Weight Intervals

Conservatively adding quantified uncertainty to similarity

Alex Ferguson Derek Bridge

Department of Computer Science
University College, Cork

a.ferguson@cs.ucc.ie d.bridge@cs.ucc.ie

Abstract

Numeric-valued similarity measures are the traditional means used in Case-Based Reasoning (CBR) systems for case retrieval, and for procedures such as clustering analysis and case classification. Similarity measures are customarily combined using the technique of taking the weighted sum of a vector of similarity values, each determined by the application of a single component measure. Osborne, Ferguson, and Bridge's framework of similarity metrics generalises this treatment, allowing similarity values to be from any partially ordered set. As a consequence of this, however, the usual weighted sum combining form is not generally usable, and alternatives must be (and have been) devised.

Presented here is a partial reconciliation of these notions: using weight-intervals, some of the desirable features of the more general metric approach may be retained, while obtaining results that are a strict generalisation of those possible with conventional weighting.

1. Introduction

In this paper, we discuss techniques of capturing uncertainty of outcome in case retrieval systems, and present a new means of doing so. By this we mean representing to the user, as well as what the 'best' case is, some notion of the degree of confidence in same, and which other cases merit serious consideration. Principally we are concerned with case retrieval in an interactive environment. As well as pure case retrieval, as we assume in our examples, this would include any system where a degree of user interaction is involved prior to (any necessary) adaptation.

Firstly in Section 2, we review the conventional techniques for measuring and combining similarity, and in Section 3, give an overview of our previously proposed generalisation. Then in Section 4, we give a partial reconciliation of the two, showing how a significant feature of the novel similarity metric framework can be retained, in a manner consistent with the behaviour of the more conventional method. In Section 5, we describe a modified elicitation technique to capture the information needed. Section 6 describes some implementation details, and in Section 7 we conclude.

2. Similarity Measures and Weighted sum

Customarily, similarity in case-based reasoning systems is denoted by a numeric value, often one normalised to the interval [0,1], as we will consider here. (Sometimes an unnormalised numeric value is used, and hence a possibly wider numeric range; the discussion in this paper is largely applicable in that context too.) Thus a *similarity measure*, used to compare the similarity of two cases, has a type such as:

$$m :: (case, case) \rightarrow [0,1]$$

For example, suppose cases possess some attribute *price*, with domain £0 to £10,000. Then we might wish to define similarity by negated absolute difference, normalised to the unit interval, thus:

$$m_{price}\ (c_1, c_2) = 1 - \mathbf{abs}\ (price(c_1) - price(c_2))/10000$$

where we use *price* here as a selector function, returning the attribute of interest from the whole case. Or for example, given an attribute *facilities* whose value is a feature-set, where the maximum possible number of features per case is n, then we might define:

$$m_{facilities}\ (c_1, c_2) = \mid facilities(c_1) \cap facilities(c_2) \mid / n$$

Then for example, if two cases have three features out of a possible five in common, their similarity value will be 0.6.

In most cases, in order to make an overall judgement of total similarity, we need to take into account more than one aspect of similarity – for example, as in the above case, both *price* and *facilities* may be of interest. The usual methodology for doing so is *weighted sum*: the similarity values obtained by applying each similarity measure, to each of two cases, are first multiplied by some weight, and are then summed together, to form an aggregate similarity value (see for example Kolodner [Ko93]). For example, if we attach weight w_{price} to the *price* measure, and $w_{facilities}$ to the *facilities* measure, then we can define a weighted composite of the two, $m_{overall}$ by:

$$m_{overall}(c_1, c_2) = (w_{price}.m_{price}(c_1, c_2) + w_{facilities}.m_{facilities}(c_1, c_2)) / (w_{price} + w_{facilities})$$

