Featureless Similarity*

Pádraig Cunningham[1] and Sarah Jane Delany[2]

[1] University College Dublin Padraig.Cunningham@ucd.ie
[2] Dublin Institute of Technology Sarahjane.Delany@comp.dit.ie

Technical Report UCD-CSI-2007-1
February 23, 2007

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 world-view – 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 feature-based. Some of these ideas have emerged in CBR research but many of them have arisen in other areas of data analysis. In fact research on Support Vector Machines(SVM) is a rich source of novel similarity representations because of the emphasis on encoding domain knowledge in the kernel function of the SVM. In this paper we review these novel *featureless* similarity measures and assess the implications these measures have for CBR research.

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 Figure 1(a)). This methodology shapes the CBR paradigm; it means that problem spaces are visualised as vector spaces, it informs the notion of a decision surface and how we conceptualise noisy cases, and it motivates feature extraction and dimension reduction which are key processes in CBR systems development.

But this approach has its drawbacks, the main issue being that it introduces a level of abstraction between the similarity mechanism and the cases that are being compared. It is inevitable that there will be a loss of information in the feature extraction process that produces the feature vector representation (labelled as F in Figure 1(a)). If similarity is assessed more directly on the raw case data ($S'(c_i, c_j)$ in Figure 1(b)) then there is the potential for the similarity score to be more accurate. This strategy also broadens the notion of similarity

* This research was supported by Science Foundation Ireland Grant No. 05/IN.1/I24

and increases the potential to incorporate domain knowledge in the similarity mechanism.

In this paper we review a number of strategies for similarity that are *featureless* in this sense and consider the implications that this new paradigm has for CBR research. Some of these strategies have already been presented in CBR research (e.g. compression-based similarity in [1] and Edit Distance in [2, 3]) but most come from other areas of Machine Learning (ML) research. We propose that these strategies can be divided into three categories:

- Information theoretic measures,
- Transformation-based measures,
- Emergent measures arising from an in-depth analysis of the data.

This idea of featureless representation has also received some attention in Pattern Recognition research under the title of *dissimilarity representations* [4] – this title is motivated by the idea that the dissimilarity (or similarity) scores between objects ((matrix S in Figure 1(b)) is the central knowledge representation.

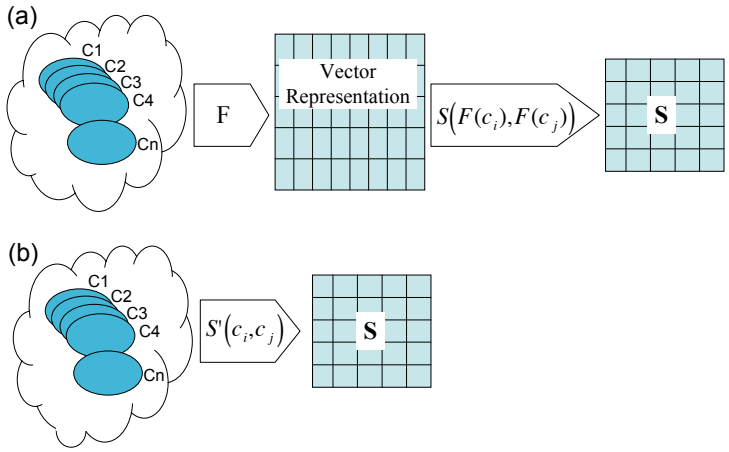

Fig. 1. A comparison of Feature-based (a) and Featureless (b) similarity assessment – in (a) F is the Feature Extraction process, $S(F(c_i), F(c_j))$ scores similarity based on features extracted from the cases and in (b) $S'(c_i, c_j)$ operates more directly on the raw case data.

Before embarking on a review of featureless similarity measures in section 3 we summarise the standard feature-based approach to similarity in section 2. Then in section 4 we will review the implications of this new similarity paradigm for CBR research. The paper concludes with a summary and suggestions for future research in section 6.

2 Feature-based Similarity

The fundamental idea in CBR is problem solving based on the retrieval of similar cases, in its simplest form this is Nearest Neighbour Classification. The intuition underlying this is quite straightforward, examples are classified based on the class of their nearest neighbours. It is often useful to take more than one neighbour into account so the technique is more commonly referred to as k-Nearest Neighbour (k-NN) Classification where k nearest neighbours are used in determining the class.

