A Taxonomy of Similarity Mechanisms for Case-Based Reasoning

Pádraig Cunningham

Abstract—Assessing the similarity between cases is a key aspect of the retrieval phase in case-based reasoning (CBR). In most CBR work, similarity is assessed based on feature value descriptions of cases using similarity metrics, which use these feature values. In fact, it might be said that this notion of a feature value representation is a defining part of the CBR worldview—it underpins the idea of a problem space with cases located relative to each other in this space. Recently, a variety of similarity mechanisms have emerged that are not founded on this feature space idea. Some of these new similarity mechanisms have emerged in CBR research and some have arisen in other areas of data analysis. In fact, research on kernel-based learning is a rich source of novel similarity representations because of the emphasis on encoding domain knowledge in the kernel function. In this paper, we present a taxonomy that organizes these new similarity mechanisms and more established similarity mechanisms in a coherent framework.

Index Terms—Machine learning, case-based reasoning, nearest neighbor classifiers.

1 Introduction

Similarity is central to CBR because case retrieval depends on it. The standard methodology in CBR is to represent a case as a feature vector and then to assess similarity based on this feature vector representation (see Fig. 6a). This methodology shapes the CBR paradigm; it means that problem spaces are visualized as vector spaces, it informs the notion of a decision surface and how we conceptualize noisy cases, and motivates feature extraction and dimension reduction, which are key processes in CBR systems development. It allows knowledge to be brought to bear on the case retrieval process through the selection of appropriate features and the design of insightful similarity measures.

However, similarity based on a feature vector representation of cases is only one of a number of strategies for capturing similarity. From the earliest days of CBR research, more complex case representations requiring more sophisticated similarity mechanisms have been investigated. For instance, cases with internal structure require a similarity mechanism that considers structure [1], [2]. More recently, a number of novel mechanisms have emerged that introduce interesting alternative perspectives on similarity. The objective of this paper is to review these novel mechanisms and present a taxonomy of similarity mechanisms that places these new techniques in the context of established CBR techniques.

Some of these novel similarity strategies have already been presented in CBR research (e.g., compression-based similarity in [3] and edit distance in [4], [5]), and others come from other areas of machine learning (ML) research. The taxonomy proposed here organizes similarity strategies into four categories (see also Fig. 1) as follows:

- direct mechanisms,
- transformation-based mechanisms,
- information-theoretic measures, and
- emergent measures arising from an in-depth analysis of the data.

The first category covers direct mechanisms that can be applied to feature vector representations and is the dominant well-established strategy. Examples of transformation-based approaches to similarity have been around for some time (e.g., graph edit distance, [1]), however, there have been a number of recent developments in this area as the resources for what are normally computationally expensive techniques have come available. The last two categories cover some exciting new directions in similarity research and placing these in the context of other approaches to similarity is the main contribution of this paper.

Strategies for similarity cannot be considered in isolation from the question of representation. For this reason, a brief review of representation strategies in CBR is provided in Section 2 before we embark on an analysis of similarity in later sections. The review of similarity techniques begins in Section 3, where the direct feature-based approach to similarity is discussed. Then transformation-based techniques are described in Section 4. Novel strategies based on ideas from information theory are discussed in Section 5. In Section 6, some novel techniques that we categorize as *emergent* are reviewed—similarity derived from random forests and cluster kernels are covered in this section. Then, in Section 7, we review the implications of these new similarity strategies for CBR research. The paper concludes with a summary and suggestions for future research in Section 8.

2 Representation

For the purpose of the analysis presented in this paper, we can categorize case representations as follows:

- feature value representations,
- structural representations, and
- sequences and strings.

- *The author is with the School of Computer Science and Informatics, Complex and Adaptive Systems Laboratory, University College Dublin, Belfield, Dublin 4, Ireland. E-mail: padraig.cunningham@ucd.ie.*

Manuscript received 4 Jan. 2008; revised 18 Aug. 2008; accepted 23 Oct. 2008; published online 5 Nov. 2008.
Recommended for acceptance by C. Clifton.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2008-01-0007.
Digital Object Identifier no. 10.1109/TKDE.2008.227.

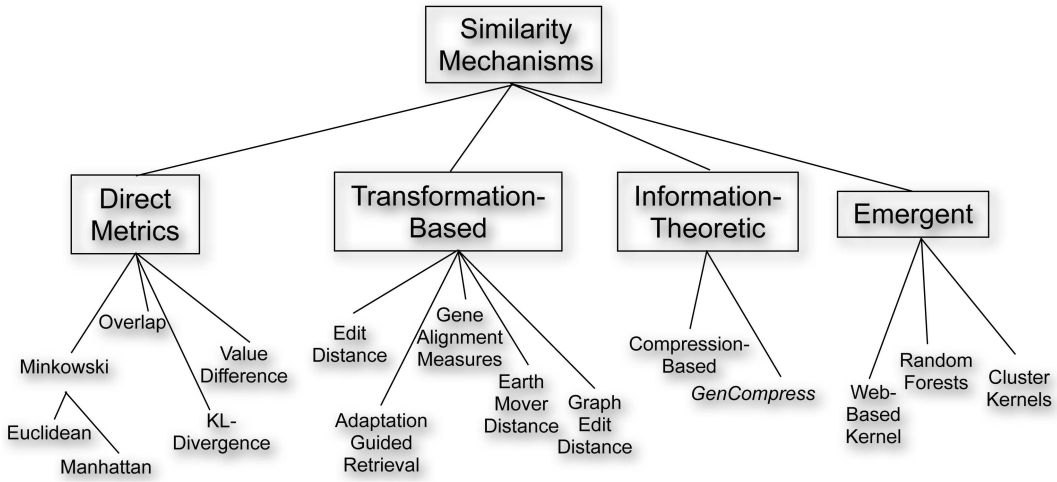


Fig. 1. A taxonomy of similarity mechanisms.

In the remainder of this section, we summarize each of these categories of representation.

2.1 Feature Value Representations

The most straightforward case representation scenario is to have a case-base D made up of $(\mathbf{x}_i)_{i\in[1,|D|]}$ training samples with these examples, which are described by a set of features F with numeric features normalized to the range [0, 1]. In this representation, each case $(\mathbf{x}_i)$ is described by a feature vector $(\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{i|F|})$.

2.1.1 Internal versus External

An important distinction in philosophy concerning concepts is the difference between intensional and extensional descriptions (definitions). An intensional description describes a concept in terms of its attributes (e.g., a bird has wings, feathers, a beak, etc.). An extensional description defines a concept in terms of instances, e.g., examples of countries are Belgium, Italy, Peru, etc. In the context of data mining, we can consider a further type of description, which we might call an *external* description. In a movie recommender system, an intensional description of a movie might be as described in Table 1 whereas an external "description" of the movie might be as presented in Table 2. The data available to a collaborative recommender system are of the type shown in Table 2, it can produce a recommendation for a movie for a user based on how other users have rated that movie. We can think of the data available to the collaborative recommender as an external feature value description whereas a content-based recommender would use an intensional (or internal) feature value representation.

2.1.2 Representation Enhancement

A recent trend in CBR, and in ML in general, has been to extend or enhance the case representation derived from the case. Wiratunga et al. have shown how techniques from association rule mining can be used to impute new features for textual cases [6]. They use the a priori association rule learner to discover word associations in a text corpus and then use these associations to add terms to the representation of a text. They show that this enhanced representation improves classification accuracy.

Whereas the work described by Wiratunga et al. extends the representation of individual cases by mining the training corpus, Gabrilovich and Markovitch present a strategy for representation enhancement based on Web mining [7]. They employ an *auxiliary* classifier to associate texts to be classified with papers in Wikipedia. The representation of the texts is augmented with features denoting the concepts associated with these Wikipedia papers. They show that this augmented representation improves the classification performance.

These two approaches to representation enhancement are similar in that they allow texts to be represented by more features. In the first scenario, this is done by mining the training corpus to discover new associations, and in the

TABLE 1
An Example of an Internal (or Intensional) Case Representation—Case Attributes Describe Aspects that Would be Considered Internal to the Case

Case D	
Title:	Four Weddings and a Funeral
Year:	1994
Genre:	Comedy, Romance
Director:	Mike Newell
Starring:	Hugh Grant, Andie MacDowell
Runtime:	116
Country:	UK
Language:	English
Certification:	USA:R (UK:15)

TABLE 2
An Example of a Set of Cases Described by *External* Features

	A	**B**	**C**	**D**	**E**	**F**	**G**
User 1	0.6	0.6	0.8			0.8	0.5
User 2		0.8	0.8	0.3	0.7		
User 3	0.6	0.6	0.3	0.5		0.7	0.5
User 4					0.7	0.8	0.7
User 5	0.6	0.6	0.8			0.7	
User 6		0.8	0.8	0.7	0.7		
User 7	0.7	0.5			0.7		
User 8					0.7	0.7	0.8

In this example, the movie case D is described by three external features, the ratings given to it by users 2, 3, and 6.

second scenario, this is done by looking outward to Wikipedia to discover new associations. In either scenario, the end result is still a feature value representation.

