In: Proc. AAAI-88 Workshop on Case-Based Reasoning, pp24-29, Eds: E. L. Rissland and J. A. King (Aug 1988), http://www.cs.utexas.edu/users/pclark/papers

Representing Arguments as Background Knowledge for the Justification of Case-Based Inferences

Peter Clark (pete@turing.ac.uk)
The Turing Institute, 36 N.Hanover St
Glasgow, UK

Abstract

This paper examines the representation of background knowledge and its use in *case-based reasoning*. Case-based reasoning can be viewed as a particular form of problem-solving, based on the assessment of similarity of a new case to previously encountered cases, and the subsequent inference that an old solution also applies to the new case. To justify such inferences, we present a representation of background knowledge as *arguments* for and against a conclusion given known facts, rather than as statements in logic or probabilistic relations. Cases are characterized by the set of arguments for and against a hypothesis of interest, and the resolution of conflicting arguments for a new case is obtained by firstly locating an old case where the same or a similar conflict occurred, then secondly transferring the resolution from the old case to the new case. In this way, we are able both to represent weak, possibly conflicting fragments of background knowledge and also learn about the relative strengths of such arguments from cases where the outcome of conflicting arguments is known. We provide a description of this method in logical form, and analyze the assumptions under which it is valid, its limitations and possible future extensions.

1 Introduction

This paper presents a formalism for expressing background knowledge for a problem-solving task, in which knowledge is expressed not by a set of logical axioms but as *arguments* for and against a hypothesis being true. A declaration of the relative strengths of arguments may be supplied by the user beforehand, or learned from experience as new examples are encountered with known outcomes. We present this formalism as a method for *case-based reasoning*, although it can also be used in other forms for reasoning.

2 The Problem

2.1 Case-Based Reasoning

Case-based reasoning can be viewed as a particular form of problem-solving, based on the assessment of similarity of a new case to previously encountered cases and the consequent transfer of an old solution to the new case. Such inferences are typically justified using background knowledge rather than by statistical trends observed amongst the known cases (cf. rule induction methods). Known cases act as 'islands of certainty', and background knowledge is used to relate new cases to these 'islands'. We can define such reasoning as the process of inferring that a *concluding property* Q holds of a particular situation or object T (the *target*) from the fact that T shares a property or set of properties P with another situation/object S (the *source*) that has property Q. The process may be summarized schematically as follows:

$$\frac{\begin{array}{l}\mathtt{P(S)} \wedge \mathtt{Q(S)} \\ \mathtt{P(T)}\end{array}}{\mathtt{Q(T)}}$$

Under some conditions this process is referred to as analogy, although there is some debate as to what these conditions are. For example, Gentner has argued that when the similarity P is solely some 'abstract relational structure' then this process is one of analogy [Gentner, 1988]. We merely note that the processes of case-based reasoning and analogy are closely related.

An important property of case-based reasoning is that neither the previous case nor the background knowledge alone is sufficient to infer the conclusions reached. Given solely a previous case with no other background knowledge, as in the above argument, there is clearly insufficient data for concluding that Q(T) is a justifiable fact to infer. Similarly, background knowledge alone should not be able to conclude Q(T) if the use of previous cases in reasoning is to be non-redundant (the *non-redundancy problem* [Davies and Russell, 1987]). From the case and background knowledge together though, we are able to draw conclusions – we can view this as:

(i) using background knowledge to justify the transfer of solutions from old to new cases
(ii) using specific cases to strengthen background knowledge

It should be noted that these are two equivalent ways of viewing the same process. Case-based reasoning is typically described in terms of (i), but it is also useful to view such reasoning in terms of (ii) whereby examples extend background knowledge to allow previously unattainable conclusions to be drawn.

2.2 Justifying Case-Based Inferences

Given solely a database of previous cases, there is insufficient information for defining under what conditions case-based inferences are and are not justified. Consequently, we need to ask what extra background knowledge *is* required to distinguish between valid and invalid conclusions, and how that knowledge should be used. We review some approaches already taken.

Justifying that a conclusion from a previous case also applies to a current case requires justifying that the properties the cases have in common are 'relevant' to the conclusion (ie. that changing the properties will cause the conclusion's truth to change), and that other properties are irrelevant. By using previous cases, we are able to relax the requirement for specifying the relationships between properties and conclusions exactly, and instead merely represent that some relationship of relevance does exist.