The basic idea is as shown in Figure 2 which depicts a 3-Nearest Neighbour 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 neighbours are of class O so it is classified as an O. The situation for q_2 is a bit more complicated at it has two neighbours of class X and one of class O. This can be resolved by simple majority voting or by distance weighted voting (see below).

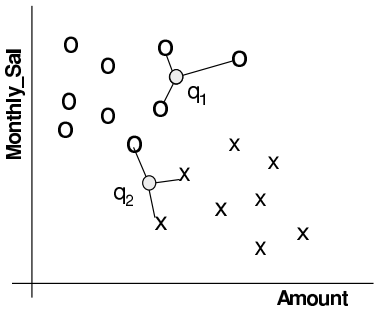

Fig. 2. A simple example of 3-Nearest Neighbour Classification

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

Let us assume that we have a training dataset 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 normalised to the range [0,1]. Each training example is labelled with a class label $y_j \in Y$. Our objective is to classify an unknown example $\mathbf{q}$. For each $\mathbf{x}_i \in D$ we can calculate the distance between $\mathbf{q}$ and $\mathbf{x}_i$ as follows:

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

There are a large range of possibilities for this distance metric; a basic version for continuous and discrete attributes would be:

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

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

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

$$Vote(y_j) = \sum_{c=1}^{k} \frac{1}{d(\mathbf{q}, \mathbf{x}_c)^n} 1(y_j, y_c) \tag{3}$$

Thus the vote assigned to class y_j by neighbour $\mathbf{x}_c$ is 1 divided by the distance to that neighbour, i.e. $1(y_j, y_c)$ returns 1 if the class labels match and 0 otherwise. In equation 3 n would normally be 1 but values greater than 1 can be used to further reduce the influence of more distant neighbours.

2.1 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 criteria (where $d(x, y)$ refers to the distance between two objects x and y):

1. $d(x, y) \geq 0$; non-negativity
2. $d(x, y) = 0$ iff $x = y$; identity
3. $d(x, y) = d(x, y)$; 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 optimisations to the basic k-NN algorithm that require the use of a proper metric [5, 6]. In brief, these techniques can identify the nearest neighbour 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 equations 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

$$MD_p(\mathbf{q}, \mathbf{x}_i) = \left(\sum_{f \in F} |\mathbf{q}_f - \mathbf{x}_{if}|^p \right)^{\frac{1}{p}} \tag{4}$$

The L_1 Minkowski distance is 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 neighbour 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 it takes a chess king to reach any square on the board.

2.2 Kullback-Leibler Divergence and the χ^2 Statistic

The Minkowski distance defined in (4) is a very general metric that can be used in a k-NN classifier for any data that is represented as a feature vector. When working with image data a convenient representation for the purpose of calculating distances is a colour histogram. An image can be considered as a grey-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 formula (4) 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 (5) [7] and the χ^2 statistic (6) [8].

$$d_{KL}(H, K) = \sum_{i=1}^{N} h_i \log\left(\frac{h_i}{k_i}\right) \tag{5}$$

$$d_{\chi^2}(H, K) = \sum_{i=1}^{N} \frac{h_i - m_i}{h_i} \tag{6}$$

where H and K are two histograms, h and k are the corresponding vectors of bin values and $m_i = \frac{h_i + k_i}{2}$.

While these measures have sound theoretical support in information theory and in 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 [8] 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.

2.3 Summary

The important point to emphasise is that the use of feature-based representations in CBR has wide ranging implications. It formulates the problem space as a vector space, it emphasises the dimensionality of the data as an issue and it abstracts the similarity mechanism from the raw data.

3 Featureless Similarity Measures

The alternative strategy is depicted in Figure 1(b) where similarity is assessed more directly from the raw data. In this section we describe a variety of similarity measures that are not based on a vector space representation of the data. Some of these measures are genuinely *featureless* in the sense that the raw data is used in similarity assessment (e.g. compression-based similarity). Others, such as the Earth-Mover Distance, are based on a quantization of the data so they are not strictly featureless, however the quantization of the image does not produce a vector space representation. We organise these *featureless* similarity metrics as follows:

- Information Theoretic Measures
  - Compression-based similarity for text [9, 1]
  - Information-based gene sequence similarity [10]
- Transformation-based measures
  - Edit distance (Levenshtein distance) [11]
  - Alignment kernels for biological sequences [12]
  - Earth-mover distance [8] (also knows as the Mallows distance [13])
- Emergent measures
  - Random forests [14]
  - Cluster kernels [15]

3.1 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.