2.2 Structured Representations

Practical experience in the development of CBR systems has demonstrated that a simple feature vector is not adequate to represent the complexity of cases encountered in practice: it is often the case that cases have internal structure that needs to be represented [8], [1], [9], [10], [11]. Given that research on structural case representations is driven by the requirements of practical applications, there is considerable variety in the types of structured representation that has been covered in the literature. However, the following three categories cover the alternatives fairly well:

- The most straightforward structure (beyond a flat feature vector) is a **hierarchical structure** whereby the features value themselves reference nonatomic objects. For instance, in case, $\mathbf{x}_i = (\mathbf{x}_{i1}, \ldots, \mathbf{x}_{if}, \ldots, \mathbf{x}_{i|F|})$, the feature $\mathbf{x}_{if}$ could reference a case structure. This simple extension allows for the description of cases with a complex hierarchical structure [8], [10], [12], [13].
- The next most general structure is a **network structure** (typically a semantic network): there is a long history of semantic network-based case representations in CBR and in the related research area of analogical reasoning [2], [14], [11]. Whereas in a hierarchical representation, there is only one link type, the part-of link; in a network structure, there can be many link types with a rich link semantics.
- The final type of structured representation we distinguish here is the **flow structure**. These can be cases that represent plans (see Veloso and Carbonell [15]) or workflows as described by Minor et al. [9]. From a structural perspective, these flow representations share many of the characteristics of hierarchical and network representations, however, since they model an activity, they also have a temporal dimension.

These three categories describe situations where cases have an internal structure that is more elaborate than a simple feature vector representation. The other way in which case representations can have structure is when the case vocabulary itself (i.e., the attributes) has a structured representation—for instance, if the attribute values are organized into a taxonomy. This scenario is described very well by Bergmann and Stahl [8], Bergmann [16]. When the network structure described above is combined with such a rich attribute vocabulary, then we have an object-oriented case representation [8], [16].

2.3 String and Sequence Representations

In many circumstances, such as helpdesk applications, for instance, experiences will be recorded in free text [17], [18]. An obvious example highlighted by Bergmann is a Frequently Asked Questions list [16]. Such a free text representation greatly influences the strategies for case similarity that can be employed. The most straightforward representation for free text that supports similarity assessment is the bag-of-words strategy from information retrieval.

TABLE 3
An Example of a Bag-of-Words Representation of Three Simple Sentences

	Easiest	Online	School	Earth	Info.	Computer	Science	Graduate	Board	Meeting	Please	Find	Attached	Agenda
1	×	×	×	×										
2			×		×	×	×	×	×	×				
3			×					×	×	×	×	×	×	×

The bag-of-words approach can be illustrated with the following example. Consider three sentences representing three email messages in a spam filtering scenario as follows:

1. The easiest online school on earth.
2. Here is the information from Computer Science for the next Graduate School Board meeting.
3. Please find attached the agenda for the Graduate School Board meeting.

The first sentence is from a spam message and the other two are from legitimate e-mails. The bag-of-words representation of these sentences is shown in Table 3. With stop words removed (the, on, here, is, etc.), the sentences are represented as vectors in a 14-dimensional space. Thus, these strings can be converted to a vector space representation and direct similarity methods can be applied (see Section 3). In Section 4, we examine transformation-based approaches to string matching, and in Section 5, we consider information-theoretic measures.

3 Direct Similarity Mechanisms

The fundamental idea in CBR is problem solving based on the retrieval of similar cases, in its simplest form, this is nearest neighbor classification. The intuition underlying this is quite straightforward, examples are classified based on the class of their nearest neighbors. It is often useful to take more than one neighbor into account, so the technique is more commonly referred to as k-nearest neighbor (k-NN) classification, where the k-nearest neighbors are used in determining the class. In most circumstances, cases are represented as feature vectors and similarity is assessed *directly* from these features.

The basic idea is as shown in Fig. 2, which depicts a 3-nearest neighbor classifier on a two-class problem in a two-dimensional feature space. In this example, the decision for q_1 is straightforward—all three of its nearest neighbors are of class O, so it is classified as an O. The situation for q_2 is slightly complicated at it has two neighbors: one of class X and another of class O. This can be resolved by simple majority voting or distance weighted voting (see below).

So k-NN classification has two stages: the first is the determination of the nearest neighbors and the second is the determination of the class using those neighbors.

Let us assume that we have a training data set D made up of $(\mathbf{x}_i)_{i \in [1,|D|]}$ training samples. The examples are described by a set of features F and any numeric features have been normalized to the range [0, 1]. Each training example is labeled with a class label $y \in Y$. Our objective is to classify an unknown example $\mathbf{q}$. For each $\mathbf{x}_i \in D$, we can

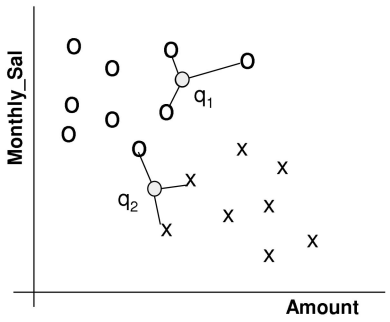


Fig. 2. A simple example of 3-nearest neighbor classification.

calculate the distance between $\mathbf{q}$ and $\mathbf{x}_i$ as follows (w_f is the weight assigned to feature f):

$$d(\mathbf{q}, \mathbf{x}_i) = \sum_{f \in F} w_f \delta(\mathbf{q}_f, \mathbf{x}_{if}). \quad (1)$$

This distance metric has the merit that it allows knowledge to be brought to bear on the assessment of similarity. The set of features F can be chosen to reflect the important features in the domain and knowledge can also be encoded in the design of the δ function. A basic version for continuous and discrete attributes would be:

$$\delta(\mathbf{q}_f, \mathbf{x}_{if}) = \begin{cases} 0, & f \text{ discrete } \& \ \mathbf{q}_f = \mathbf{x}_{if}, \\ 1, & f \text{ discrete } \& \ \mathbf{q}_f \neq \mathbf{x}_{if}, \\ |\mathbf{q}_f - \mathbf{x}_{if}|, & f \text{ continuous.} \end{cases} \quad (2)$$

While this metric handles continuous attributes reasonably well, it does a poor job with discrete attributes. Nonmatching discrete attributes contribute maximally to the distance while matching attributes don't contribute at all. Stanfill and Waltz [19] proposed the value difference metric (VDM) to address this issue. The VDM uses class conditional probabilities for discrete attribute values to refine the contribution of these attributes to the distance calculation. As with the basic metric defined in (1), the VDM is based on a weighted sum across feature values:

$$vdm(\mathbf{q}, \mathbf{x}_i) = \sum_{f \in F} w(\mathbf{q}_f) \delta(\mathbf{q}_f, \mathbf{x}_{if}), \quad (3)$$

but with the weight dependent on the feature value in the query $\mathbf{q}$. The distance contribution for each feature is calculated as follows:

$$\delta(\mathbf{q}_f, \mathbf{x}_{if}) = \sum_{y \in Y} \left(P(y|\mathbf{q}_f) - P(y|\mathbf{x}_{if}) \right)^2, \quad (4)$$

where $P(y|\mathbf{q}_f)$ is the conditional probability of class label y given feature value $\mathbf{q}_f$, i.e., the proportion of instances with attribute value $\mathbf{q}_f$ that are of class y. The weight is calculated as follows:

$$w(\mathbf{q}_f) = \sqrt{\sum_{y \in Y} P(y|\mathbf{q}_f)^2}. \quad (5)$$

This weight will be high for attribute values that are discriminating between the class labels. The principle underlying the VDM is that the difference between two attribute values is based on the class conditional probabilities for those attribute values. There are a number of other metrics that develop on this basic principle of using conditional probabilities to refine similarity measures for continuous attributes. Two refinements that have been shown to improve classification performance are the heterogenous VDM by Wilson and Martinez [20] and the minimum risk metric proposed by Blanzieri and Ricci [21].

