallenged. Pearl [5, pp. 7-8] argues that the way new knowledge can 'undercut' an otherwise sensible inference (one example of which was discussed in Section 7) argues against modularity of knowledge. However, it seems that, with knowledge expressed as patterns, it is possible to model this kind of interaction between items of knowledge (see Section 7). It is also possible to model the similar kind of interaction which can be seen in nonmonotonic reasoning (Section 5).

8.2.3 The working hypothesis that diverse kinds of processing may be integrated within a single scheme

Given the goal of integrating such things as unsupervised inductive learning, parsing, best-match pattern recognition and information retrieval, probabilistic reasoning, planning and problem solving, and others within a single scheme, it seemed necessary, as with inductive learning, to adopt a simple, uniform, format for knowledge which would lend itself to these diverse kinds of processing.

8.2.4 Representation of the syntax of natural language

The major assumption in Sections 8.2.1 and 8.2.3 is that patterns as they are defined in this research can indeed represent complex forms of knowledge in an efficient and succinct manner.

The best evidence to date relates to the representation of syntactic forms in natural language. In [14], examples are presented showing that one-dimensional patterns are adequate for the represention of context-free phrase-structure grammars and, contrary to what one might expect from such a simple format, they can can represent quite subtle 'context-sensitive' aspects of the syntax of natural language.

The key to this expressive power, is the relatively 'powerful' search processes in the ICMAUS framework, embodied in the SP model.

8.2.5 Representation of other kinds of knowledge

There are many parallels between problems of representing the structure of a natural language and problems of representing various kinds of non-linguistic knowledge. For example, in [12] (Section 3.3) we saw how non-linguistic patterns with internal structure could be represented in essentially the same manner as sentences or phrases with internal structure. In Section 3.4 of the same article we saw how class hierarchies with inheritance could be represented in the same way. And in Section 3.5 of that article we saw how 'polythetic' categories could be represented in a manner exactly analagous to a phrase-structure grammar.

As we saw in [12], other kinds of structure which can be represented with patterns include networks, trees and rules (Section 4). Other examples from this article include geometric analogies in simplified form (Section 6) and we have seen how patterns can be used to represent the kind of information for which a causal network might otherwise be used (Section 7).

8.3 Frequency and conditional probability

In a Bayesian network, each node stores a table of the conditional probabilities of producing individual outputs and combinations of outputs for various combinations of input. By contrast, each pattern in an ICMAUS store of knowledge has a single value for the frequency of occurrence of the pattern in some language sample (in the case of linguistic applications) or the frequency of occurrence of the entity from some external domain which the pattern represents (see discussion in Section 2.1 of [12]).

In itself, this difference is not of any great significance, especially since, in some applications like decision networks or trees (Section 4.3 of [12]), one ICMAUS pattern may correspond to one row of a table of conditional probabilities.

In terms of simplicity, frequencies of patterns seem to have a substantial advantage over tables of conditional probabilities when there is partial matching between patterns. As was noted in Section 4.4 of [12], any partial match between patterns may be regarded as an expression of an 'if-then' relationship: the 'hit' symbols (those that have been matched) may be seen as the 'if' part of the pattern and the unmatched symbols can be interpreted as the 'then' part of the pattern.

For any one string of symbols of length N which is partially matched to another string, the number of possible 'if-then' relationships for the given string is the number of possible subsequences (H) of hit symbols within the given string. This is:

$H = 2^{N} - 1.$

For all but the very shortest strings, the value of H is very large.

For each 'if-then' relationship, there is a conditional probability. If a single frequency value will do for each pattern, this is clearly preferable to storing what in most cases will be very large numbers of conditional probabilities. As we have seen (Section 2 of [12]) there is, in the ICMAUS scheme, a reasonably simple method of calculating the relative probabilities of alternative conclusions for a given input.

9 Conclusion

In these three articles, I have tried to show the potential of the ICMAUS framework for modelling a wide range of types of probabilistic reasoning. The generality of the framework in this connection, coupled with its apparent potential to model several other aspects of computing, suggests that this approach to understanding probabilistic reasoning merits further investigation and development.

References

1 Evans, T. G. (1968) A Program for the Solution of a Class of Geometric-Analogy Intelligence-Test Questions. In M. Minsky (ed.), *Semantic Information Processing*, Cambridge Mass.: MIT Press, pp. 271-353.

2 Gammerman, A. J. (1993) Geometric Analogy Problems by Minimum Length Encoding, *Proceedings of the 4th Conference of the International Federation of Classification Societies, Paris (IFCS-93)*, August-September, 1993.

3 Gammerman, A. J. (1991) The representation and manipulation of the algorithmic probability measure for problem solving. *Annals of Mathematics and Artificial Intelligence,* 4, 281-300.

4 Ginsberg, M. L. (1994) AI and non-monotonic reasoning. In D. M. Gabbay, C. H. Hogger and J. A. Robinson (Eds.), *Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3, Non-monotonic Reasoning and Uncertain Reasoning*. Oxford: Clarendon Press, pp. 1-33.

5 Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Francisco, California.

6 Rissanen, J. (1978) Modelling by the shortest data description. *Automatica-J,*.IFAC

14, 465-471.

7 Schurz, G. (1998) Probabilistic semantics for Degrande's conditional logic and a counter example to his default logic. *Artificial Intelligence* 102(1), 81-95.

8 Shortliffe, E. H. (1976). *MYCIN: Computer-Based Medical Consultations*, Elsevier, New York. Based on a PhD thesis, Stanford University, Stanford, CA, 1974.

9 Solomonoff, R. J. (1964). A formal theory of inductive inference. Parts I and II. *Information and Control* 7: 1-22 and 224-254.

10 Wallace, C. S. and Boulton, D. M. (1968) An information measure for classification. *Computer Journal,* 11 (2), 185-195.

11 Wolff, J. G. (submitted, a) Probabilistic reasoning as information compression by multiple alignment, unification and search (I): introduction.

12 Wolff, J. G. (submitted, b) Probabilistic reasoning as ICMAUS (II): calculation of probabilities, best-match pattern recognition and information retrieval, and reasoning with networks, trees and rules.

13 Wolff, J. G. (submitted, c) Probabilistic reasoning as ICMAUS (III): hypothetical reasoning, geometric analogies, default values, nonmonotonic reasoning, and modelling 'explaining away'. (This article).

14 Wolff, J. G. (submitted, d) Parsing as information compression by multiple alignment, unification and search. Two articles on which this paper is based ("Parsing ... SP52" and "Parsing ... Examples") may be obtained from http://www.sees.bangor.ac.uk/~gerry/sp_summary.html.

15 Wolff, J. G. (1988) Learning syntax and meanings through optimization and distributional analysis. In Y. Levy, I. M. Schlesinger and M. D. S. Braine (eds.), *Categories and Processes in Language Acquisition.* Lawrence Erlbaum, Hillsdale, NJ. Reprinted in Chapter 2 of Wolff (1991).

16 Wolff, J. G. (1982) Language acquisition, data compression and generalization. *Language & Communication*, 2, 57-89. Reprinted in Chapter 3 of Wolff (1991).