Models of Similarity for Case-Based Reasoning

Hugh Osborne
School of Computing and Mathematics
University of Huddersfield
(h.r.osborne@hud.ac.uk)

Derek Bridge
Department of Computer Science
University College, Cork
(d.bridge@cs.ucc.ie)

Abstract

We motivate and present *similarity metrics*, the similarity measurement framework we have developed for use within Case-Based Reasoning Systems. In this framework similarities are values from *any* type on which a complete lattice is defined. This gives us a wide range of intuitive ways of measuring similarity and a large number of ways in which different metrics can be combined. The paper concludes by showing how the framework might be deployed in categorisation.

Context

Case-Based Reasoning

Case-Based Reasoning (CBR) may be used in domains for which the following 'slogan' holds true:

Similar problems have similar solutions.

Past problem-solving experiences are encoded as *cases* and stored in a *case base*. Each case typically includes a *description* of the problem and the *solution* found. A new problem is used as a *probe* to interrogate the case base. The CBR system *retrieves* cases that have problem descriptions that are *similar* to that of the probe. The solutions from one or more of the most similar cases retrieved might, after *adaptation*, be used to solve the new problem (Aamodt & Plaza, 1994).

The focus of our work and of this paper is the similarity-based retrieval part of the CBR process. We have been developing a general framework for measuring similarity that both subsumes and extends many existing approaches (Osborne & Bridge, 1996a; 1996b; 1997a; 1997b; 1997c). Before giving an overview of the framework, we make explicit some of our assumptions and factors that have motivated our approach.

Some assumptions

We assume that each case comprises a number of attributes, or features, with values that may be numeric, symbolic or structured (e.g. sets). While a case representation based on attribute-value pairs may seem restrictive, in fact it imposes no substantive restrictions at all. All that we assume is the existence of suitable *projection functions*, π_i for each attribute i, which, when applied to a case, deliver the value of attribute i. This allows our framework to accommodate a variety of case-based work:

- in some systems only a subset of the case information might be stored as fields of a case record, the rest being, e.g., part of an indexing structure (Barletta & Mark, 1988);
- in other systems cases have a distributed representation, e.g. the use of case 'snippets' (Redmond, 1990);
- and in some systems, certain attributes of a case, e.g. the so-called 'deep features' (Ashley & Rissland, 1988), might not be stored at all but, rather, they would be inferred from (stored) 'surface features' (Koton, 1988).

By assuming that projection functions can be implemented to deliver the desired attribute-value pairs, no matter how and where they are stored, our treatment of similarity is isolated from these differences.[1]

Note also that the use of attribute-value pairs does not preclude the representation of recursive concepts. One particular example is the use of feature structures as described by Jantke (1994) and Plaza (1995). We are confident that our model subsumes this approach.

Following fairly well-established practice (e.g. Tversky & Krantz, 1970), we conceptualise computation of $c_1 \sim c_2$, the similarity of cases c_1 and c_2, as the combination of comparisons of the individual attribute-values in the two cases. The similarity of each pair of corresponding attribute-values in the two cases, $\pi_i(c_1)$ and $\pi_i(c_2)$, is computed by a function $\sim_i$. Then, some function f combines the individual similarities:

$$c_1 \sim c_2 = f(\pi_1(c_1) \sim_1 \pi_1(c_2), \ldots, \pi_n(c_1) \sim_n \pi_n(c_2))$$

The application of a similarity measure can be realised in a number of ways. Broadly, though, CBR systems take either a *representational* or a *computational* approach (Porter, 1989).

In the representational approach, cases reside in a data structure in which, e.g., proximity denotes similarity. Such approaches can be efficient: they are effectively optimised towards retrieval according to the 'hard-coded' similarity measure. This can be of especial value when similarity measurement requires the application of large bodies of domain-specific knowledge: the knowledge will be applied once per

[1]We also assume that projection functions can deliver any of the types of case information represented in case bases (e.g. facets of the problem description, its wider situation or context, its solution, the solution's outcome, etc.); this allows similarity measures to be sensitive to aspects of the case other than the problem description (e.g. the adaptability of the solution (Smyth & Keane, 1996)).

case at case base update time, rather than being applied afresh on every retrieval (Porter, 1989). However, this form of optimisation can lead to a loss of flexibility (McCartney & Sanders, 1990) as it may be hard or inefficient to access the case base in different ways.