In order to perform nearest neighbor classification, the k-nearest neighbors are selected based on this distance metric. Then, there are a variety of ways in which the k-nearest neighbors can be used to determine the class of $\mathbf{q}$. The most straightforward approach is to assign the majority class among the nearest neighbors to the query.

It will often make sense to assign more weight to the nearer neighbors in deciding the class of the query. A fairly general technique to achieve this is distance weighted voting, where the neighbors get to vote on the class of the query case with votes weighted by the inverse of their distance to the query as follows:

$$Vote(y) = \sum_{c=1}^{k} \frac{1}{d(\mathbf{q}, \mathbf{x}_c)^n} 1(y, y_c). \quad (6)$$

Thus, the vote assigned to class y by neighbor $\mathbf{x}_c$ is 1 divided by the distance to that neighbor, i.e., $1(y, y_c)$ returns 1 if the class labels match, and 0, otherwise. In (6), the variable n would normally be 1, but values greater than 1 can be used to further reduce the influence of more distant neighbors.

3.1 Similarity and Distance Metrics

While the terms *similarity metric* and *distance metric* are often used colloquially to refer to any measure of affinity between two objects, the term *metric* has a formal meaning in mathematics. A metric must conform to the following four axioms (where $d(x, y)$ refers to the distance between two objects x and y):

1. $d(x, y) \geq 0$; nonnegativity.
2. $d(x, y) = 0$ iff $x = y$; identity.
3. $d(x, y) = d(y, x)$; symmetry.
4. $d(x, z) \leq d(x, y) + d(y, z)$; triangle inequality.

It is possible to build a k-NN classifier that incorporates an affinity measure that is not a proper metric, however, there are some performance optimizations to the basic k-NN algorithm that require the use of a proper metric [22], [23]. In brief, these techniques can identify the nearest neighbor of an object without comparing that object to every other object but the affinity measure must be a metric, in particular, it must satisfy the triangle inequality.

The basic distance metric described in (1) and (2) is a special case of the Minkowski Distance metric—in fact, it is the 1-norm (L_1) Minkowski distance. The general formula for the Minkowski distance is

$$MD_p(\mathbf{q}, \mathbf{x}_i) = \left(\sum_{f \in F} |\mathbf{q}_f - \mathbf{x}_{if}|^p \right)^{\frac{1}{p}}. \quad (7)$$

The L_1 Minkowski distance is also known as the Manhattan distance and the L_2 distance is the euclidean distance. It is unusual but not unheard of to use p values greater than 2. Larger values of p have the effect of giving greater weight to the attributes on which the objects differ most. To illustrate this, we can consider three points in

2D space; $A = (1, 1), B = (5, 1)$, and $C = (4, 4)$. Since A and B differ on one attribute only, the $MD_p(A, B)$ is 4 for all p, whereas $MD_p(A, C)$ is 6, 4.24, and 3.78 for p values of 1, 2, and 3, respectively. So C becomes the nearer neighbor to A for p values of 3 and greater.

The other important Minkowski distance is the L_∞ or Chebyshev distance.

$$MD_\infty(\mathbf{q}, \mathbf{x}_i) = \max_{f \in F} |\mathbf{q}_f - \mathbf{x}_{if}|.$$

This is simply the distance in the dimension in which the two examples are most different; it is sometimes referred to as the chessboard distance as it is the number of moves that takes a chess king to reach any square on the board.

3.2 Set-Theoretic Measures

The Minkowski distance defined in (7) is a very general metric that can be used in a k-NN classifier for any data that are represented as a feature vector. However, there has been much discussion, particularly in the area of image analysis, on the appropriateness of this geometric approach to similarity. Santini and Jain [24] provide an interesting discussion on the appropriateness of these metric axioms and some alternative axioms based on a set-theoretic perspective on similarity. An alternative to this geometric perspective on similarity is to view similarity in terms of feature overlap. This perspective on similarity has received a lot of attention in the psychological literature, especially in the literature based on the seminal work of Tversky [25]. Similarity in Tversky's theory is a function of common features and distinctive features as follows:

$$s(x, y) = f(X \cap Y) - \alpha f(X \backslash Y) - \beta f(Y \backslash X), \tag{8}$$

where X and Y are the sets of features representing objects x and y and α and β are weights associated with the distinctive features. Tversky further formalizes this in his ratio model:

$$S_T(x, y) = \frac{f(X \cap Y)}{f(X \cap Y) + \alpha f(X \backslash Y) + \beta f(Y \backslash X)}. \tag{9}$$

A practical version of this measure is the Tanimoto measure [26] that gives equal status to all components:

$$S_{Tn}(x, y) = \frac{|X \cap Y|}{|X \cap Y| + |X \backslash Y| + |Y \backslash X|}. \tag{10}$$

There are a number of similarity or distance measures based on this set-theoretic perspective. For instance, the Jaccard index [27] measures the proportion of features that two objects share:

$$S_J(x, y) = \frac{|X \cap Y|}{|X \cup Y|}. \tag{11}$$

The Dice similarity coefficient [28] is similar to the Jaccard index but gives more weight to common features:

$$S_D(x, y) = \frac{2|X \cap Y|}{|X| + |Y|}. \tag{12}$$

While these set-theoretic similarity measures are important in their own right, they are also widely used in experimental validation of assessment of reproducibility. For instance, the Jaccard index is widely used as a validation measure in clustering [29].

3.3 Kullback-Leibler Divergence and the χ^2 Statistic

When working with image data, a convenient representation for the purpose of calculating distances is a color histogram. An image can be considered as a gray-scale histogram H of N levels or bins, where h_i is the number of pixels that fall into the interval represented by bin i (this vector h is the feature vector). The Minkowski distance (7) can be used to compare two images described as histograms. L_1, L_2, and less often L_∞ norms are used.

Other popular measures for comparing histograms are the Kullback-Leibler divergence (13) [30] and the χ^2 statistic (14) [31]:

$$d_{KL}(H, K) = \sum_{i=1}^{N} h_i \log\left(\frac{h_i}{k_i}\right), \tag{13}$$

$$d_{\chi^2}(H, K) = \sum_{i=1}^{N} \frac{h_i - m_i}{h_i}, \tag{14}$$

where H and K are two histograms, h and k are the corresponding vectors of bin values, and $m_i = \frac{h_i + k_i}{2}$. Since the Kullback-Leibler divergence is a measure of the difference between two probability density functions, it can also be used for continuous data by replacing the summation in (13) with an integral.

While these measures have sound theoretical support in information theory and statistics, they have some significant drawbacks. The first drawback is that they are not metrics in that they do not satisfy the symmetry requirement. However, this problem can easily be overcome by defining a modified distance between x and y that is in some way an average of $d(x, y)$ and $d(y, x)$—see [31] for the Jeffrey divergence, which is a symmetric version of the Kullback-Leibler divergence.

A more significant drawback is that these measures are prone to errors due to bin boundaries. The distance between an image and a slightly darker version of itself can be great if pixels fall into an adjacent bin as there is no consideration of adjacency of bins in these measures.

3.4 Symbolic Attributes in Taxonomies

In Section 2.2, we mentioned that one way in which structure can be incorporated in the case representation is by organizing feature values into a taxonomy of is-a relationships [16]. In this situation, the structure of the taxonomy contains information about the similarity between two feature values. Wu and Palmer [32] proposed the following measure to quantify this:

$$\delta(\mathbf{q}_f, \mathbf{x}_{if}) = \frac{2N}{N(\mathbf{q}_f) + N(\mathbf{x}_{if}) + 2N}, \tag{15}$$

where $N(\mathbf{q}_f)$ and $N(\mathbf{x}_{if})$ are the number of edges between the corresponding feature values and their common parent $T(\mathbf{q}_f, \mathbf{x}_{if})$ in the hierarchy and N is the number of edges between this common parent node and the root of the taxonomy. This measure is interesting in that, in addition to considering the distances to the common parent node, it also considers the distance from that node to the root of the taxonomy. While this *edge counting* measure is conceptually simple and can be calculated directly, it has the potential drawback that it assigns the same status to *all* edges. It does not capture the fact that some is-a relationships are closer

than others. In Section 5, we describe an information-theoretic measure for similarity in taxonomies proposed by Resnik that overcomes this [33].