One method of representing the relevance of a feature to a concluded fact is to use some numerical measure or weight on each feature (eg. [Kibler and Aha, 1987]), and the relevance of an old case to a new is found by summing the weights on the features they have in common. Hence, a 'most relevant' old case can be located, and the resulting conclusion for the old case also then drawn for the new case. This method has several problems associated with it though. Most seriously, it permits only a highly restricted representation of how features interact in a domain (allowing only statements of the form 'feature X has degree of influence W_X on the conclusion'). Interactions between features are not representable (features are assumed independent), resulting in a highly limited ability to model the domain. In addition, the selection of appropriate weightings on attributes is a notoriously difficult task. The Protos system [Bareiss *et al.*, 1987] has attempted to overcome some of these deficiencies by measuring relevance on the basis of a (numeric) assessment of user-supplied explanations of how features relate to a conclusion. This allows a more knowledge-based assessment of relevance, but at the price of requiring more numerical parameters to be specified for the system to function.

Davies and Russell [Davies and Russell, 1987] proposed a weaker form of representation of domain knowledge called *determinations*. Here the user expresses that the presence or absence of particular features (ie. the truth or falsity of a statement that a case has certain features) will determine the presence or absence of other features, but is not required to specify exactly for what combination of present and absent features the relationship is valid. Instead, *polar variables* are used to represent the unknown truth values. So, for example, to say that P *determines* Q we would write:

$$(\forall x \ P(x) \Rightarrow Q(x)) \vee (\forall x \ P(x) \Rightarrow \neg Q(x))$$

expressed in determination notation as:

$$P(x) \succ i \, Q(x)$$

where i is a polar variable. The values of polar variables which make such expressions true are found *from examples*, and thus the combination of the determination rule and a previous case allows the drawing of conclusions about a new case.

There are, however, some drawbacks with the use of determinations for reasoning also. Firstly there are often many features which determine a conclusion. As we require an exact match of determining features between a current and previous case to draw a conclusion, the chances of this match becomes increasingly unlikely as the number of such features grows (assuming uniform distribution of examples in example space). Secondly, background knowledge is often of a slightly different form – we would usually not say "whether or not a person's lazy, experienced and was previously sacked determines whether or not they will be a good employee", but rather that "being experienced suggests that they will be a good employee, but their being lazy and previously sacked suggests the opposite". In other words, we often know more about features than just the minimal fact that they are influential in some way. It is this knowledge we aim to capture in our formalism.

Finally, a representation of weak background knowledge has been proposed by Ashley and Rissland in the domain of law [Ashley and Rissland, 1987], [Ashley and Rissland, 1988]. Their system Hypo includes a representation of the relative (rather than absolute) influences of different features or 'dimensions' on a hypothesis (eg. the presence of feature X constitutes *increased* evidence for conclusion Y). Thus knowledge which is too weak alone to draw conclusions, but is strong enough to do so given previous cases, is representable.

3 A Representation of Arguments

3.1 Aims

The above methods all represent and use background knowledge, which although alone insufficient for drawing conclusions, *does* allow conclusions to be drawn when extra knowledge from previous cases is also used. However, one method of reasoning which experts frequently use in problem-solving is based on the search and use of *arguments* for and against a hypothesis of interest. None of the above methods fully captures this form of reasoning, and consequently in this paper we attempt to define and formalize what arguments are and how they can be used. Our aims are thus to:

- Define what is meant by an argument, and how arguments are used
- Model how previous cases can be used in conjunction with arguments to solve new problems
- Avoid the use of numerical measures of probability or belief, deemed unintuitive and problematic by many (see [Cheeseman, 1988] and subsequent debate for an excellent discussion)

This last aim is particularly important, as it distinguishes this work from other attempts to handle uncertainty with numerical approaches. To address this task we only deal with *relative* strengths of arguments, rather than any absolute measure of strength (eg. as might be represented by a number).

3.2 Properties of Arguments

Here we define the basic properties of arguments for our representation. Following this, we give an example of how arguments can then be used for problem-solving, and then in the next section provide a formalization of these properties using logic.

Arguments are considered as form of implication, which we denote by $A \xrightarrow{arg} B$. If A is true then we say that argument $A \xrightarrow{arg} B$ *supports* (is 'for') B. This implication has different semantics from logical implication ($A \Rightarrow B$), as follows:

1. Where there are only arguments for B, and none against B (ie. none for $\neg B$), then B is true
2. Where there some arguments for B and some against B, we have a *conflict*. Conflicts cannot be resolved

(ie. result in B or ¬B being concluded) on the basis of arguments alone.