The computational approach, on the other hand, will, in its most extreme form, compute similarity 'from scratch' on each retrieval. This can be a flexible approach as nothing is hard-coded; it may be more amenable to user manipulation of the similarity measure (see below) or even manipulation through some learning process (e.g. Richter & Wess, 1991); but there may be an efficiency price to be paid.

In other systems, the case base is indexed, and case base interrogation is a two-stage process (Aamodt & Plaza, 1994): a retrieval step exploits the indexes to retrieve a possibly useful set of cases, then a similarity measure is applied to these cases. This is clearly a compromise between pure representational and pure computational approaches.

Although our presentation has a computational flavour, nothing in our framework precludes hard-coding of all or parts of the similarity measures and so our framework is general enough to cover representational approaches too.

Some motivation

We have been most concerned with *interactive* CBR systems. A human user is responsible for describing the new problem situation and judging the quality of the cases retrieved. For these systems (and possibly non-interactive CBR systems too), our contention is that there is generally *no fixed notion of similarity*: the user must be allowed to change not only the probe but also the similarity measure. This can be vividly illustrated by considering the use of CBR in on-line shopping systems (Wilke, 1997) where, e.g., each case in the case base is a product (the 'description' parts of the cases characterise the products; the 'solution' parts simply identify the products).[2] The probe may give features of the customer's ideal product; cases in the case base (products) that are similar to this ideal are recommended to the user. Similarity is taken as a sign of substitutability: if a holiday on Crete is similar enough to one on Rhodes, the customer might be prepared to settle for either.

We may need different similarity measures for different users (customers), for different goals and for different queries within a dialogue. The way that individual attributes are compared (the $\sim_i$ from above) may need to change; and the way that attribute matches are combined (function f from above) may also need to change. Imagine that the probe is the same in all the following examples. (Of course, in practice it too may change). Specifically, suppose the probe mentions destination Greece, price £ 300, travel by plane, duration one week, a four-star hotel, and golf, swimming and windsurfing as desired activities.

[2]For the purposes of this illustration, we are ignoring the fact that, for customisable products such as holidays, the final recommendation might be an amalgam of information from several cases.

- Different users (customers):
  - different $\sim_i$: one customer may rate destinations Turkey and Greece as similar (substitutable); clearly, other customers would not;
  - different f: one customer may rate price as the most important attribute; another might not.
- Different goals, e.g. selecting a winter holiday versus selecting a summer holiday:
  - different $\sim_i$: for a summer holiday, combined plane and ferry travel might be acceptable; but, for a winter holiday, it might not;
  - different f: for the winter holiday, matching well on the desired holiday duration may be important (e.g. to return to work on time), but, in the summer, when there might be more flexibility, it could be a less important factor.
- Different queries within a dialogue (refinement of queries):
  - different $\sim_i$: perhaps over the course of the dialogue the customer comes to rate three- and four-star hotels as more similar than s/he did at the start of the dialogue;
  - different f: at different points in the dialogue, the customer might want to experiment by seeing what different holiday recommendations are made when s/he is insistent and less insistent on good matches with the desired set of sporting activities.

What emerges clearly from these examples is that similarity measures need to be intuitive so that end-users can readily formulate and revise their own similarity measures. This will be the major criterion we use to judge the models of similarity that we describe in the next part of this paper and it motivates our own similarity framework.

Models of Similarity

In this section we will discuss two models of similarity, *absolute* and *relative* similarity. We will give mathematical models for both types of similarity, and argue for a third, more general model called *metric* similarity, which both subsumes and extends absolute and relative similarity. We will continue to use the symbol '$\sim$' to represent similarity measures.

Absolute Similarity

The simplest notion of similarity is one which classifies two objects as being similar or not similar — e.g. Spanish beach holidays and Greek beach holidays are similar, but Spanish beach holidays and Alpine walking holidays are not. In this model, similarity is a binary relation, i.e. $\sim :: \alpha \to \alpha \to \mathsf{Bool}$ (pronounced "the type of $\sim$ is α to α to Bool", so that $\sim$ takes two arguments of type α and returns a boolean value). This relation will be reflexive ($x \sim x$ — or, equivalently, $x \sim x = \textbf{True}$) since any object is similar to itself, and symmetric ($x \sim y \Leftrightarrow y \sim x$ — which can also be stated as $x \sim y = y \sim x$), since if one object is similar to another then the reverse holds as well. These properties are summarised in table 1(i). (There are, of course, opposing views about this. For example, Jantke (1994), citing the cognitive psychology literature, questions the assumption of symmetry. However Richter (1992) argues that similarity is symmetric and linguistic evidence to the contrary is concerned with pragmatics and not the truth or falsity of the similarity relation. In our model symmetry is a direct consequence of the definition of similarity in terms of *excess* (see below). It is a question for further research whether symmetry is an essential property in a purely similarity based approach. For the time being we will assume symmetry.) Note, however, that the relation is not transitive, since, for example, it may be that Spanish beach holidays are considered to be similar to Greek beach holidays, and Greek beach holidays similar to Greek island cruises, but that Spanish beach holidays are not considered to be similar to Greek island cruises.