3.5 Summary

These direct mechanisms for similarity assessment represent the dominant strategy in CBR research. They have the advantage of being computationally efficient and are effective in most situations. These direct mechanisms are closely tied to the use of feature-based representations in CBR. This view of representation and similarity has had a significant impact on CBR research and has wide ranging implications. It formulates the problem space as a vector space, emphasizes the dimensionality of the data as an issue, and abstracts the similarity mechanism from the raw data.

4 Transformation-Based Measures

An alternative perspective on assessing the distance or similarity between objects is the effort required to transform one into the other. This is the principle underlying the notion of edit distance, which has quite a long history in Computer Science [34]. Edit distance has already been used in CBR research as an affinity measure [4], [5], indeed, the whole CBR field of adaptation guided retrieval is based on a view of similarity as "transformation effort" [35].

4.1 Edit Distance (Levenshtein Distance)

The edit distance (ED) is the most basic of these transformation-based measures. It counts the number of insertions, deletions, and substitutions required to transform one string to another—the ED from `cat` to `rat` is 1 and the ED from `cats` to `cat` is 1. Edit distances can be calculated efficiently using dynamic programming and the algorithm is $O(n^2)$ in time and space, where n is the string length. In terms of the subject matter of this paper, edit distance has the added importance that it can be augmented with specific knowledge about the data to produce a very knowledge-based (dis)similarity measure—an example of this is presented in the next section.

4.2 Alignment Measures for Biological Sequences

The problem of assessing similarity for biological sequences has been receiving attention for many years. There are a variety of sequence alignment problems in biology, e.g., in comparing DNA, RNA, or protein sequences. The DNA sequences `GAATCCG` and `GATTGC` might be aligned as follows [36]:

$$\texttt{G}-\texttt{AATCCG}-,$$
$$\texttt{GAT}-\texttt{T}-\texttt{G}-\texttt{C}.$$

This alignment is scored as the sum of the contributions of the alignment scores ($\texttt{G} \leftrightarrow \texttt{G}, \texttt{A} \leftrightarrow \texttt{T}, \texttt{T} \leftrightarrow \texttt{T}$, etc.) minus penalty terms for the gaps. Since substitutions are allowed in DNA alignment, the alignment scores (e.g., $\texttt{A} \leftrightarrow \texttt{T}$) are read from a substitution matrix. For any pair of strings, there will be a number of possible alignments and associated scores.

In bioinformatics, there are two alternative strategies for identifying good alignments. These are *global alignment*, which assesses how one sequence can be transformed into another using a combination of simple edits, and *local alignment*, which identifies local similarities between regions of sequences. Needleman and Wunsch [37] developed the basic global sequence alignment algorithm that works over the entire length of the sequences and is effective when the sequences are of similar length and quite similar over their entire length. Smith and Waterman [38] extended this work to identify alignments that can be substantially shorter than the full sequence. This is appropriate for comparing sequences that have only small regions of similarity. The detailed performance of both local and global alignment algorithms depends on how substitutions and gaps are penalized. Both algorithms will produce the alignment above and will only differ on longer sequences that have less in common.

The actual alignment score is the one associated with the alignment with the highest score as determined by dynamic programming. This is a transformation score that is specialized for the problem at hand with similarity knowledge encoded in this substitution matrix and in the manner the gap penalty is calculated. Alignment scores such as this embody a notion of similarity that is in tune with the way biologists view the data. For instance, in local alignment, the gap penalty can be configured to penalize the opening of gaps but not the extension of a gap once it is opened. This allows alignments, where regions separated by large gaps contribute to the alignment.

Global sequence alignment is in the spirit of similarity as used in CBR in that the objective is to identify near neighbors for the purpose of reasoning. Local sequence alignment is typically used in biology for *multiple* sequence alignment, where sequences that may be quite different are aligned to identify regions or *motifs* that are conserved across many organisms.

4.3 Earth Mover Distance

The Earth Mover Distance (EMD) is a transformation-based distance for image data. It overcomes many of the problems that arise from the arbitrariness of binning when using histograms (see Section 3.3). As the name implies, the distance is based on an assessment of the amount of effort required to convert one image to another based on the analogy of transporting *mass* from one distribution to another (see Fig. 3).

In their analysis of the EMD, Rubner et al. [31] argue that a measure based on the notion of a *signature* is better than one based on a histogram. A signature $\{\mathbf{s}_j = \mathbf{m}_j, w_{\mathbf{m}_j}\}$ is a set of j clusters, where $\mathbf{m}_j$ is a vector describing the mode of cluster j and $w_{\mathbf{m}_j}$ is the fraction of pixels falling into that cluster. Thus, a signature is a generalization of the notion of a histogram, where boundaries and the number of partitions are not set in advance; instead, j should be "appropriate" to the complexity of the image [31].

For two images described by signatures $S = \{\mathbf{m}_j, w_{\mathbf{m}_j}\}_{j=1}^{n}$ and $Q = \{\mathbf{p}_k, w_{\mathbf{p}_k}\}_{k=1}^{r}$, we are interested in the work required to transfer from one to the other for a given flow pattern $\mathbf{F}$:

$$WORK(S, Q, \mathbf{F}) = \sum_{j=1}^{n}\sum_{k=1}^{r} d_{jk} f_{jk}, \tag{16}$$

where d_{jk} is the distance between clusters $\mathbf{m}_j$ and $\mathbf{p}_k$ and f_{jk} is the flow between $\mathbf{m}_j$ and $\mathbf{p}_k$ that minimizes overall cost. An example of this in a 2D color space is shown in Fig. 4. Once the transportation problem of identifying the flow that

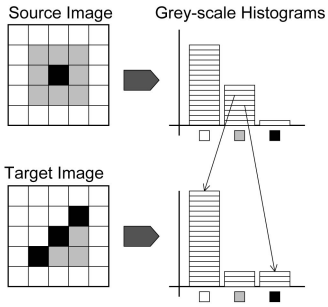


Fig. 3. An example of the EMD effort required to transform one image to another with images represented as histograms.

minimizes effort is solved (using dynamic programming), the EMD is defined to be

$$EMD(S,Q) = \frac{\sum_{j=1}^{n}\sum_{k=1}^{r} d_{jk} f_{jk}}{\sum_{j=1}^{n}\sum_{k=1}^{r} f_{jk}}. \quad (17)$$

Efficient algorithms for the EMD are described in [31], however, this measure is expensive to compute with cost increasing more than linearly with the number of clusters. Nevertheless, it is an effective measure for capturing similarity between images.

4.4 Similarity for Networks and Graphs

When a case representation has some structure, an effective similarity measure will need to take account of that structure in assessing similarity. This has long been a research topic in CBR (e.g., [1]); indeed, it predates CBR as it has been an active research area in analogy and cognitive science since the 1980s [39], [2], [14].

Perhaps the most famous measure for similarity from research on analogy is the structure mapping engine (SME) [39], [2]. One of the classic SME examples is shown in Fig. 5 [2]. This illustrates the analogy between water flow from a beaker to a vial connected by a tube and heat flow between a cup of coffee and an ice cube connected by a silver bar. As the name suggests, the key step in SME is to identify the appropriate mapping between the two domains. There are two parts to this, the identification of corresponding entities and predicates in the two domains and the construction of a maximal mapping between the structures in the two domains. It is well known now that this problem is NP-hard [40], so practical solutions will need to constrain the search in some way or use a greedy search strategy.

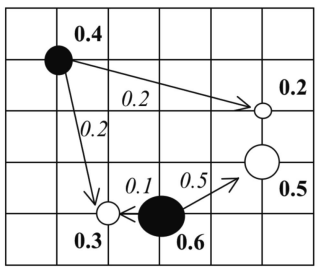


Fig. 4. An example of the EMD effort required to transform one image to another with images represented as signatures: the source image is represented by two clusters (black circles) and the target image by three clusters (white circles). The numbers in bold indicate the portion of the image represented by each cluster and the numbers on the edges show the optimum flow.

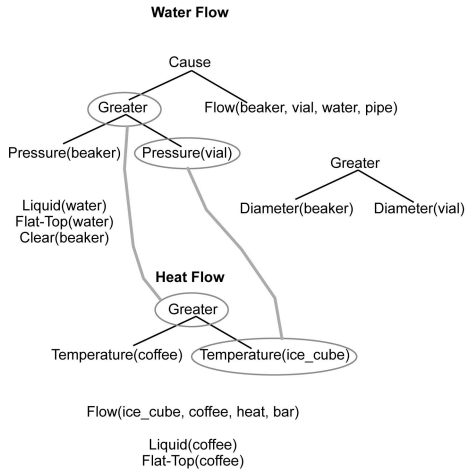


Fig. 5. A classic SME example from [2]. The analogy is between a water flow example and a heat flow example—water flows from a beaker to a vial and heat flows from a coffee cup to an ice cube. Some entities and predicates that map between the two domains are highlighted, e.g., the water pressure in the vial is analogous to the temperature in the ice cube.