3. The same conflict will always resolve in the same way. Thus, if we have a previous case where the same conflict applied, and we know the outcome of that conflict (either B or ¬B found true), then we can infer the same outcome will also occur in the current case
4. Adding an argument in favor of a fact already known to be true cannot make it false (and vice versa)

In section 5, we discuss the underlying assumptions these properties are based on, and the conditions under which these assumptions are valid. First though, we provide an example of using arguments in reasoning.

3.3 Example

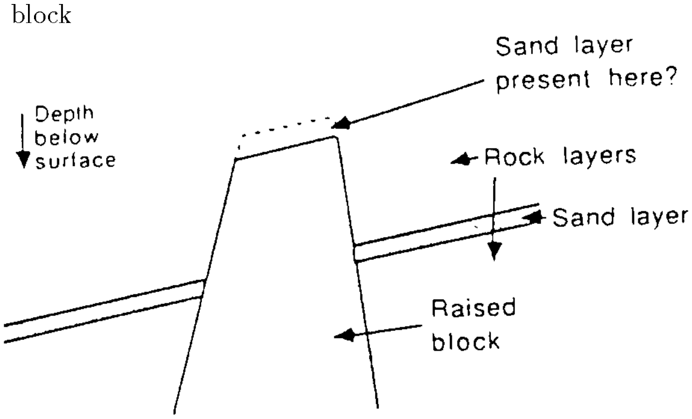
For the purposes of explanation we present an example from the domain of geology. Consider the seismic cross-section shown in figure 1, in which a geologist is interested in determining whether there is a layer of oil-bearing sand over the raised block. A geologist unfamiliar with the region may know the various factors which suggest or do not suggest that there is sand over the block, but does not know enough about the region to determine how these various 'arguments' for or against the hypothesis interact. Examples of such arguments might be, for example:

$$
\begin{array}{lll}
\#1: \texttt{Over\_block(x)} & \xrightarrow{\text{arg}} & \neg\texttt{Sand\_at(x)} \\
\#2: \texttt{Sand\_nearby(x)} & \xrightarrow{\text{arg}} & \texttt{Sand\_at(x)} \\
\#3: \texttt{Late\_fault(x)} & \xrightarrow{\text{arg}} & \texttt{Sand\_at(x)} \\
\#4: \neg\texttt{Late\_fault(x)} & \xrightarrow{\text{arg}} & \neg\texttt{Sand\_at(x)} \\
\#5: \texttt{Environment(x,d)}\wedge & & \\
\quad\;\; \texttt{Unfavorable(d)} & \xrightarrow{\text{arg}} & \neg\texttt{Sand\_at(x)} \\
\#6: \texttt{Fault\_size(x,Small)} & \xrightarrow{\text{arg}} & \texttt{Late\_fault(x)}
\end{array}
$$

As a consequence, from known observations, the geologist can accumulate supporting arguments for and against a case, but as yet be unable to resolve it. In order to find a resolution to the problem, he or she will then search for *already drilled* wells similar to the one under consideration in the nearby region and observe how a similar conflict of arguments was resolved in those cases. Then, using this knowledge, the geologist hypothesises that their resolution will also be the same in the current case.

Thus, we have a two way process:

Figure 1: A seismic cross-section showing a raised fault block



1. Cases with known outcomes supply information about how to resolve conflicting sets of arguments
2. Information about how to resolve conflicting sets of arguments supply information about how to predict the outcomes in new cases

We introduce the relation $\texttt{Stronger}(a_B, a_{\neg B})$ to describe this, where a_B is the set of all arguments *for* some fact B (ie. all $A \xrightarrow{\text{arg}} B$ where A is true) and $a_{\neg B}$ is the set of all arguments *against* B (ie. for ¬B). This predicate describes the relative strengths of different sets of arguments. Initially, this knowledge about argument sets may be partially or even fully unknown. However, as examples with known outcomes are encountered, such knowledge can be gradually learned and applied to known cases.