Compression-Based Similarity for Text: In recent years the idea of basing a similarity metric on compression has received a lot of attention [9, 16]. Indeed Li et al. [9] 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 [9]. A definition of similarity based on Kolmogorov complexity is:

$$d_{Kv}(x, y) = \frac{Kv(x|y) + Kv(y|x)}{Kv(xy)} \tag{7}$$

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

$$d_C(x, y) = \frac{C(x|y) + C(y|x)}{C(xy)} \tag{8}$$

$C(x)$ 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

$$d_{NC}(x, y) = \frac{C(xy) - \min(C(x), C(y))}{\max(C(x), C(y))} \tag{9}$$

It is important that $C(.)$ should be an appropriate compression metric for the type of data. Delany and Bridge [1] show that compression using Lempel-Ziv (GZip) 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).

Information-Based Similarity for Biological Sequences An interesting characteristic of gene sequence data is that the data is typically not compressible using standard (text) compression techniques [17]. 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 specialised algorithms are required to compress them. Li. et al. [10] have shown that a compression based similarity metric can be very effective for phylogenitic studies provided that a compression algorithm specialised for the data is used – they use the *GenCompress* algorithm developed by Chen et al. [17] 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.

3.2 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 [11]. Edit Distance has already been used in CBR research as an affinity measure [2, 3], indeed the whole CBR field of Adaptation Guided retrieval is based on a view of similarity as 'transformation effort' [18].

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.

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 protein sequences `GAATCCG` and `GATTGC` might be aligned as follows [19]:

```
G-AATCCG-
GAT-T-G-C
```

This alignment is scored as the the sum of the contributions of the alignment scores (`G` $\leftrightarrow$ `G`, `A` $\leftrightarrow$ `T`, `T` $\leftrightarrow$ `T`, etc.) minus penalty terms for the gaps. The alignment scores (e.g. `A` $\leftrightarrow$ `T`) are read from a substitution matrix. For any pair of strings there will be a number of possible alignments and associated scores. The actual alignment score is the one associated with the alignment with the highest score – the Smith-Waterman algorithm determines this using dynamic programming [19]. This is a transformation score that is specialised 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.

Earth Mover Distance The Earth Mover Distance (EMD) is a 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 2.2). 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 Figure 3).

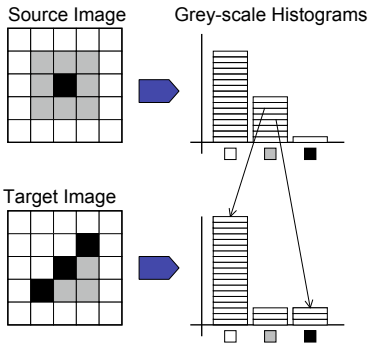

Fig. 3. An example of the EMD effort required to transform one image to another with images represented as histograms.

In their analysis of the EMD Rubner et al. [8] 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 generalisation 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 [8].

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 $\mathbf{F}$:

$$WORK(S, Q, \mathbf{F}) = \sum_{j=1}^{n} \sum_{k=1}^{r} d_{jk} f_{jk} \tag{10}$$

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 minimises overall cost. An example of this in a 2D colour space is shown in Figure 4. Once the transportation problem of identifying the flow that minimises effort is solved (using dynamic programming) the EMD is defined to be:

$$EMD(S, Q) = \frac{\sum_{j=1}^{n} \sum_{k=1}^{r} d_{jk} f_{jk}}{\sum_{j=1}^{n} \sum_{k=1}^{r} f_{jk}} \tag{11}$$

Efficient algorithms for the EMD are described in [8] 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.

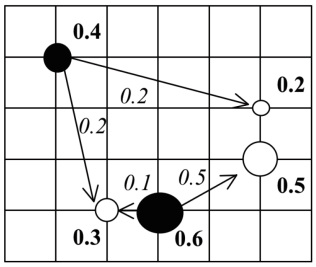

Fig. 4. An example of the EMD effort required to transform one image to another with images represented as signatures: the source image is represented by two clusters (black circles) and the target image by three clusters.

3.3 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 [20]. Techniques that represent this new direction in ML research are; Random Forests [14], Ensemble Clustering [21], and Stability-based cluster validation [22]. These might be viewed as analytic processes that are allowed to *simmer* for an extended period to produce a characterisation of the data. Of interest here are novel similarity scores that emerge from this characterisation. The two techniques we consider here are Random Forests and Cluster Kernels: these are not *featureless* given that the basic analysis is done on a feature-based representation of the data. However, the emergent similarity measures are not describable in a feature space.