From this early work on structural similarity in cognitive science, the research has established a more formal computational basis in the wider context of graph (or subgraph) isomorphism [1], [9], [10], [16]. The graph-theoretic basis for structure mapping as a similarity measure is laid out very well by Bunke and Messmer [1] and Bergmann [16]. Bergmann identifies two distinct strategies for assessing similarity between graphs: these are graph matching measures and mechanisms based on edit distance. The most fundamental graph matching problem is (complete) graph isomorphism, however, this is less relevant in CBR because exact matches are unlikely to occur. Instead, subgraph isomorphism and the problem of identifying the largest common subgraph are the relevant graph matching measures. It is interesting that the most recent work on structural similarity in CBR [9] employs graph edit distance, which belongs to the second set of strategies described by Bergmann.

The work by Minor et al. [9] is based on the original graph edit distance framework proposed by Bunke and Messmer [1]. They propose that each case is represented as a directed graph $g = (N, E, \alpha, \beta)$, where

- N is a finite set of nodes,
- $E \subseteq N \times N$ is the finite set of edges,
- $\alpha : N \mapsto L_N$ is the node labeling function, and
- $\beta : E \mapsto L_E$ is the edge labeling function.

In this framework, the distance between two graphs g and g' is the shortest sequence of edit operations that converts g to g'. A sequence of edit operations can be denoted by $s = (e_1, e_2, \ldots, e_n)$ and the cost of the sequence is $c(s) = \sum_{i=1}^{n} c(e_i)$, where $c(e_i)$ is the cost of edit operation e_i. The distance measure based on this is

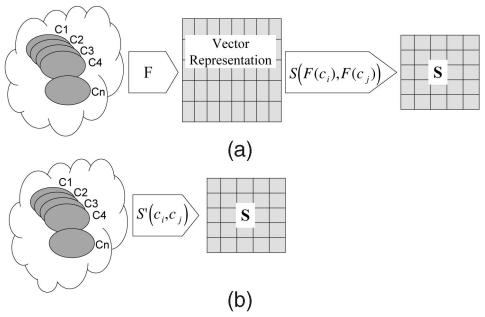


Fig. 6. A comparison of (a) feature-based and (b) featureless similarity assessment—in (a), F is the feature extraction process, $S(F(c_i), F(c_j))$ scores similarity based on features extracted from the cases, and in (b), $S'(c_i, c_j)$ operates more directly on the raw case data.

$$d(g, g') = min\left\{\sum_{i=1}^{n} c(e_i)\right\},$$

where $(e_1, \ldots, e_n)$ transforms q to q'.

The edit operations are deletion, insertion, or substitution of edges or nodes ($3 \times 2 = 6$ types of operation in all). The subgraph isomorphism problem remains NP-complete when cast in this edit distance framework. Bunke and Messmer show that for practical problems, the computational complexity can be relieved somewhat by maintaining a library of commonly occurring subgraphs, which they call models. Minor et al. show that this graph edit distance approach to similarity for structured cases is effective for the real-world problem of workflow management.

5 Information-Theoretic Measures

It should not be surprising that information theory offers a variety of techniques for assessing the similarity of two objects. Perhaps the most dramatic of these is compression-based similarity. A compression-based approach to similarity has the advantage that it works directly on the *raw* case representation (see Fig. 6b), thus it avoids the feature extraction process that produces the feature vector representation, where information might be lost (Fig. 6a). Delany and Bridge [3] refer to this as a feature-free approach to case representation.

5.1 Compression-Based Similarity for Text

In recent years, the idea of basing a similarity metric on compression has received a lot of attention [41], [42]. Indeed, Li et al. [41] refer to this as *The* similarity metric. The basic idea is quite straightforward; if two documents are very similar, then the compressed size of the two documents concatenated together will not be much greater than the compressed size of a single document. This will not be true for two documents that are very different. Slightly more formally, the difference between two documents A and B is related to the compressed size of document B when compressed using the codebook produced when compressing document A.

The theoretical basis of this metric is in the field of Kolmogorov complexity, specifically in conditional Kolmogorov complexity [41]. A definition of similarity based on Kolmogorov complexity is

$$d_{Kv}(x, y) = \frac{Kv(x|y) + Kv(y|x)}{Kv(xy)}, \tag{18}$$

where $Kv(x)$ is the length of the shortest program that computes x, $Kv(x|y)$ is the length of the shortest program that computes x when y is given as an auxiliary input to the program, and $Kv(xy)$ is the length of the shortest program that outputs y concatenated to x. While this is an abstract idea, it can be approximated using compression

$$d_C(x, y) = \frac{C(x|y) + C(y|x)}{C(xy)}. \tag{19}$$

$C(x)$ is the size of data x after compression and $C(x|y)$ is the size of x after compressing it with the compression model built for y. If we assume that $Kv(x|y) \approx Kv(xy) - Kv(y)$, then we can define a practical compression distance

$$d_{NC}(x, y) = \frac{C(xy) - \min(C(x), C(y))}{\max(C(x), C(y))}. \tag{20}$$

It is important that $C(.)$ should be an appropriate compression metric for the type of data. Delany and Bridge [3] show that compression using the Lempel-Ziv compression algorithm [43] is effective for text. They show that this compression-based metric is more accurate in k-NN classification than distance-based metrics using a bag-of-words representation of the text (i.e., a feature vector).

5.2 Information-Based Similarity for Biological Sequences

An interesting characteristic of gene sequence data is that the data are typically not compressible using standard (text) compression techniques [44]. Information theory tells us that biological sequences should be compressible because they encode information, i.e., they are not random. However, the regularity in biological sequences is more subtle than in text, thus, specialized algorithms are required to compress them. Li et al. [45] have shown that a compression-based similarity metric can be very effective for phylogenetic studies provided that a compression algorithm specialized for the data is used—they use the *GenCompress* algorithm developed by Chen et al. [44], which is based on approximate matching. This research is of general importance because it illustrates a novel strategy to bring domain knowledge to bear in assessing similarity.

It is worth mentioning that, even though we have included this in the Information-theoretic category, it has some of the characteristics of a Transformation-based metric.

5.3 Similarity in a Taxonomy

An edge counting method for assessing the similarity between two feature values in a taxonomy is shown in (15). As stated already, this has the drawback that it does not capture the fact that some is-a relationships are closer than others. Resnik has proposed an information-theoretic measure that overcomes this to a large extent [33]. It is based on the idea that the information content of a concept in a taxonomy can be quantified as the negative log likelihood $-\log(\mathrm{p(c)})$, where $\mathrm{p}(c)$ is the probability of observing the concept c. If the taxonomy has a single root, then the information content of the root node is 0. In this framework, the similarity between two features is the

information content of the common parent node with the highest information content:

$$\delta(\mathbf{q}_f, \mathbf{x}_{if}) = \max_{c \in \{S(\mathbf{q}_f, \mathbf{x}_{if})\}} -\log(\mathrm{p(c)}), \tag{21}$$

here, $S(\mathbf{q}_f, \mathbf{x}_{if})$ represents the common parent concepts of the two feature values under consideration. Other than in situations of multiple inheritance, this concept c will be $T(\mathbf{q}_f, \mathbf{x}_{if})$ the first common parent. Then the similarity is simply $\mathrm{p}(T(\mathbf{q}_f, \mathbf{x}_{if}))$, the frequency of occurrence of $T(\mathbf{q}_f, \mathbf{x}_{if})$ and its descendants in the training data.

This measure for similarity within taxonomies has performed well in evaluations and is worthy of consideration [33], [46].

6 Emergent Measures

The great increase in computing power available in recent years has resulted in a sea change in ML research objectives. Speeding up algorithms is less important now, instead the challenge is to find ways to usefully exploit the power available [47]. Techniques that represent this new direction in ML research are random forests [48], ensemble clustering [49], and stability-based cluster validation [50]. The thing these techniques have in common is that significant processing power is applied in order to produce a characterization of the data. This characterization of the data may present a significant opportunity for knowledge discovery. Of interest here are novel similarity scores that emerge from this characterization, often as a *side effect* of an analysis of the data for another purpose. The three techniques we consider here are random forests, cluster kernels, and Web-based kernels. These techniques are *emergent* similarity measures in that the similarity measure emerges as a side effect of an analysis of the data that also had another purpose.