Given two or more measures of similarity of this kind, e.g., for different attributes of a case, it is easy to combine them to give new similarities with obvious intuitive meanings — for example conjunction (x is similar to y only if it is similar on both attributes) and disjunction (similarity on either one of the attributes is sufficient to establish the similarity of two cases as a whole). While this gives a simple model, it does not correspond particularly well to people's intuitive concept of similarity in which there is a notion of relative similarity — Greek beach holidays are *more* similar to Spanish beach holidays than they are to Greek cruises. A common solution to this problem is outlined in the next section.

Relative Similarity

A more complex concept of similarity assigns some score — a numeric value — to the degree of similarity. This score is often normalised to be in the interval $[0, 1]$. Similarity is then a dyadic function to numeric values — $\sim :: \alpha \rightarrow \alpha \rightarrow [0, 1]$. Again this function must have certain properties. Firstly, any object is totally similar to itself — $x \sim x = 1$. This corresponds to reflexivity in our definition of absolute similarity. Secondly, the similarity of x to y must equal the similarity of y to x — $x \sim y = y \sim x$ (this corresponds to the symmetry of absolute similarity). See table 1(ii-a).

Again, similarity measures, e.g. on different attributes, can be easily combined. For example, one could take the average similarity of two measures, the maximum, the minimum, the product or numerically-weighted versions of these (to assign different importances to matches on different attributes).

This model is actually a specific instance of a more general model. In this more general model, a similarity is *any* real number. We can also introduce the idea of varying degrees of similarity between different pairs of 'identical' objects. While at first this seems strange, it may have an intuitive motivation. Consider measuring the similarity not just of instances (i.e. concrete objects such as two holidays) but also of abstract concepts (e.g. two kinds of holiday), where these are like concrete instances but are underspecified in some respect (e.g. specifying the destination but not the activities on offer). The degree of similarity of two identical abstract concepts could depend on the degree of specification of the concepts (e.g. 'Greek beach holiday' might be more similar to 'Greek beach holiday' than 'beach holiday' is to 'beach holiday') and/or the variety of concrete instances they describe (e.g. 'beach holiday' might be more similar to 'beach holiday' than 'alpine holiday' is to 'alpine holiday' assuming that there is a wider variety of alpine holidays than beach holidays). We replace the requirement that the similarity of any object to itself be 1 (or some other fixed maximal value) with the condition that an object is at least as similar to itself as it is to any other object. The properties of general relative similarities are given in table 1(ii-b). A good overview of different ways of building relative similarity measures is given in Griffiths' thesis (1997).

This model solves some of the problems of the model of absolute similarity. On occasion it may be a very natural way to compute similarity (e.g. numeric-valued attributes may be especially well-suited to numeric-valued similarity measures). However, in numerous other cases (e.g. especially for attributes that take on symbolic or structured values) assigning a numeric value to similarity is often arbitrary, and hard to justify in an intuitively obvious way. For example, what number adequately characterises the similarity of Greek beach and cruise holidays? It is also, arguably, too often the case that these measures are used in ways that treat only the ordering of the numbers as significant, and this brings dangers. For example, the designers of such a measure might assign a similarity of 0.8 to Greek and Spanish beach holidays and one of 0.5 to Greek beach and cruise holidays. Their intention may be that the user should understand Greek and Spanish beach holidays to be more similar to each other than Greek beach and cruise holidays are; they may not want the user to think that they are more similar to degree 0.3 (i.e. the ordering is significant but the actual quantities are not). But, while the designers may know that the actual quantities are not meaningful, if users in interactive CBR systems have access to the numbers, they may well interpret them as meaningful (particularly if the numbers are not meaningful in some measures but are meaningful in other measures). Finally, while the various ways of combining these measures (averages, maxima, products, etc.) may at first sight appear to have some intuitive meaning (for example, the product is compounding the degrees of similarity), when one considers the possible arbitrariness of the original values, these operators only seem to be increasing the ad hoc nature of the model.