Note that we re-normalise by division by the sum of the constituent weights, ensuring the result remains in the desired range. This generalises straightforwardly to a weighted sum of any number of component similarity measures, each defined on some attribute or aspect of the cases. If we have k similarity measures ($s_1 \ldots s_k$), then we associate with them a vector of k weights, ws = $w_1 \ldots w_k$, and compute the overall similarity measure as:

$$m_{ws}\ (c_1, c_2) = (\Sigma_{i=1..k}\ w_i\ .\ s_i\ (c_1, c_2)) / \Sigma_{i=1..k}\ w_i$$

A number of criticisms can be made of this means of combining different aspects of similarity. Firstly, as we make explicit in the next section, it is restricted to numeric-valued (and

hence necessarily total-ordered) similarity values. Thus for example feature-set-valued similarity measures [MJ97], the use of first-order terms to represent similarity [Pl95], and other partially ordered representations in the framework of Osborne, Bridge, and Ferguson[OB97, FB99] cannot be combined using this method.

Secondly, values for each of the weights must be obtained, which can be problematic as these do not have a clear, intuitive meaning. If not obtained by direct interrogation of the end user, they must be supplied by some combination of the knowledge engineer, the interface and the users of the CBR system. Nor is the effect of a given weight immediately evident, in terms of the best cases that will be retrieved. Relatedly, if the numeric value of weighted sum computation is presented to the end user, by way of justification and quantification of the results, then the interpretation of this value (or indeed any graphical or linguistic representation derived from it) is far from straightforward.

However, the limitation we wish to concentrate on here, particularly in the context of interactive case retrieval based systems, is that which we elsewhere [FB00a] term 'spurious precision'. That is, because weighted-sum based similarity measures are necessarily totally ordered (on the real number line), all cases are strictly ranked, one above the other (apart from exact ties, caused by cases identical on all attributes considered, or by arithmetic co-incidence). This ordering is then used to produce the single best match, or often the best *n* matches, regardless of whether these distinctions are large, or very fine. This is not necessarily cognitively reasonable, however, as the degree of accuracy with which the weights were obtained does not necessarily correspond to the precision with which we can discriminate between cases by their computed similarity values. This is especially so when the net similarity measure is arrived at by weighted sum, due to the difficulties in obtaining weight values with any high degree of accuracy and genuine precision.

The end user of the system is, accordingly, offered no help as to whether there really is a single 'clear winner', or if there are 'near ties', meriting closer scrutiny by the user herself before deciding which is genuinely the best match, on the basis of mature consideration. Nor is the absolute difference in total similarity values of any immediate use here – the difficulties we mention in meaningful presentation of these numeric values still apply. Nor is it clear how this might be automated, for essentially the same reason: the magnitude of these numbers, and of differences between them, can only be interpreted with knowledge of the similarity measures themselves, the effects of the chosen weights, and of the 'sensitivity' of the user to those differences.

In Section 4, we return to the problem of making determinations of this sort; in the following section, we examine a different proposed remedy for this, and the previously made criticisms of weighted sum.

3. Similarity Metrics and Combining forms

One prescription for the various difficulties with similarity and weighting is the framework of *similarity metrics*, first proposed by Osborne and Bridge [OB97], presented here in terms of Ferguson and Bridge's modified treatment [FB99]. A similarity metric is used just as is a similarity measure, and represents a strict generalisation in terms of its signature:

$$m :: (case, case) \rightarrow S$$

where S is the set of values used to represent the degrees of similarity between the cases. S may be a numeric set, as before, or its members may themselves be sets; for example:

$$m'_{facilities}(c_1, c_2) = facilities(c_1) \cap facilities(c_2)$$

Here, rather than as before measuring the size of the intersection of two feature sets numerically, we use *the intersection itself*, as a set-valued result, to represent the degree of similarity. Or as per Plaza [Pl95], the members of the similarity result set may be first-order terms. The key requirement we make on this set S is that it have some *partial order* defined on it, relating the elements in such a way that we can tell whether or not a given degree of similarity is or is not greater than some other. For numeric-valued metrics, equivalent to traditional similarity measures, we use the normal number line ordering, <. When degrees of similarity are represented by sets (as in $m'_{facilities}$, above), we can order the similarity values using set inclusion: for example, suppose the features with which we are concerned are drawn from the set {ski, swim, surf}. That is, in the usual notation, we define the partial order ($\sqsubseteq$) between two such similarity values x and y by:

$$x \sqsubseteq y \triangleq x \subseteq y$$

The following diagram, Figure 1, shows the points of the similarity type, each of which is a subset of the feature set domain, ordered as above.