Consider, for example, the geologist knows the following facts for the to-be-drilled well (Well1 say):

```
Over_block(Well1)
Sand_nearby(Well1)
Environment(Well1,Paleo)
Unfavorable(Paleo)
Fault_size(Well,Small)
```

As a consequence, the arguments *for* Sand_at(Well1) will be the set:

$$\{\#2, \#3\}$$

Similarly the arguments against Sand_at(Well1) are:

$$\{\#1, \#5\}$$

Now:

1. If we are *told* that, in fact, Sand_at(Well1) is true, then we can infer the set of arguments 'for' is stronger than the set of arguments 'against', ie. that
$$\texttt{Stronger}(\{\#2, \#3\}, \{\#1, \#5\})$$
was true.
2. If we wished to *find* whether Sand_at(Well1) was true, we would search for a previous case where similar conflicting argument sets *also* applied, and observe how they were resolved.
A previous case where an identical conflict occurred can thus be used to resolve a current conflict. In addition, because arguments for a fact increase evidence for a fact (by definition), a previous case where a subset of the 'for' arguments was found stronger than a superset of the arguments 'against' (and vice versa) will also resolve the conflict. This property is an important part of the semantics of arguments, for example the following examples of known Stronger relation (possibly acquired from previous cases) are also sufficient to resolve the above conflict:
$$\texttt{Stronger}(\{\#2\}, \{\#1, \#5\})$$
$$\texttt{Stronger}(\{\#2, \#3\}, \{\#1, \#5, \#7, \#8\})$$

We can view this representation as characterizing a case in our knowledge base not by the primitive features which it contains, but by the sets of arguments for and against the hypothesis of interest. As a consequence, we are allowing our domain knowledge, rather than 'best matching' on featural similarities, to determine which previous cases are relevant to a new case. Our domain knowledge provides arguments, and past cases allow the incremental learning of how conflicting argument sets should be resolved.

4 Representation and Semantics

4.1 Preliminaries

In the following, we use the following notation:

$\mathtt{x}, \mathtt{y}, \ldots$	variables
$\underline{\mathtt{x}}, \underline{\mathtt{y}}, \ldots$	sets of variables
$\mathtt{C}, \mathtt{D}, \ldots$	constants
$\underline{\mathtt{C}}, \underline{\mathtt{D}}, \ldots$	sets of constants

When we discuss arguments for and against some fact's truth, we are actually making a meta-level statement about that fact. When referring to a fact itself (rather than its truth value), we follow the convention of Genesereth and Nilsson of writing the fact in quotes [Genesereth and Nilsson, 1987]. This convention is as follows:

$\mathtt{G(C)}$	the value of predicate $\mathtt{G(C)}$ (true/false)
$\mathtt{"G(C)"}$	the predicate $\mathtt{G(C)}$ itself
$\mathtt{"G(<x>)"}$	the predicate $\mathtt{G(x)}$ with $\mathtt{x}$'s *value* substituted in ($<, >$ denote unquoting). (eg. if $\mathtt{x} = \mathtt{Fred}$, then $\mathtt{"G(<x>)"="G(Fred)"}$).

4.2 Notation

The user expresses domain knowledge about arguments in using predicate *schema*[1] of the form below. To translate the semantics of such a schema into logic, we introduce the relation $\mathtt{Arg(N,G)}$ meaning "argument $\mathtt{N}$ applies in support of (ie. 'for') $\mathtt{G}$":

$$\mathtt{N} : \mathtt{F}(\underline{\mathtt{x}}, \underline{\mathtt{y}}) \xrightarrow{\text{arg}} \mathtt{G}(\underline{\mathtt{x}})$$

is a notation for

$$\forall \underline{\mathtt{x}}(\ (\exists \underline{\mathtt{y}}\ \ \mathtt{F}(\underline{\mathtt{x}}, \underline{\mathtt{y}})) \Leftrightarrow \mathtt{Arg(N, "G(<\underline{x}>)")}\ ) \quad (1)$$

loosely translated as:
"**if** the condition $\mathtt{F}$ is true
then argument $\mathtt{N}$ supports (is 'for') $\mathtt{G}$"