For the reasons given in the previous paragraph, relative similarity measures may be hard to formulate and revise. This led us to develop a new model of similarity.

Metric Similarity

We argue that, though relative similarity is an improvement on absolute similarity, it is still not general enough. We have mentioned its main problem: assigning a numeric score to similarity is often arbitrary and non-intuitive. One simple generalisation that circumvents this is to measure similarity not only using reals but to allow similarity functions that return values from any total order. But this raises another problem: degrees of similarity (particularly once one allows similarity measures that return values other than just booleans and reals) need not be comparable. Consider, for example, the problem of measuring the similarity of sets. Suppose holidays offer activities such as golf, swimming, windsurfing and paragliding, henceforth g, s, w and p. The set $\{g,s,w\}$ is obviously similar (to some degree) to the set $\{g,w,p\}$; it is also similar to $\{g,s,p\}$. However, is $\{g,s,w\}$ more or less similar to $\{g,w,p\}$ than it is to $\{g,s,p\}$? Requiring the value of a similarity to be an element of a total order is too restrictive. Degrees of similarity (as in the example) need not be comparable. So the innovation in our similarity metrics framework is to relax that restriction too. We use similarity functions that return values from partial orders.

In fact, we are a bit more specific than previously stated. We require *complete lattices* (here denoted $\mathcal{L}$), rather than arbitrary partial orders: for us, a similarity measure is a dyadic function to *any* type on which a complete lattice is defined. We will not give details here of the definition of complete lattices. Suffice it to say that they are partial orders that satisfy certain additional properties. Proper definitions are given in, e.g., (Birkhoff, 1967); see also (Osborne & Bridge, 1997b). A good (and only slightly incomplete) understanding of this paper is possible for those who do not know what a complete lattice is, and we proceed on this basis.

The requirements for the similarity measure carry over simply from the previous model, replacing the usual ordering on reals ($\geq$) with the ordering defined on the lattice ($\sqsupseteq_{\mathcal{L}}$). Compare the properties of metric similarites given in table 1(iii) to those of relative similarities in table 1(ii-b).

(i)

Absolute Similarity	
Type	Properties
$\alpha \to \alpha \to \mathsf{Bool}$	$x \sim x = \textbf{True}$ $x \sim y = y \sim x$

(ii) a

Relative Similarity (First Model)	
Type	Properties
$\alpha \to \alpha \to [0,1]$	$x \sim x = 1$ $x \sim y = y \sim x$

(ii) b

Relative Similarity (Second Model)	
Type	Properties
$\alpha \to \alpha \to \Re$	$x \sim x \geq x \sim y$ $x \sim y = y \sim x$

(iii)

Metric Similarity	
Type	Properties
$\alpha \to \alpha \to \mathcal{L}$	$x \sim x \sqsupseteq_{\mathcal{L}} x \sim y$ $x \sim y = y \sim x$

Table 1: Type and properties of similarities

A metric similarity example is given in table 2. In this example we are shown, for each case (holiday) in the case base, the set of sporting activities offered. We are also given a target set, i.e. the value from the probe — our ideal set of activities. Note that no case offers exactly our ideal. The metric similarity used in this example is the disjoint union, i.e. the set of all elements that are in either of the two sets being compared, but not in both. The lattice in which the results of the similarity are to be compared is the inverse of the power set lattice.[3] So one would compute the disjoint union of the probe value and the value in each of the cases and judge which were most similar to the probe using the lattice. The most similar sets are therefore those having the smallest disjoint union with the target set as shown in table 2.[4]

Values in cases:	$\{\{g,w,p\},\{g,p\},\{g,s,w,p\},\{g,s\},$ $\{g\},\{g,w\},\{p\},\{g,s,p\},\{s,w,p\}\}$
Value in the probe:	$\{g,s,w\}$
Metric similarity:	$\cup \setminus \cap$
Lattice:	Inverse of power set lattice
Most similar:	$\{\{g,s,w,p\},\{g,s\},\{g,w\}\}$

Table 2: An example of metric similarity

One strength of our framework is the wide variety of measures that it covers. For example, relative similarities are instances of the framework. One possibly useful definition would be $x \sim y = \text{abs}(x-y)$, the absolute value of their difference. The similarities so-computed would be compared using the lattice defined on the reals by the ordering $\leq$.[5] (Note that this is not quite the same as the relative similarities defined earlier, since it rates low numbers more highly than large numbers.) So, for example, if the probe specifies £ 300 as the ideal price of the holiday, holidays that cost £ 350 are similar to the ideal to degree £ 50 (i.e. abs$(300-350)$); holidays that cost £ 500 are similar to degree £ 200, which is less similar than the £ 350 holidays (because 50 is higher in the lattice than 200).