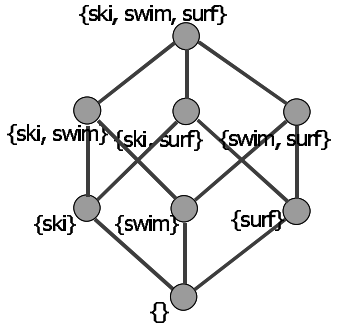

Figure 1. Feature sets as similarity values.

Notice that for example, the set {ski, swim} is greater than the set {ski}: two cases which resemble each other to the former extent are more similar than two which do so only to the latter.

However, notice that the point {swim, surf} is *not* greater than {ski} (nor vice versa). These points are not related by the partial order: they are said to be *incomparable*. The significance of this in case retrieval is that when we have two cases, with incomparable degrees of similarity to the probe case, then we might regard the relative merit of those two cases as being unspecified: the system has not been provided with enough information in order to make a discrimination at that level of detail, and so both must be presented to the user as (potentially) 'equally good' matches. (Or in a context of non-interactive case retrieval, these incomparably similar cases, must all be preserved for consideration in the next stage, e.g. attempted case adaptation.) Thus in contrast to the traditional measures plus weighting, we can say that rather than spurious precision, such metrics exhibit *quantified uncertainty*: rather than attempting to make discrimination beyond the known accuracy and quality of the input data, we represent the inability to do so (or rather, the lack of wisdom of doing so) explicitly via these incomparable points.

As regards the other drawbacks of traditional metrics, we offer the following brief summary. It is immediately clear that metrics offer greater generality; in fact, by making only the minimal assumption that the results be partially ordered, it is general enough to include every proposed type of function proposed to measure similarity, and in a sense is the most general possible, without an entirely different approach. On elicitation and explanation, we have allowed the user's perception of similarity (or anticipated perception, where a knowledge engineer performs this work) to be specified differently than with numeric weights. This has the benefit that this elicitation need not necessarily be done at the level of numeric values, and is consequently less onerous in having to be less 'exact'; and that we avoid inaccuracies introduced, either by a knowledge engineer anticipating distinctions that do not exist in the user's mind, or by forcing such choices prematurely from the user. These problems, and a proposed remedy in terms of similarity metrics, are discussed in more detail in our earlier work.

As we discuss below, we have also the benefit in terms of eliminating the need for eliciting weights, when we choose a different means of combining metrics.

However, we have also introduced a fresh difficulty. Although strictly a generalisation in terms of type, it is no longer possible to use weighted sum to combine metrics, as they are no longer necessarily of any numeric type. Instead, we can use any of a number of different *combining forms* to do this in the general case: the standard order-theoretic notions of product, which makes *no* discrimination between the relative importance of the two component similarities, and strict prioritisation, which makes one absolutely more significant than the other [FB99]. We have also proposed a novel combinator, *generalised prioritisation*, which covers a continuum of effect between product and strict prioritisation, relying on elicitation of *indifference relations* rather than weights.

However, these techniques all lead to a further difficulty: they are each different in effect from weighted sum as a combinator. Where each component metric is numeric-valued, and thus we could choose to use any of the above metric combinators, *or* the weighted sum method, the two are behaviourally quite different. This is unfortunate in two respects: firstly, it introduces a discontinuity of effect for purely technical reasons, having nothing to do with the user's knowledge or intent in interacting with the system – if one 'revises' a system from one framework to the other, its behaviour is significantly altered. Secondly, none of the combinators mentioned replicate the effect obtained by weighting of 'pro-rating' each aspect; where a smaller or larger similarity in one respect can compensate for variations in another, linearly. That is, a low similarity value in one component, but with a high weight, may count for more than a higher similarity in a lower-weighted attribute. Whether this is always an appropriate semantic effect is arguable from domain to domain, but it seems likely to remain desirable in at least some application domains.