In the argument schema, $\mathtt{G}(\underline{\mathtt{x}})$ denotes a single (possibly negated) n-ary relation and $\underline{\mathtt{x}}$ the set of free variables it contains, $\mathtt{F}(\underline{\mathtt{x}}, \underline{\mathtt{y}})$ denotes a boolean combination of relations with free variables $\underline{\mathtt{x}}$ (shared with $\mathtt{G}$) and $\underline{\mathtt{y}}$ (unshared), and $\mathtt{N}$ is a label (eg. a number) for the relation. This is interpreted as meaning that, for all $\mathtt{G}(\underline{\mathtt{x}})$ where the condition $\mathtt{F}(\underline{\mathtt{x}}, \underline{\mathtt{y}})$ is true, argument $\mathtt{N}$ supports (is 'for') the conclusion $\mathtt{G}(\underline{\mathtt{x}})$. For example the following argument

$$\#1 : \mathtt{Owns\_car(x,c)} \wedge \mathtt{Big(c)} \xrightarrow{\text{arg}} \mathtt{Rich(x)}$$

expresses that if someone owns a big car then this is a supporting argument for their being rich. If we knew that, say, $\mathtt{Owns\_car(Fred, Lotus)}$ and $\mathtt{Big(Lotus)}$ then from equation 1 we conclude

$$\mathtt{Arg(\#1,"Rich(Fred)")}$$

Similarly, we can now collect the set of *all* arguments which support some fact $\mathtt{G}(\underline{\mathtt{C}})$. We denote this set using the predicate $\mathtt{Args}$ as follows:

$$\forall \mathtt{a_G}, \underline{\mathtt{x}}\ \ \mathtt{Args(a_G, "G(<\underline{x}>)")}$$
$$\textit{iff}\ \ \mathtt{a_G} = \{\mathtt{n} | \mathtt{Arg(n, "G(<\underline{x}>)")}\}$$

[1] A predicate schema is obtained from a sentence (closed formula) by optionally replacing object constants with variables and optionally removing quantification from variables. As a consequence, a schema may contain free variables

4.3 Semantics of Arguments

We now consider the semantics of arguments, based on reasoning with the sets of arguments $\mathtt{a_G}$ 'for' and $\mathtt{a_{\neg G}}$ 'against' some fact $\mathtt{G}(\underline{\mathtt{C}})$.

Firstly, we say that sets of arguments for and against a conclusion will completely *determine* the truth of that conclusion. In other words, conflicts involving the same sets of arguments will always result in the same conclusion. This can be expressed in logic as:

$$\forall \mathtt{a_G}, \mathtt{a_{\neg G}}((\forall \underline{\mathtt{x}}\ \mathtt{Args(a_G,"G(<\underline{x}>)")} \wedge \mathtt{Args(a_{\neg G},"\neg G(<\underline{x}>)")} \Rightarrow \mathtt{G}(\underline{\mathtt{x}}))$$
$$\vee (\forall \underline{\mathtt{x}}\ \mathtt{Args(a_G,"G(<\underline{x}>)")} \wedge \mathtt{Args(a_{\neg G},"\neg G(<\underline{x}>)")} \Rightarrow \neg \mathtt{G}(\underline{\mathtt{x}})))$$

loosely translated as:
"For all cases with the same arguments for and against $\mathtt{G}$,
either $\mathtt{G}$ will always be true
or $\mathtt{G}$ will always be false"[2]

If all the $\mathtt{G}(\underline{\mathtt{x}})$s are true, we say that the set of arguments $\mathtt{a_G}$ is *stronger* than the set $\mathtt{a_{\neg G}}$, and vice versa if all the $\mathtt{G}(\underline{\mathtt{x}})$s are false. In order to aid comprehensibility and ease bookkeeping operations within the system, we introduce a relation $\mathtt{Stronger}$ to express this fact:

for all $\mathtt{G}(\underline{\mathtt{C}})$ with sets of arguments $\mathtt{a_G}$ 'for' and $\mathtt{a_{\neg G}}$ 'against':

$$\left.\begin{array}{lcl} \mathtt{G}(\underline{\mathtt{C}}) & \Leftrightarrow & \mathtt{Stronger(a_G, a_{\neg G})} \\ \neg \mathtt{G}(\underline{\mathtt{C}}) & \Leftrightarrow & \mathtt{Stronger(a_{\neg G}, a_G)} \end{array}\right\} \quad (2)$$

Thus, new cases provide new knowledge of the $\mathtt{Stronger}$ relation between argument sets, and this can conversely be used to resolve sets of conflicting arguments for new cases.