Other examples of metrics that we could define include:

- Boolean-valued: By ordering the booleans, we get a lattice that could be the result type of a metric that would subsume

[3]The power set lattice is the lattice defined on all subsets of $\{g,s,w,p\}$ and ordered by $\subseteq$. So, e.g., $\{g\}$ is lower than $\{g,s\}$, which is lower than $\{g,s,w\}$. In the inverse of this lattice, $\{g,s,w\}$ is lower than $\{g,w\}$, which is lower than $\{w\}$. $\{g,s\}$ and $\{s,w\}$ are examples from these lattices that are incomparable (neither is a subset of the other).

[4]The way we deal with the more realistic scenario, where the probe activities or any superset of them would be ideal, is beyond the scope of this paper. The approach is based on *excesses* (see below) and is covered in (Osborne & Bridge 1996b).

[5]Of course, $\leq$ defines a total ordering, but these are always lattices (whereas not all partial orders are lattices). To give a *complete* lattice, we require a top and bottom element, and so, strictly speaking, the result type of this metric would need to be, e.g., $\Re \cup \{\infty, -\infty\}$.

absolute similarity.

- Linguistic-hedge-valued: A suitably ordered set of linguistic hedges (*"very", "quite", "fairly"*, etc.) could be the result type of another metric.

- Feature-structure-valued: Where cases are represented by first-order terms or feature structures, their anti-unification gives a similarity metric with a result lattice based on subsumption. (Jantke (1994) and Plaza (1995) describe similarity measures of this kind. We are confident, although we have not yet established it for certain, that their work is subsumed by our framework.)

All these and more are part of the framework. Many will be intuitive, which should make for easy formulation and revision.

In our framework we can also make use of the many operations on lattices that preserve the lattice property.

A lattice homomorphism, for example, is a function which, when applied to a particular lattice, maps values in the lattice to values from another type but 'carries over' the original ordering.[6] Two (of many) uses of these are:

- Conversion between different types of metric. For example, a numeric-valued metric can be converted into a boolean-valued metric by mapping the lattice on the reals to one on the booleans (e.g. using thresholding);

- Capturing levels of indifference. For example, the price metric described above treats £ 350 holidays as more similar to £ 300 holidays than £ 355 holidays are (because, taking the absolute values of the differences, $50 \leq 55$). A user, however, would probably regard both prices as 'just about' equally similar to £ 300. Integer division of similarities by 10 'collapses' intervals of similarities to single numbers. In particular, now, the £ 350 and £ 355 holidays are equally similar to the £ 300 holiday (i.e. $50 \div 10 = 55 \div 10$, where $\div$ is integer division).

Lattice sums, products and prioritisations (Osborne & Bridge, 1997b) allow combination of two (or more) lattices. We use these as the basis of operations for combining metrics. It is striking that we can combine metrics of different result types, *without inter-conversion*. A set-valued metric, for example, can be combined directly with a numeric-valued one. There is no requirement to convert, e.g., the set-valued metric to a numeric-valued metric prior to combining them. (Of course, if conversion is desirable, this too is possible: a lattice homomorphism can be used to carry out the conversion, before use of the sum, product or prioritisation operator.)

By way of example, consider the similarity metrics we have been using on prices and desired activities (the absolute value of the numeric difference and the disjoint union, respectively). Suppose the probe specifies a £ 300 holiday that offers activities g, s and w, i.e. $(300, \{g, s, w\})$, and suppose that the relevant values from the cases in the case base are: $(300, \{g, s\})$, $(300, \{g\})$ and $(400, \{g, s, w\})$. It is obviously easy to compute the similarities for the two individual attributes: $(50, \{w\})$, $(50, \{s, w\})$ and $(150, \emptyset)$, respectively. And we can see that, as far as price is concerned, the first two cases are equally similar to the probe and the third case is less similar to the probe, while, as far as activities are concerned, the third case is more similar than the first, which is more similar than the second. But how would we combine these judgements to determine which case is most similar overall?