This discontinuity leads us to consider whether a reconciliation is possible. Can we retain the effect of weighting, at least on those similarity types for which that is applicable, while retaining some of the benefits of the more general framework – in particular, capturing uncertainty?

4. Weight Intervals

Our idea is to use *intervals*, for both the weight parameters of the combining form, and to represent the similarity values themselves. These correspond, conceptually, to estimated uncertainty in the inputs (the relative importances of each similarity component), and to computed uncertainty of the overall similarity result.

We therefore for each weighted combination specify a vector of intervals (thus, in effect, a pair of weight vectors), $[w_1, w'_1], \ldots, [w_k, w'_k]$, where each lower and upper point of each interval is, as before, a real number in the range $0 \leq w_i \leq 1$. We normalise the resultant similarity interval by dividing both lower and upper intervals by the sum of the upper weights, obtaining:

$$m_{ws}(c_1, c_2) = [\Sigma_{i=1..k}\, w_i \,.\, s_i(c_1, c_2) / t,\ \Sigma_{i=1..k}\, w'_i \,.\, s_i(c_1, c_2) / t]$$

where $t = \Sigma_{i=1..k}\, w'_i$.

Note the following special cases: if the weights in every pair are equal ($w_i = w'_i$), we recover behaviour identical to that of single weights, since then the computed interval is then also a single point. Conversely, if every weight interval is set equal to [0,1], this indicates no knowledge whatsoever about the relative importance of each component.

We use the following partial order on similarity intervals. The idea is that one interval should be placed above another in the similarity ordering only when it is *entirely* above the other; that is, every point in the first interval is greater than every point in the second. If the two overlap, then as

there are points in each set that are greater than some point in in the other, then such intervals are judged to be incomparable. Thus:

$$[x, x'] \sqsubset [y, y'] \triangleq x' < y$$

So for example, the interval [0.55, 0.65] is less than the interval [0.7, 0.75], but is *not* less than the interval [0.6, 0.9]; they are incomparable.

As previously noted, this represents a direct extension only of numeric-valued measures, in contrast to the intended generality of the similarity metric framework. However, a further extension is possible: as set out in earlier work [FB00b], we can achieve a weighting-like affect in non-numeric domains by way of suitable conversion functions. As we argue there, rather than converting single similarity values directly to real numbers in isolation, more accurate results are obtained by means of *difference functions*, where we compare compound similarity values componentwise, applying a single difference function to each component pair of values to be compared. Weighting each component and a final comparison are then performed as conceptually separate steps. We can extend the same technique to use weight intervals; the difference functions play the same role as before, and in the weighting step we introduce intervals to capture uncertainty, as we describe in this paper. This permits the further modification of the difference functions themselves to return intervals, where these are also seen as a source of inaccuracy. We do not currently take this approach, and each such interval construction would necessarily be done done in a domain-specific manner.

5. Elicitation

Our weight intervals construct does not directly help with the issue of obtaining meaningful values for weights from users. Indeed, in a sense it may appear that we have worsened the problem: rather than a single weight vector, it seems we must now obtain two. As this is clearly not desirable, we propose not to elicit intervals directly, but rather as a *median* value, and a *confidence*. The first of these plays exactly the same role as the single weight that would be used in a traditional weighted-sum system. It can be elicited by any of the means that have been proposed to aid this process: via a GUI, or by means of linguistic hedges.