With this definition, we can observe how to resolve conflicts between argument sets by locating cases where an identical conflict has occurred in the past. We now express a second semantic property of arguments, namely that adding an argument in favor of some fact $\mathtt{G}(\underline{\mathtt{x}})$ can only 'increase the strength' of evidence for $\mathtt{G}(\underline{\mathtt{x}})$ (ie. cannot change its truth value). We can express this simply as follows:

$$\forall \mathtt{a_G}, \mathtt{a_{\neg G}}, \mathtt{a'_G}, \mathtt{a'_{\neg G}}\ \ \mathtt{Stronger(a_G, a_{\neg G})} \Rightarrow \mathtt{Stronger(a'_G, a'_{\neg G})}$$
$$\mathtt{where}\ \ \mathtt{a'_G} \supseteq \mathtt{a_G}, \mathtt{a'_{\neg G}} \subseteq \mathtt{a_{\neg G}} \quad (3)$$

It should be noted that the $\mathtt{Stronger}$ relation is not assumed to be transitive[3] and thus does not define an ordering on sets of arguments. This is because we are trying to model an expert's reasoning, where sometimes transitivity of arguments is not observed. Instead, we only compare cases with either identical arguments, or argument pairs based on the subset-superset relation above (eqn 3).

Finally we state that any set of arguments is at least stronger than no arguments:

$$\forall \mathtt{a_G}\ \ \mathtt{a_G} \neq \{\} \Rightarrow \mathtt{Stronger(a_G, \{\})} \quad (4)$$

and that the relation is antisymmetric:

$$\forall \mathtt{a_G}, \mathtt{a_{\neg G}}\ \ \mathtt{Stronger(a_G, a_{\neg G})} \Leftrightarrow \neg \mathtt{Stronger(a_{\neg G}, a_G)}$$

[2] This can be expressed using Davies and Russell's notation for determinations [Davies and Russell, 1987] as:

$$\mathtt{Args(a_G, "G(<\underline{x}>)")} \wedge \mathtt{Args(a_{\neg G}, "\neg G(<\underline{x}>)")} \succ \mathtt{i}\,\mathtt{G}(\underline{\mathtt{x}})$$

$\mathtt{i}$ being a *polar variable* representing the truth value of $\mathtt{G}(\underline{\mathtt{x}})$.

[3] ie. $\mathtt{Stronger(A,B)} \wedge \mathtt{Stronger(B,C)} \not\Rightarrow \mathtt{Stronger(A,C)}$

4.4 Computation with Arguments

The semantics of arguments are defined in terms of the sets a_G 'for' and $a_{\neg G}$ 'against' some fact $G(\underline{C})$. However, in a practical situation, there will in general be the problem that we may not be able to determine these sets from our existing theory. This will occur if neither the truth nor falsity of an argument's condition can be determined, either due to limited computational resources or incompleteness of the theory. In order to cope with this problem, we show how the semantics can be re-written in terms of sets of arguments which *are* findable by the reasoning system.

Consider we have the following four arguments

$$\#1 : A(x) \xrightarrow{arg} G(x)$$
$$\#2 : B(x) \xrightarrow{arg} G(x)$$
$$\#3 : C(x) \xrightarrow{arg} G(x)$$
$$\#4 : D(x) \xrightarrow{arg} \neg G(x)$$

and also know the facts:

$$A(Fred)$$
$$\neg C(Fred)$$

Here we have insufficient information to tell whether #2 is an argument for G(Fred) or not, as we are unable to tell whether the condition B(Fred) is true. Consequently, we cannot determine the set a_G of arguments 'for' G(Fred). However, we *can* determine lower and upper bounds on this set a_G by finding those sets of arguments which can be shown to apply and can't be shown to not apply respectively. For example, we can easily conclude from the above that:

$$\{\#1\} \subseteq a_G \subseteq \{\#1, \#2\}$$

Consequently, we will now show how the above semantics for arguments can be simply rewritten in terms of lower and upper bounds. Following from this, we show that the tighter these bounds, the more likely we are to be able to draw conclusions about new cases from old ones. Thus we show there is a trade-off between expending resources and the ability to reach conclusions.