In analogy with the way feature-based representations can be internal or external as described in Section 2.1.1, these emergent mechanisms can be categorized as internal or external. Random forests and cluster kernels are internal in that they uncover new relationships from an analysis within the data set whereas Web-based kernels bring new knowledge by leveraging processing that was carried out for the purpose of Web indexing and search.

6.1 Random Forests

A random forest is an ensemble of decision trees [48]. The general strategy for generating a random forest is as follows:

1. For each ensemble member, the training set D is subsampled with replacement to produce a training set of size $|D|$. (The remaining cases are referred to as the out-of-bag (OOB) cases for that ensemble member).
2. Where F is the set of features that describes the data, $m \ll |F|$ is selected as the number of features to be used in the feature selection process. At each stage (i.e., node) in the building of a tree, m features are selected at random to be the candidates for splitting at that node.

In order to ensure diversity among the component trees, no pruning is employed as would be normal in building decision trees. It is normal when building a random forest to generate many more ensemble members than would be used in other ensemble techniques—100 or even 1,000 trees might be built. The effort expended on building these trees has the added benefit of providing an analysis of the data.

In particular, a novel similarity measure *emerges* from all of these trees. The idea is to track the frequency with which cases (both training and OOB) are located at the same leaf node. Every leaf node in every tree is examined and a $|D| \times |D|$ matrix is maintained, where cell (i, j) is incremented each time cases i and j share the same leaf node. If the matrix entries are divided by the number of trees, we have a proximity measure that is *in tune* with the classification algorithm (the random forest). In [51], we have shown that this similarity metric is more effective than a conventional feature-based similarity metric on a wide range of classification tasks.

6.2 Cluster Kernels

Cluster kernels are relevant in the context of semisupervized learning, where only some of the available data are labeled [52]. Cluster kernels allow the unlabelled data to influence similarity. This is driven by the *cluster assumption* that class labels do not change in regions of high density—instead, there should be some correspondence between cluster boundaries and the unknown class boundaries. Thus, the cluster kernel is a composition of a standard kernel built from the labeled data and a kernel derived from clustering *all* the data. This is a general principle and one embodiment of this idea for protein sequence classification is [53]:

$$K(x_i, x_j) = K_{orig}(x_i, x_j) \cdot K_{bag}(x_i, x_j), \tag{22}$$

where $K_{orig}()$ is a basic neighborhood kernel and $K_{bag}()$ is a kernel derived from repeated clustering of all the data. $K_{bag}(x_i, x_j)$ is essentially a count of the number of times x_i and x_j turned up in the same cluster (this is in the same spirit as the strategy used in random forests). Thus, we have a mechanism that adjusts the similarity metric using measures drawn from repeated clustering of the labeled and unlabelled data. Evaluation of the use of cluster kernels in semisupervized learning shows promising results [53], [52].

6.3 Web-Based Kernel

One way to discover that two phrases are related would be to find that they are connected by documents returned in Web search. This is the approach taken by Sahami and Heilman [54] in their Web-based kernel for text snippet similarity. The basic idea is that the sets of documents returned when two text snippets are submitted as Web queries will show similarities if the text snippets are related—even if the snippets have no terms in common.

An illustration of a Web-based kernel in operation is shown in Fig. 7. The two text snippets to be compared are x and y and the vectors corresponding to the return sets from presenting these to a Web search engine are $R(x)$ and $R(y)$. The return sets are restricted to the first n documents and each document is represented by m terms selected by the standard term-frequency inverse-document frequency (TFIDF) criterion. The text snippets can now be represented by $C(x)$ and $C(y)$, the centroids of $R(x)$ and $R(y)$. The Web-based kernel is simply the dot-product of $C(x)$ and $C(y)$.

Because the Web-based kernel effectively leverages other processing for Web indexing and search, it is able to

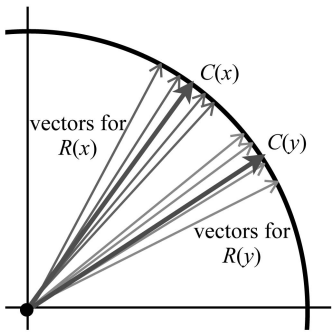


Fig. 7. The principle underlying the Web-based kernel. $R(x)$ and $R(y)$ are the vectors corresponding to the return sets for x and y. $C(x)$ and $C(y)$ are the centroid vectors for these vectors.

identify similarities between text snippets that have *no* surface similarity. Some examples of this (reproduced from [54]) are shown in Table 4. For instance, there is a strong connection between "UN Secretary General" and "Kofi Annan" but not between "UN Secretary General" and "George W. Bush." Sahami and Heilman [54] also show how similarity based on this strategy is very effective for the task of *related query suggestion*.

7 Implications for CBR Research

The new perspectives on similarity outlined in this paper do not represent a paradigm shift in CBR research since direct assessment of similarity based on a feature value representation will still be the dominant strategy. Instead, what we have is a selection of new methodologies for similarity in CBR that may be used as alternatives to the direct strategy in some circumstances. These new methodologies have two important ramifications for CBR as follows:

- In some circumstances (e.g., information-theoretic measures) the role of the vocabulary knowledge container [55] is de-emphasized and the role of the similarity knowledge container is increased.
- Because these new methodologies are typically computationally intensive, the importance of strategies for speeding up case-retrieval is increased.

These issues will be discussed in the following sections.

7.1 De-Emphasizing the Vocabulary Knowledge Container

A popular perspective on the organization of knowledge in CBR is Richter's knowledge containers model [55]—there are four containers:

1. The case description language (vocabulary).
2. The similarity measure.
3. The solution transformation knowledge (adaptation knowledge).
4. The cases themselves.

An important aspect of the knowledge containers' view of CBR is the way it highlights the high-level design decisions in the development of a CBR system. Different design choices have the effect of moving knowledge from one container to another. Clearly, the information-theoretic strategies, described in Section 5, move knowledge from the vocabulary knowledge container into the similarity measure.

TABLE 4
Web-Based Kernel Scores for "Similarity" between Individuals and Their Positions—Example Taken from [54]

Title	Name	WBK score
UN Secretary-General	Kofi Annan	0.825
UN Secretary-General	George W. Bush	0.110
US President	George W. Bush	0.688

In our early work on case-based spam filtering, we worked with a vector space representation of messages [56]. This approach meant that careful consideration had to be given to feature extraction and feature selection strategies. Feature selection in the context of concept drift is a particular problem. The solution we adopted required a periodic feature reselection process. This is all much more straightforward when compression-based similarity is used [3] as there is no feature extraction and selection.

This is also true for transformation-based measures such as edit distance and string alignment kernels. However, the EMD does have a feature extraction stage, where the signature is created using clustering—the granularity of this signature is key to the effectiveness of the algorithm. The important point is that the use of novel similarity measures can greatly simplify the design of other aspects of the system.

7.2 Computational Complexity

Computationally expensive metrics such as the EMD and compression-based (dis)similarity metrics focus attention on the computational issues associated with case-based classifiers. Basic case-based classifiers that use a simple Minkowski distance will have a time behavior, that is, $O(|D||F|)$, where D is the training set and F is the set of features that describes the data, i.e., the distance metric is linear in the number of features and the comparison process increases linearly with the amount of data. The computational complexity of the EMD and compression metrics is more difficult to characterize, but a case-based classifier that incorporates an EMD metric is likely to be $O(|D|n^3 \log n)$, where n is the number of clusters [31]. The computational cost of compression-based similarity depends on the compression algorithm—for text, GZip is roughly linear while PPM is $O(n^2)$ [57]. Even the linear time GZip is much slower than a feature-based measure. Delany and Bridge show that compression-based similarity using GZip can be 200 times slower than the feature-based alternative [3]. There has been considerable research on alternatives to the exhaustive search strategy that is used in the standard k-NN algorithm. Here is a summary of four of the strategies for speeding up nearest neighbor retrieval:

- **Case retrieval nets (CRNs).** These are perhaps the most popular technique for speeding up the retrieval process. The cases are preprocessed to form a network structure that is used at retrieval time. The retrieval process is done by *spreading activation* in this network structure. CRNs can be configured to return exactly the same cases as k-NN [58], [59]. However, CRNs depend on a feature-based case representation.