The confidence value indicates the degree of specificity the user (or knowledge engineer) wishes to attach to the selected median weight value; a maximal confidence value indicates complete certainty about the chosen weight (say, w), corresponding exactly to the conventional weighting scheme (that is, to the single-point interval [w, w]). A minimal confidence value indicates *complete* uncertainty about the appropriate value for the weight, corresponding to the entire interval [0,1].

We can relate these two notions as follows. Given a median weight *w*, and a confidence value *c*, we compute from this a weight interval $[w_1, w_2]$, where the computed weights w_1 and w_2 are defined in order to produce an interval centred on *w*, of width proportional to (1-*c*), by:

$$w_1 = w.c, \; w_2 = w.(2-c)$$

For example, w = 0.7, c = 0.8 translates to the weight interval [0.56, 0.84]. Notice that we use multiplicative factors to capture uncertainty, rather than additive terms, due to the way we normalise to the unit interval. As the intervals constructed by the technique might exceed the bounds of [0,1], we indeed need to normalise at this step; having constructed the vector of intervals, $[w_1, w'_1], \ldots, [w_k, w'_k]$, if any of the upper bounds w'_i exceed 1, we should divide all weights by the maximum such, thus:

$$[w_1 / w^m, w'_1 / w^m], \ldots, [w_k / w^m, w'_k / w^m]$$

where $w^m = \max \{w'_1, \ldots, w'_k\}$. Note this is a reasonable thing to do as the significance of the weights is not their absolute but their relative values; this rescaling has no effect, once the weighted sum comparison rule is applied.

The additional problem remains of eliciting the confidence itself. A numeric value may be obtained directly, or as with weights, by graphical means or linguistic hedges.

6. Implementation

An experimental implementation has been constructed, using the 'Holidays' case base (where the focus is on interactive case retrieval, and no adaptation is performed). Evaluation of the relative cognitive merits of conventional weights, general metrics, and interval weights is being carried out, using each of these different means of querying the same case base.

The weight intervals scheme is implemented on top of the system we outline in earlier work [FB99], which handles similarity metrics in the general case; as intervals form a partial order, we can use the mechanism of traversing the case base, maintaining a set of maximal similarity elements of the cases seen so far, and comparing each fresh case against each of that maximal set. This introduces a certain overhead compared to the simple similarity measure case, where only a single maximum needs to be maintained, especially when the partial order used is a very 'flat' one, i.e. containing large numbers of incomparable elements, and hence a potentially large set of maxima.

This can be compared to finding the most similar cases using traditional measures; given a probe *p*, a case base *C*, and a similarity metric m, we can find the maximal similarity value b, and the corresponding cases, B, as shown in Algorithm 1. This has complexity class $O(n)$, where n is the size of case base, if we take the similarity measure calculation to be of constant time.

```
b ← 0;
B ← {};
for c ∈ C do
  s ← m(p,c);
  if s > b then
    b ← s;
    B ← {c};
  else if s = b then
    B ← B ∪ {c};
```

Algorithm 1: finding maximally similar cases using numeric measures

Compare this with the computation necessary in the general case for similarity metrics, as shown in Algorithm 2. Here the best similarity values form a *set*, b, which must be searched for each fresh case considered to maintain the correct current maxima. Therefore in the worst case, this can be of complexity $O(ln)$, where l is the cardinality of the set of maxima.

```
b ← {};
B ← {};
for c ∈ C do
  s ← m(p,c);
  if ∄x ∈ b : s ⊏ x then
    b ← {s} ∪ {x : x ∈ b, not (x ⊏ s)};
    B ← {c} ∪ {d : d∈B, m(p,d) ∈ b};
```

Algorithm 2: finding maxima using general metrics

However, we can improve on this situation, in the particular case of intervals. Rather than a set of weighted-sum intervals, and a set of corresponding cases, we can maintain just a single consolidated interval representing all the maxima, and a 'shortlist' of cases which have appeared in any of the working maxima.

```
[b1, b2] ← [0, 0];
B ← {};
for c ∈ C do
    [s1, s2] ← m(p,c);
    if s1 > b2 then
        B ← {c};
    else if s2 > b1 then
        B ← B ∪ {c};
    [b1, b2] ← [max (b1, s1), max (b2, s2)]
B ← {c : c ∈ B, [s1, s2] = m(p,c), s2 ≥ b1}
```