4.4.1 Reasoning with Bounds on Arguments

Consider we have established lower and upper bounds on the set of arguments a_G for some fact $G(\underline{C})$. We denote these bounds a_G^- and a_G^+ respectively, ie:

$$a_G^- \subseteq a_G \subseteq a_G^+$$

Using equation 3, we can rewrite equation 2 as follows. Firstly, if we wish to know the truth of $G(\underline{C})$, then the following Stronger relations will allow us to deduce it:

$$\left.\begin{array}{lcl} \mathtt{Stronger}(a_G^-, a_{\neg G}^+) & \Rightarrow & G(\underline{C}) \\ \mathtt{Stronger}(a_{\neg G}^-, a_G^+) & \Rightarrow & \neg G(\underline{C}) \end{array}\right\} \quad (5)$$

This follows straightforwardly from eqn 3 as

$$\mathtt{Stronger}(a_G^-, a_{\neg G}^+) \Rightarrow \mathtt{Stronger}(a_G, a_{\neg G})$$

Conversely, if we already *know* the truth of a particular case $G(\underline{C})$, we can infer knowledge about the Stronger relations:

$$\left.\begin{array}{lcl} G(\underline{C}) & \Rightarrow & \mathtt{Stronger}(a_G^+, a_{\neg G}^-) \\ \neg G(\underline{C}) & \Rightarrow & \mathtt{Stronger}(a_{\neg G}^+, a_G^-) \end{array}\right\} \quad (6)$$

These equations can be intuitively understood as corresponding to the most conservative estimates we can make of the arguments a_G and $a_{\neg G}$ for and against a fact. For example, if at least all the arguments definitely shown to be in favor of a fact (a_G^-) are stronger than all the arguments that could possibly be against ($a_{\neg G}^+$), it must be the case that the fact is true; the actual set of arguments 'for' (a_G) can only be larger and the set 'against' ($a_{\neg G}$) only smaller, which only strengthens the case for $G(\underline{C})$ being true (eqn 3).

4.4.2 Computational Trade-Off

Given lower and upper bounds on the set of arguments for a fact, we can use the equations 5 to find whether the fact is indeed true. What, then, is to stop us simply using the set of none ({}) and all ({...everything...}) respectively for these bounds? The answer is that the wider the bounds, the less can be concluded about previous cases and consequently the less likely we are to be able to resolve conflicts in new cases. In the extreme of using the the sets of none and all arguments as bounds, all that can be concluded given a fact is true is $\mathtt{Stronger}(a_G^+, a_G^-)$ (eqn 6), ie. Stronger({...everything...}, {}) – which tells us nothing more than we already knew (eqn 4).

As an analogy, consider deciding who will win a game of football given upper and lower bounds on the players who will turn up on each side. Only if I know from some previous game that the lower bound (or less) on one side can beat the upper bound (or more) on the other can I reach a decision. Thus it is in my interest to try to find the tightest bounds possible on who will play for each team in order to improve my ability to decide, by reference to previous games, who will win. If all I know is that somewhere between no-one and everyone could turn up on either side, I will be totally unable to decide who will win; conversely knowing the teams exactly maximizes my likelihood of finding a previous game where some subset of one team beat a superset of the other.

Thus, where the set of arguments which apply to a case cannot be exactly determined, the reasoning system should expend effort to bound this set. For many applications, exhaustive enumeration of known and possible arguments would be the most practical approach to take.

5 Assumptions and their Validity

We have presented a representation of background knowledge in a form weaker than logical implication, and described how it can be used in reasoning. Here we discuss the underlying assumptions and future extensions required to this work.

The basic assumptions the formalism makes are that:

- Argument conflicts will always resolve the same way
- Adding arguments for a conclusion always strengthens the case for it.

Underlying these two assumptions are several strong, basic assumptions:

1. We assume that the data we have about a given case is sufficient to *completely determine* the values of any unknown features we are interested in
2. We assume that the set of arguments provided by the user is *sufficiently complete* to avoid the same arguments being resolved in different ways for different

cases. If there is some additional argument which may apply in a conflict and is significant enough to change its outcome, but the system is unaware of it, then the assumption that conflicts known to the system will always be resolved the same way will be violated

3. We assume that the arguments provided are *correct*, such that adding an argument in favor of a case will only strengthen the case

These assumptions constrain the applications for which such reasoning is valid to those where extensive weak background knowledge and detailed information about cases are available. The domain of legal reasoning is one which is particularly suitable.

Thus we are assuming that a complete database of arguments has been supplied by the user beforehand. In particular, the user must currently supply all the arguments against a fact as well as for it; for example as well as stating that properties 'feathers', 'wings' and 'beak' all constitute evidence for 'bird', the user must also state they constitute evidence against 'reptile', as 'bird' and 'reptile' are mutually exclusive. To state that all evidence for X is also evidence against anything mutually exclusive with X is a large requirement on the user, and to automate the enumeration and collection of arguments would be a useful extension.