- **Footprint-based retrieval.** As with all strategies for speeding up nearest neighbor retrieval, Footprint-Based Retrieval involves a preprocessing stage to organize the training data into a two-level hierarchy on which a two-stage retrieval process operates. The preprocessing constructs a competence model, which identifies "footprint" cases that are landmark cases in the data. This process is not guaranteed to retrieve the same cases as k-NN, but the results of the evaluation of speedup and retrieval quality are nevertheless impressive [60].
- **Fish and shrink.** This technique requires the distance to be a true metric as it exploits the triangle inequality property to produce an organization of the case-base into candidate neighbors and cases excluded from consideration. Cases that are remote from the query can be bounded out so that they need not be considered in the retrieval process. Fish and Shrink can be guaranteed to be equivalent to k-NN [23].
- **Cover trees for nearest neighbor.** This technique might be considered the state of the art in nearest neighbor speedup. It uses a data structure called a Cover Tree to organize the cases for efficient retrieval. The use of Cover Trees requires that the distance measure is a true metric, however, they have attractive characteristics in terms of space requirements and speedup performance. The space requirement is $O(n)$, where n is the number of cases; the construction time is $O(c^6 n \log n)$ and the retrieval time is $O(c^{12} \log n)$, where c is a measure of the inherent dimensionality of the data [22].

While CRNs are only applicable for speeding up retrieval for feature-based representations, the other three techniques are more broadly applicable. Fish and Shrink and Cover Trees do require similarity measures that are true metrics and will not work for measures that do not obey the triangle inequality. However, Footprint-Based Retrieval is applicable for most similarity measures.

8 Conclusion

This paper presents an attempt to organize the broad range of strategies for similarity assessment in CBR into a coherent taxonomy. The taxonomy has four main categories: Direct, Transformation-Based, Information-Theoretic, and Emergent measures. We have emphasized the centrality of feature value representation of cases coupled with direct similarity measures in the CBR paradigm. We have given several examples, where alternative similarity measures have provided great benefits in CBR. In some circumstances, the benefit is that the overall design of the system is simplified or it may be that the alternative metric simply offers better accuracy because it embodies specific knowledge about the data. This is very much in line with current research on support vector machines, where it is understood that the effectiveness of the classifier depends very much on the appropriateness of the kernel function for the data (i.e., in encoding domain knowledge in the kernel function).

Thus, there are two considerations for CBR research. The first is that a broader perspective on similarity along the lines discussed in this paper may be useful. The second is that the practicality of these computationally expensive similarity measures may depend on clever retrieval techniques such as Cover Trees to make them computationally tractable.

Acknowledgments

The author is grateful to Derek Bridge, Sarah Jane Delany, and Jean Lieber for helpful discussions on this paper. This research was supported by the Science Foundation Ireland Grant Nos. 05/IN.1/I2, 05/IN.1/I24, and EU-funded Network of Excellence Muscle—Grant No. FP6-507752.

References

[1] H. Bunke and B.T. Messmer, "Similarity Measures for Structured Representations," *Proc. European Workshop Case-Based Reasoning (EWCBR '93)*, S. Wess, K.D. Althoff, and M.M. Richter, eds., pp. 106-118, 1993.

[2] B. Falkenhainer, K.D. Forbus, and D. Gentner, "The Structure-Mapping Engine," *Proc. Conf. Assoc. for the Advancement of Artificial Intelligence (AAAI '86)*, pp. 272-277, 1986.

[3] S.J. Delany and D.G. Bridge, "Catching the Drift: Using Feature-Free Case-Based Reasoning for Spam Filtering," *Proc. Int'l Conf. Case-Based Reasoning (ICCBR '07)*, R. Weber and M.M. Richter, eds., pp. 314-328, 2007.

[4] J.L. Arcos, M. Grachten, and R.L. de Mántaras, "Extracting Performers' Behaviors to Annotate Cases in a cbr System for Musical Tempo Transformations," *Proc. Int'l Conf. Case-Based Reasoning (ICCBR '03)*, K.D. Ashley and D.G. Bridge, eds., pp. 20-34, 2003.

[5] E. Costello and D.C. Wilson, "A Case-Based Approach to Gene Finding," *Proc. Fifth Int'l Conf. Case-Based Reasoning Workshop CBR in the Health Sciences*, pp. 19-28, 2003.

[6] N. Wiratunga, I. Koychev, and S. Massie, "Feature Selection and Generalisation for Retrieval of Textual Cases," *Proc. European Conf. Case Based Reasoning (ECCBR '04)*, P. Funk and P.A. González-Calero, eds., pp. 806-820, 2004.

[7] E. Gabrilovich and S. Markovitch, "Overcoming the Brittleness Bottleneck Using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge," *Proc. 21st Nat'l Conf. Artificial Intelligence*, pp. 1301-1306, 2006.

[8] R. Bergmann and A. Stahl, "Similarity Measures for Object-Oriented Case Representations," *Proc. European Workshop Case-Based Reasoning (EWCBR '98)*, B. Smyth and P. Cunningham, eds., pp. 25-36, 1998.

[9] M. Minor, A. Tartakovski, and R. Bergmann, "Representation and Structure-Based Similarity Assessment for Agile Workflows," *Proc. Seventh Int'l Conf. Case-Based Reasoning (ICCBR '07)*, R.O. Weber and M.M. Richter, eds., pp. 224-238, 2007.

[10] E. Plaza, "Cases as Terms: A Feature Term Approach to the Structured Representation of Cases," *Proc. Int'l Conf. Case-Based Reasoning (ICCBR '95)*, M.M. Veloso and A. Aamodt, eds., pp. 265-276, 1995.

[11] K.E. Sanders, B.P. Kettler, and J.A. Hendler, "The Case for Graph-Structured Representations," *Proc. Int'l Conf. Case-Based Reasoning (ICCBR '97)*, D.B. Leake and E. Plaza, eds., pp. 245-254, 1997.

[12] B. Smyth and P. Cunningham, "Déjà Vu: A Hierarchical Case-Based Reasoning System for Software Design," *Proc. European Conf. Artificial Intelligence (ECAI '92)*, pp. 587-589, 1992.

[13] B. Smyth, M.T. Keane, and P. Cunningham, "Hierarchical Case-Based Reasoning Integrating Case-Based and Decompositional Problem-Solving Techniques for Plant-Control Software Design," *IEEE Trans. Knowledge and Data Eng.* vol. 13, no. 5, pp. 793-812, Sept. 2001.

[14] M.T. Keane and M. Brayshaw, "The Incremental Analogy Machine: A Computational Model of Analogy," *Proc. European Working Session on Learning (EWSL '88)*, pp. 53-62, 1988.

[15] M.M. Veloso and J.G. Carbonell, "Case-Based Reasoning in PRODIGY," *Machine Learning: A Multistrategy Approach*, R.S. Michalski and G. Teccuci, eds., vol. IV, pp. 523-548, Morgan Kaufmann, 1994.

[16] R. Bergmann, "Experience Management: Foundations, Development Methodology, and Internet-Based Applications," *Lecture Notes in Computer Science,* vol. 2432, Springer, 2002.

[17] M. Lenz and K. Ashley, *Proc. AAAI '98 Workshop Textural Case-Based Reasoning,* 1998.

[18] H. Shimazu, "A Textual Case-Based Reasoning System Using xml on the World-Wide Web," *Proc. Fourth European Workshop Case-Based Reasoning,* B. Smyth and P. Cunningham, eds., pp. 274-285, 1998.

[19] C. Stanfill and D.L. Waltz, "Toward Memory-Based Reasoning," *Comm. ACM,* vol. 29, pp. 1213-1228, 1986.

[20] D. Wilson and T. Martinez, "Improved Heterogeneous Distance Functions," *J. Artificial Intelligence Research,* vol. 6, pp. 1-34, 1997.

[21] E. Blanzieri and F. Ricci, "A Minimum Risk Metric for Nearest Neighbor Classification," *Proc. 16th Int'l Conf. Machine Learning,* pp. 22-31, 1999.

[22] A. Beygelzimer, S. Kakade, and J. Langford, "Cover Trees for Nearest Neighbor," *Proc. 23rd Int'l Conf. Machine Learning (ICML '06),* 2006.

[23] J. Schaaf, "Fish and Shrink. A Next Step Towards Efficient Case Retrieval in Large-Scale Case Bases," *Proc. European Workshop Case-Based Reasoning (EWCBR '96),* I. Smith and B. Faltings, eds., pp. 362-376, 1996.