It is dealing with this problem that leads others to interconvert (e.g. make both similarities numeric, so that they can be combined using a weighted-average or the like; see (Wilson & Martinez, 1997) for a review of possible approaches). But, in our framework, a metric may have as its result type *any* type on which a complete lattice is defined. There is no reason why we should not define a complete lattice on pairs such as $(50, \{w\})$, $(50, \{s, w\})$ and $(150, \emptyset)$. And, indeed, we have operators that can create such lattices from the individual lattices. For example, the *prioritisation* of the original price lattice over the original activities lattice would give a new lattice in which $(150, \emptyset) \sqsubset (50, \{s, w\}) \sqsubset (50, \{w\})$, i.e. $(50, \{w\})$ is the highest similarity, so the first case is the most similar to the probe.[7] (Obviously, other ways of combing the lattices will give other effects.)

Of course, having formed such a compound metric, nothing now stops further application of lattice homomorphisms. For example, if we have combined two boolean-valued metrics we could then use a lattice homomorphism to give us their conjunction or disjunction.

The wide range of ways metrics can be combined (particularly given that inter-conversion is not necessary) is another strength of our framework.

Excesses

As we have discussed, a major criterion in our work is easy formulation and revision of similarity measures. We believe that it is often easier for a user to 'rank' objects, and even to say *how much* 'better' one object is compared with another, than it is for a user to decide on the degrees of similarity of objects. For this reason, we have also defined rankings (here called *excesses*) (Osborne & Bridge, 1997b, 1997c), and then define similarity measures in terms of these (see below).

The generalisations of similarities described in the previous section of this paper (from absolute to relative to metric) can be charted in our specifications of excesses. There are boolean excess functions, where an object exceeds another or it does not; numerical excess functions, in which an object may exceed another to some numeric degree; and lattice-

[6]Again, we spare the reader the mathematical details.

[7]The prioritisation of $\mathcal{L}_1$ over $\mathcal{L}_2$ (where $\sqsubseteq_1$ and $\sqsubseteq_2$ are the orderings on $\mathcal{L}_1$ and $\mathcal{L}_2$ respectively), yields a new lattice ordered by $\sqsubseteq$:

$$(x_1, x_2) \sqsubseteq (y_1, y_2) = (x_1 \sqsubset_1 y_1) \vee (x_1 = y_1 \wedge x_2 \sqsubseteq_2 y_2)$$

i.e. the ordering is determined by $\sqsubseteq_1$, with $\sqsubseteq_2$ being used to resolve 'ties'.

valued excesses, where excesses are drawn from any type on which a complete lattice is defined, and so need not be comparable. Again, lattice-valued excesses subsume the boolean and numeric valued ones.

These excess functions, and the papers in which they are presented are summarised in table 3.

Excesses		
Name	Type	Osborne & Bridge
Ordinal (i.e. absolute)	$\alpha \to \alpha \to \mathsf{Bool}$	1996a
Cardinal (i.e. relative)	$\alpha \to \alpha \to \Re$	1996b, 1997a
Metric	$\alpha \to \alpha \to \mathcal{L}$	1997b, 1997c

Table 3: Names and types of excess relations

We will use '$\trianglelefteq$' to represent excess relations. $x \trianglelefteq y = v$ is pronounced "y exceeds x by v"; the value v is also called the "excess of y over x". The only property we require of excess relations is a form of transitivity:

$$[(x \trianglelefteq y \sqsubset_{\mathcal{L}} y \trianglelefteq x) \wedge (y \trianglelefteq z \sqsubset_{\mathcal{L}} z \trianglelefteq y)] \Rightarrow (x \trianglelefteq z \sqsubseteq_{\mathcal{L}} z \trianglelefteq x)$$

Forms of reflexivity and antisymmetry are not necessary from a mathematical point of view (e.g. our definition of maxima and the way we define similarity metrics from excess relations both give results that satisfy the properties we expect of maxima and similarity relations without imposing reflexivity and antisymmetry).

An excess relation for holiday prices, $\trianglelefteq_p$, might be defined as:

$$x \trianglelefteq_p y = \textbf{if } (x \geq y) \ x - y \textbf{ else } 0.$$

So, $300 \trianglelefteq_p 250 = 50$ and $250 \trianglelefteq_p 300 = 0$, i.e. a £ 250 holiday exceeds —is better than— a £ 300 holiday by £ 50, and a £ 300 holiday exceeds a £ 250 holiday by £ 0. The excesses are, of course, compared using the ordering defined on the result lattice, which in this case would be the usual $\geq$ ordering on the reals, i.e. exceeding by £ 50 is better than exceeding by £ 0. (Note that for the similarity metric in the previous section we used the ordering $\leq$.)