There are several other ways in which this theory could be extended to relax these assumptions, and we briefly speculate on these here. Most importantly, cases may arise where the same arguments have resolved themselves in different ways in the past. Firstly, some variation on taking a 'majority verdict' on the outcome could be applied. Secondly, the outcomes of cases where the resolution of conflicts doesn't logically imply (by our formalism) the outcome of a current conflict could also be taken into account. Thirdly, the presence of identical arguments resolved in different ways is suggestive of an incompleteness or incorrectness in the domain knowledge of arguments, and may trigger extension or alteration of the knowledge. All these options, though, move from a logical towards a probabilistic theory of a domain. Further research is needed to extend this formalism in this way.

Finally, it should be noted that in this formalism we are making the same assumptions as Davies and Russell in their theory of determinations [Davies and Russell, 1987], and that the method presented here can be viewed as an extension of this work. The set of arguments for a fact can be viewed as a factorization of the determination for that fact, where the (possibly large) set of determining features is split into subsets and each set labelled 'for' or 'against'. Additional conclusions can be drawn using background knowledge in the form of arguments which could not be using determinations, for example given the arguments $A(x) \xrightarrow{arg} Q(x)$, $B(x) \xrightarrow{arg} Q(x)$ and $C(x) \xrightarrow{arg} \neg Q(x)$, and a previous case $\{A(C), \neg B(C), C(C), Q(C)\}$, we can now conclude $Q(D)$ for the new case $\{A(D), B(D), C(D)\}$, a conclusion not possible using determinations.

6 Discussion and Conclusion

Case-based reasoning involves the use of a weak domain theory to justify the transfer of conclusions from previous cases to new cases, when the domain theory alone is unable to perform such a task. The essential issues, then, which must be addressed in case-based reasoning are how to represent such weak knowledge and how to justifiably use it in combination with past cases to reach such conclusions. In this paper, we have described a representation of such weak knowledge in the form of *arguments* for and against a case, and a method by which such arguments can be used to justify applying old solutions to new cases. We have presented a logical formalism for such a method and discussed the assumptions under which conclusions arrived at in this way are correct. We can view this method as characterizing a case in a knowledge base not by the primitive features which describe it but by the arguments for and against the hypothesis of interest.

One extension required of this work, on which we have briefly speculated, is to relax the assumption that the same set of conflicting arguments will always resolve the same way. In this paper, in addition to presenting a knowledge-based theory of case-based inference based on this assumption, we hope to have presented a basis on which such an extended theory relaxing this assumption can be formed.

Acknowledgements

This research arose out of work with Enterprise Oil plc, and I particularly thank Dave Rhodes and Tony Scurr for their involvement. Thanks also to Tim Niblett, Simon Grant and Robin Boswell for valuable discussions on earlier versions of this paper.

References

[Ashley and Rissland, 1987] Kevin D. Ashley and Edwina L. Rissland. Compare and contrast, a test of expertise. In *AAAI-87*, pages 273–278, 1987.

[Ashley and Rissland, 1988] Kevin D. Ashley and Edwina L. Rissland. Dynamic assessment of relevancy in a case-based reasoner. In *Fourth Conference on AI Applications*, pages 208–214, IEEE, LA, 1988.

[Bareiss *et al.*, 1987] E. R. Bareiss, B. W. Porter, and C. C. Wier. PROTOS: an exemplar-based learning apprentice. In P. Langley, editor, *Proc. 4th International Workshop on Machine Learning*, Kaufmann, Ca, 1987.

[Cheeseman, 1988] Peter Cheeseman. An inquiry into computer understanding. *Computational Intelligence*, 4(1), Feb 1988. (followed by 23 commentaries).

[Davies and Russell, 1987] Todd R. Davies and Stuart J. Russell. A logical approach to reasoning by analogy. In *IJCAI-87*, pages 264–270, 1987.

[Genesereth and Nilsson, 1987] Michael R. Genesereth and Nils J. Nilsson. *Logical Foundations for Artificial Intelligence*. Kaufmann, Ca, 1987.

[Gentner, 1988] Dedre Gentner. Analogical inference and analogical access. In Armand Prieditis, editor, *Analogica*, pages 63–88, Kaufmann, Ca, 1988.

[Kibler and Aha, 1987] Dennis Kibler and David W. Aha. Learning representative exemplars of concepts: an initial case study. In P. Langley, editor, *Proc. 4th International Workshop on Machine Learning*, Kaufmann, Ca, 1987.