Algorithm 3: finding maxima using interval-valued metrics

The modified calculation using intervals is described in Algorithm 3. We maintain a current maximum interval $b = [b_1, b_2]$, and shortlist of cases B, and we compute the similarity of the next case c, which we take to be the interval $s = [s_1, s_2]$. If s is strictly better than b, we reset the shortlist B to be $\{c\}$, only: this new case is evidently more similar to the probe than any cases seen

previously. If *s* and *b* overlap, we add the current case to the shortlist: this case has similarity incomparable (or equal) to some earlier case. Otherwise, we do nothing, as the case must have been less similar than some other. We then update the maximal interval to the (possibly) new value found by taking the maxima of the two lower extents, and the two upper extents, respectively, of *s* and *b*. After a complete traversal of the case base, the shortlist *B* is then pruned by comparison with the final maximal interval *b*, to obtain the set of truly maximal similarity cases for the whole case base, eliminating those shortlisted cases which are worse than *b*. This represents only a constant factor extra cost over simple weighting, and thus remains of complexity *O(n)*.

7. Conclusions

In this paper, as elsewhere, we argue that there are certain significant limitations in the conventional techniques for constructing, and for combining, similarity measures. What is novel here is that we have attempted to find further commonality between the customary approach, and our previous prescription. By means of our weighting-compatible treatment of uncertainty in this paper, we have provided something of a halfway house between traditional similarity measures and our general similarity metrics, essentially compatible with both. It remains to make a more systematic and empirical evaluation of the relative merits of each of these treatments, in order to determine under what circumstances each of these offers the greatest utility.

References

[FB99] A. B. Ferguson and D. G. Bridge. Generalised Prioritisation: A New Way of Combining Similarity Metrics. *Procs. of Tenth Irish Conference on Artificial Intelligence & Cognitive Science*, pp.137-142, 1999, Derek Bridge, Ruth Byrne, Barry O'Sullivan, Steven Prestwich and Humphrey Sorensen (eds.).

[FB00a] A. B. Ferguson and D. G. Bridge. Indifference Relations: Being Purposefully Vague in Case-Based Retrieval. *Advances in Case-Based Reasoning (Procs. of the 5th European Workshop)*, E.Blanzieri and L.Portinale (eds.), Lecture Notes in Artificial Intelligence 1898, pp.74-85, Springer, 2000.

[FB00b] A. B. Ferguson and D. G. Bridge. Generalised Weighting: A generic combining form for similarity metrics. *Procs. of Eleventh Irish Conference on Artificial Intelligence & Cognitive Science*, J. Griffith and C. O'Riordan (eds.), pp.169-179, 2000

[Ko93] J. L. Kolodner, (1993). *Case Based Reasoning*. Morgan Kaufmann, 1993.

[MJ97] D. Matuschek and K. Jantke. Axiomatic characterizations of structural similarity for case-based reasoning. *FLAIRS-97, Proc. Florida AI Research Symosium*, Daytona Beach, FL, USA, May 11 – 14, 1997.

[OB97] H. R. Osborne and D. G. Bridge: Similarity Metrics: A Formal Unification of Cardinal and Non-Cardinal Similarity Measures, D. B. Leake and E. Plaza (eds.), *Case-Based Reasoning Research and Development* (Procs. of the Second International Conference on Case-Based Reasoning), Lecture Notes in Artificial Intelligence 1266, pp.235-244, Springer, 1997

[Pl95] E. Plaza: Cases as terms: A feature term approach to the structured representation of cases, in M.Veloso & A.Aamodt (eds.), *Case-Based Reasoning Research and Development* (Procs. of 1st International Conference on Case-Based Reasoning), LNAI-1010, pp.265-276, Springer, 1995.