An excess relation for sporting activities, $\trianglelefteq_a$, might be defined as:

$$x \trianglelefteq_a y = y \setminus x.$$

So, $\{g\} \trianglelefteq_a \{g, s, w\} = \{s, w\}$, i.e. a holiday offering g, s and w exceeds one that offers only g by $\{s, w\}$. The ordering on the result lattice is $\subseteq$, i.e. this excess relation has the usual power set lattice as its result type (see footnote 3). So exceeding by $\{s, w\}$ is better than exceeding by just $\{s\}$ but incomparable with exceeding by $\{g, w\}$.

The result lattices in excess relations may be operated on by lattice homomorphisms and combined using sums, products and prioritisations, giving the same advantages as before.

A user who constructs an excess relation may apply the relation to the case base and take the maxima. Perhaps more usually a user will want to define a similarity metric that can be used in CBR applications. We do this by first defining a distance relation $\leftrightsquigarrow$ from the excess relation:

$$x \leftrightsquigarrow y = x \trianglelefteq y \sqcup_{\mathcal{L}} y \trianglelefteq x$$

where $\sqcup_{\mathcal{L}}$ is the least upper bound in the result lattice $\mathcal{L}$. In the case of $\trianglelefteq_p$, the least upper bound operator is 'max' and $\max(x \trianglelefteq_p y, y \trianglelefteq_p x)$ simplifies to $\mathrm{abs}(x - y)$; in the case of $\trianglelefteq_a$, the least upper bound operator is 'union' and $(x \trianglelefteq_a y \cup y \trianglelefteq_a x)$ simplifies to the disjoint union. These are the similarity metrics we used in the previous section. The only other change is to remember that the result lattice in excess relations ($\trianglelefteq$) and distance functions ($\leftrightsquigarrow$) is the inverse of what is wanted when measuring similarity (i.e. for the similarity metrics we must use $\leq$ not $\geq$ and the inverse of the power set lattice).

Categorisation

Our work on similarities and excesses suggests a generalisation that could be used in work on categorisation: the use of lattice-valued functions.

Suppose, for simplicity of exposition, that we know what categories there are. Then, for each, we need a membership function that will tell us whether (or to what degree) objects belong to that category. Absolute and relative categorisation is familiar (from classical and fuzzy set theory); the authors are not aware of any work that proposes metric categorisation.

Categorisation	
Name	Type
Absolute	$\alpha \to \mathsf{Bool}$
Relative	$\alpha \to \Re$
Metric	$\alpha \to \mathcal{L}$

Table 4: Names and types of categorisation functions

In absolute categorisation, an object is a member of a class or it isn't; in relative categorisation, there is numerically-denoted fuzziness; and, in metric categorisation, membership degrees are values whose relative strengths can be determined from a lattice (although this means that they might be incomparable). Lattice homomorphisms, sums, products and prioritisations may again give advantages to the metric approach (e.g. easy combination of several categorisation functions even when they have different result-types). Table 4 gives the types of these three classes of categorisations.

Categorisation functions may be derived from similarity functions. If we have a similarity function and an object whose category is known (e.g. a prototypical instance, if such exists), we can produce a categorisation function for that category by partial application of the similarity function to the object, i.e. by 'freezing in' the object as the first argument of the similarity function, thus turning the binary similarity function into a unary categorisation function. Obviously, absolute, relative and metric categorisation functions will be derived from absolute, relative and metric similarity functions, respectively.

Given that we can derive categorisations from similarities, the question obviously arises: is the reverse possible — given a categorisation, is it possible, in a natural way, to derive a similarity? The answer is, conditionally, yes — though it is less general than deriving categorisations from similarities and requires the introduction of an operator, $\oplus$, which will, itself, turn out to be a similarity.

Given a categorisation, i.e. a function δ that returns degrees of membership, we can define a similarity $x \sim y = (\delta\, x) \oplus (\delta\, y)$ where, in order that this satisfy the properties of similarities (table 1(iii)), $\oplus$ must itself satisfy the same properties. In other words, $\oplus$ must itself be a similarity of the lattice concerned. (Again, as one might expect, from absolute, relative and metric categorisations we would derive absolute, relative and metric similarities and $\oplus$ will need to be an absolute, relative and metric similarity, respectively.)