[24] S. Santini and R. Jain, "Similarity Measures," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 9, pp. 871-883, Sept. 1999.

[25] A. Tversky, "Features of Similarity," *Psychological Rev.,* vol. 84, pp. 327-352, 1977.

[26] T. Tanimoto, "An Elementary Mathematical Theory of Classification and Prediction [Z]," technical report, IBM Corp., 1958.

[27] P. Jaccard, "The Distribution of the Flora in the Alpine Zone," *New Phytologist,* vol. 11, pp. 37-50, 1912.

[28] L. Dice, "Measures of the Amount of Ecologic Association between Species," *Ecology,* vol. 26, pp. 297-302, 1945.

[29] D. Greene, A. Tsymbal, N. Bolshakova, and P. Cunningham, "Ensemble Clustering in Medical Diagnostics," *Proc. 17th IEEE Symp. Computer-Based Medical Systems (CBMS '04),* pp. 576-581, 2004.

[30] S. Kullback and R.A. Leibler, "On Information and Sufficiency," *Annals of Math. Statistics,* vol. 22, pp. 79-86, 1951.

[31] Y. Rubner, C. Tomasi, and L.J. Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval," *Int'l J. Computer Vision,* vol. 40, pp. 99-121, 2000.

[32] Z. Wu and M.S. Palmer, "Verb Semantics and Lexical Selection," *Proc. 32nd Ann. Meeting Assoc. for Computational Linguistics (ACL '94),* pp. 133-138, 1994.

[33] P. Resnik, "Using Information Content to Evaluate Semantic Similarity in a Taxonomy," *Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI '95),* pp. 448-453, 1995.

[34] V. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," *Problems in Information Transmission,* vol. 1, pp. 8-17, 1965.

[35] B. Smyth and M.T. Keane, "Adaptation-Guided Retrieval: Questioning the Similarity Assumption in Reasoning," *Artificial Intelligence,* vol. 102, pp. 249-293, 1998.

[36] J.P. Vert, H. Saigo, and T. Akutsu, "Local Alignment Kernels for Biological Sequences," *Kernel Methods in Computational Biology,* B. Schölkopf, K. Tsuda, and J.P. Vert, eds., MIT Press, 2004.

[37] S. Needleman and C. Wunsch, "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins," *J. Molecular Biology,* vol. 48, pp. 443-453, 1970.

[38] T. Smith and M. Waterman, "Identification of Common Molecular Subsequences," *J. Molecular Biology,* vol. 147, pp. 195-197, 1981.

[39] D. Gentner, "Structure-Mapping: A Theoretical Framework for Analogy," *Cognitive Science,* vol. 7, pp. 155-170, 1983.

[40] T. Veale and M.T. Keane, "The Competence of Sub-Optimal Theories of Structure Mapping on Hard Analogies," *Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI '97),* vol. 1, pp. 232-237, 1997.

[41] M. Li, X. Chen, X. Li, B. Ma, and P.M.B. Vitányi, "The Similarity Metric," *IEEE Trans. Information Theory,* vol. 50, no. 12 , pp. 3250-3264, Dec. 2004.

[42] E.J. Keogh, S. Lonardi, and C. Ratanamahatana, "Towards Parameter-Free Data Mining," *Proc. ACM Int'l Conf. Knowledge Discovery and Data Mining,* W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, eds., pp. 206-215, 2004.

[43] J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression," *IEEE Trans. Information Theory,* vol. 23, no. 3, pp. 337-343, May 1977.

[44] X. Chen, S. Kwong, and M. Li, "A Compression Algorithm for DNA Sequences and Its Applications in Genome Comparison," *Proc. Int'l Conf. Research in Computational Molecular Biology (RECOMB '00),* vol. 107, 2000.

[45] M. Li, J.H. Badger, X. Chen, S. Kwong, P.E. Kearney, and H. Zhang, "An Information-Based Sequence Distance and Its Application to Whole Mitochondrial Genome Phylogeny," *Bioinformatics,* vol. 17, pp. 149-154, 2001.

[46] N. Bolshakova, F. Azuaje, and P. Cunningham, "Incorporating Biological Domain Knowledge into Cluster Validity Assessment," *Proc. EvoWorkshops—Applications of Evolutionary Computing,* F. Rothlauf, J. Branke, S. Cagnoni, E. Costa, C. Cotta, R. Drechsler, E. Lutton, P. Machado, J.H. Moore, J. Romero, G.D. Smith, G. Squillero, and H. Takagi, eds., pp. 13-22, 2006.

[47] S. Esmeir and S. Markovitch, "Anytime Induction of Decision Trees: An Iterative Improvement Approach," *Proc. 21st Nat'l Conf. Artificial Intelligence,* pp. 348-355, 2006.

[48] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, pp. 5-32, 2001.

[49] D. Greene, A. Tsymbal, N. Bolshakova, and P. Cunningham, "Ensemble Clustering in Medical Diagnostics," *Proc. IEEE Symp. Computer-Based Medical Systems (CBMS '04),* pp. 576-581, 2004.

[50] T. Lange, V. Roth, M.L. Braun, and J.M. Buhmann, "Stability-Based Validation of Clustering Solutions," *Neural Computation,* vol. 16, pp. 1299-1323, 2004.

[51] A. Tsymbal, M. Pechenizkiy, and P. Cunningham, "Dynamic Integration with Random Forests," *Proc. European Conf. Machine Learning (ECML '06),* J. Fürnkranz, T. Scheffer, and M. Spiliopoulou, eds., pp. 801-808, 2006.

[52] O. Chapelle, J. Weston, and B. Schölkopf, "Cluster Kernels for Semi-Supervised Learning," *Advances in Neural Information Processing Systems (NIPS),* S. Becker, S. Thrun, and K. Obermayer, eds., pp. 585-592, MIT Press, 2002.

[53] J. Weston, C.S. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W.S. Noble, "Semi-Supervised Protein Classification Using Cluster Kernels," *Bioinformatics,* vol. 21, pp. 3241-3247, 2005.

[54] M. Sahami and T. Heilman, "A Web-Based Kernel Function for Measuring the Similarity of Short Text Snippets," *Proc. 15th Int'l Conf. World Wide Web,* pp. 377-386, 2006.

[55] M.M. Richter, "Introduction," *Case-Based Reasoning Technology,* M. Lenz, B. Bartsch-Spörl, H.D. Burkhard, and S. Wess, eds., pp. 1-16, Springer, 1998.

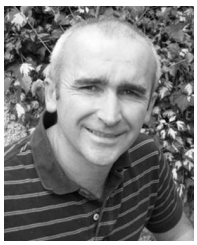
[56] S. Delany, P. Cunningham, and B. Smyth, "ECUE: A Spam Filter That Uses Machine Learning to Track Concept Drift," *Proc. 17th European Conf. Artificial Intelligence (ECAI '06),* G.C.S Brewka, A. Perini, and P. Traverso, eds., pp. 627-631, 2006.

[57] T.C. Bell, I.H. Witten, and J.G. Cleary, *Text Compression.* Prentice Hall, 1990.

[58] M. Lenz, H.-D. Burkhard, and S. Brückner, "Applying Case Retrieval Nets to Diagnostic Tasks in Technical Domains," *Proc. European Workshop Case-Based Reasoning (EWCBR '96),* I.F.C. Smith and B. Faltings, eds., pp. 219-233, 1996.

[59] M. Lenz and H.D. Burkhard, "Case Retrieval Nets: Basic Ideas and Extensions," *KI—Kunstliche Intelligenz–96: Advances in Artificial Intelligence (Proc. 20th Ann. German Conf. AI),* pp. 227-239, 1996.

[60] B. Smyth and E. McKenna, "Footprint-Based Retrieval," *Proc. Int'l Conf. Case-Based Reasoning (ICCBR '99),* K.D. Althoff, R. Bergmann, and K. Branting, eds., pp. 343-357, 1999.



Pádraig Cunningham received the BE and MEng Sci degrees from NUI Galway and the PhD degree from Dublin University in 1989. He is a professor of knowledge and data engineering in the School of Computer Science and Informatics at the University College Dublin. His current research focus is on the use of machine learning techniques in processing high-dimension data. He became a fellow of the European Coordinating Committee on Artificial Intelligence (ECCAI) in 2004. He has published more than 150 peer-reviewed papers in the general area of applied AI, focusing on machine learning and knowledge-based systems for decision support in multimedia, bioinformatics, and medicine.