There is much that needs investigation in this relationship between categorisation and similarity, not least the question of what to do when categories are not known in advance but must be derived from the data. Metric similarity provides a richer arena for such investigation.

Conclusions

We have presented absolute, relative and metric versions of similarity, excess and categorisation functions. We have shown some of the relationships between these different types of function. Our generalisation to functions that return values from any type on which a complete lattice is defined provides a wider range of intuitive functions and provides for easy combination of different functions.

References

Aamodt, A. & Plaza, E. (1994) Case-based reasoning: Foundational issues, methodological variations, & systems approaches. *AI Communications*, *vol.7(1)*, pp.39–59

Ashley, K. & Risland, E. (1988) Waiting on weightings: A symbolic least commitment approach, *Procs AAAI*, pp.239–244

Barletta, R. & Mark, W. (1988) Explanation-based indexing of cases. *Procs AAAI*, pp.541–546

Birkhoff, G. (1967) *Lattice Theory*. American Mathematical Society

Griffiths, A.D. (1997) *Inductive Generalisation in Case-Based Reasoning Systems*. Technical Report YCST 97/02, University of York Department of Computer Science

Jantke, J.P. (1994) Nonstandard concepts of similarity in case-based reasoning. In H.H.Bock, W.Lenski & M.M.Richter (eds.), *Information Systems & Data Analysis: Prospects, Foundations, Applications*, pp.29–44, Springer

Koton, P. (1988) Reasoning about evidence in causal explanations. *Procs AAAI*, pp.256–261

McCartney, R. & Sanders, K.E. (1990) The case for cases: A call for purity in case-based reasoning. *Procs AAAI Symposium on CBR*, pp.12–16

Osborne, H. & Bridge, D.G. (1996) Parallel Retrieval from Case Bases. *Procs Second UK Workshop on Case-Based Reasoning*, Salford

Osborne, H. & Bridge, D.G. (1996) A Case Base Similarity Framework. In I.Smith & B.Faltings (eds.), *Advances in Case-Based Reasoning*, LNAI 1168, pp.309–323, Springer

Osborne, H. & Bridge, D.G. (1997) *A Formal Analysis of Case Base Retrieval*. Technical Report YCS 281, University of York Department of Computer Science

Osborne, H. & Bridge, D.G. (1997) Similarity Metrics: A Formal Unification of Cardinal & Non-Cardinal Similarity Measures. In D.B.Leake & E.Plaza (eds.) *Case Based Reasoning Research & Development*, pp.235–244, Springer

Osborne, H. & Bridge, D.G. (1997) We're All Going on a Summer Holiday: An Exercise in Non-Cardinal Case Base Retrieval. *Procs Sixth Scandinavian Conference on Artificial Intelligence*, Helsinki

Plaza, E. (1995) Cases as terms: A feature term approach to the structured representation of cases. *Procs First International Conference on Case-Based Reasoning*, Springer

Porter, B.W. (1989) Similarity assessment: computation vs. representation. *Procs DARPA Case Based Reasoning Workshop*, pp.82–84

Redmond, M.A. (1990) Distributed cases for case-based reasoning: facilitating use of multiple cases. *Procs AAAI*, pp.304–309

Richter, M.M. (1992) Classification & learning of similarity measures. *Studies in Classification, Data Analysis & Knowledge Organisation*, Springer

Richter, M.M. & Wess, S. (1991) Similarity, uncertainty & case-based reasoning in PATDEX. In R.S.Boyer (ed.), *Automated Reasoning: Essays in Honour of Noody Bledsoe*, pp.249–265, Kluwer

Smyth, B. & Keane, M (1996) Adaptation-guided retrieval: using adaptation knowledge to guide the retrieval of adaptable cases. *Procs Second UK Workshop on Case-Based Reasoning*, Salford

Tversky, A. & Krantz, D.H. (1970) The dimensional representation & the metric structure of similarity data. *Journal of Mathematical Psychology*, *vol.7*, pp.572–596

Wilke, W. (1997) Case-Based Reasoning & Electronic Commerce. *http://wwwagr.informatik.uni-kl.de/∼lsa/CBR/ECommerce*

Wilson, D.R. & Martinez, T.R. (1997) Improved Heterogeneous Distance Functions. *Journal of Artificial Intelligence Research*, *vol.6*, pp.1–34