Random forests A Random Forest is an ensemble of decision trees [14]. The general strategy for generating a Random forest is as follows:

1. For each ensemble member the training set D is sub-sampled 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, $m << |F|$ 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 1000 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 [23] we have shown that this similarity metric is more effective than a conventional feature-based similarity metric on a wide range of classification tasks.

Cluster kernels Cluster Kernels are relevant in the context of semi-supervised learning where only some of the available data is labelled [24]. 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 labelled 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 [15]:

$$K(x_i, x_j) = K_{orig}(x_i, x_j) \cdot K_{bag}(x_i, x_j) \tag{12}$$

where $K_{orig}()$ is a basic neighbourhood 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 labelled and unlabelled data. Evaluations of the use of cluster kernels in semi-supervised learning shows promising results [15] [24].

4 Implications for CBR Research

The new perspective on similarity outlined in this paper is not a paradigm shift in CBR research since the feature-based view of similarity will still be the dominant strategy. Instead this represents a new methodology for similarity in CBR. This featureless methodology has two important ramifications for CBR:

- Where it is used, it de-emphasises the vocabulary knowledge container [25] and increases the role of the similarity knowledge container.
- It increases the importance of strategies for speeding up case-retrieval as these featureless similarity measures are usually very computationally expensive.

These issues will be discussed in the following sections.

5 De-emphasising the Vocabulary Knowledge Container

A popular perspective on the organisation of knowledge in CBR is Richter's knowledge containers model [25] - 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 a 'featureless' approach to similarity moves knowledge from the vocabulary knowledge container into the similarity measure.

In our early work on case-based spam filtering we worked with a vector-space representation of messages [26]. 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 re-selection process. This is all much more straightforward when compression-based similarity is used [1] 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 a novel featureless similarity measure can greatly simplify the design of other aspects of the system.

5.1 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 behaviour that is $O(|D||F|)$ where D is the training set and F is the set of features that describe 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 characterise 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 [8]. The computational cost of compression-based similarity depends on the compression algorithm – for text GZip is roughly linear while PPM is $O(n^2)$ [27]. 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 [1].

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-neighbour retrieval:

- **Case-Retrieval Nets:** Case-Retrieval Nets (CRNs) are perhaps the most popular technique for speeding up the retrieval process. The cases are pre-processed 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 [28, 29]. However, CRNs depend on a feature-based case representation.
- **Footprint-Based Retrieval:** As with all strategies for speeding up nearest-neighbour retrieval, Footprint-Based Retrieval involves a preprocessing stage to organise 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 which 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 speed-up and retrieval quality are nevertheless impressive [30].
- **Fish & Shrink:** This technique requires the distance to be a true metric as it exploits the triangle inequality property to produce an organisation of the case-base into candidate neighbours 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 & Shrink can be guaranteed to be equivalent to k-NN [5].
- **Cover Trees for Nearest Neighbor:** This technique might be considered the state-of-the-art in nearest-neighbour speed-up. It uses a data-structure called a Cover Tree to organise 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 speed-up 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 [6].

While CRNs have no role in systems that incorporate featureless similarity, the other three techniques are applicable and will help to make these expensive similarity measures more generally usable.

6 Conclusion

In this paper we have discussed the centrality of the feature-value representation of cases in the CBR paradigm. We have argued that in some circumstances benefits can accrue from the use of similarity measures that work more directly on the raw data. The benefit might be that the overall design of the system is simplified or it may be that the featureless metric simply offers better accuracy because it embodies specific knowledge about the data. This is very much in line with current research on SVMs 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.

Acknowledgements

The authors are grateful to Derek Bridge for helpful discussions about this work.

References

1. Delany, S., Bridge, D.: Feature-based and feature-free textual cbr: A comparison in spam filtering. In Bell, D., Milligan, P., Sage, P., eds.: Proceedings of the 17th Irish Conference on Artificial Intelligence and Cognitive Science (AICS'06). (2006) 244–253
2. Arcos, J.L., Grachten, M., de Mántaras, R.L.: Extracting performers' behaviors to annotate cases in a cbr system for musical tempo transformations. In Ashley, K.D., Bridge, D.G., eds.: ICCBR. Volume 2689 of Lecture Notes in Computer Science., Springer (2003) 20–34
3. Costello, E., Wilson, D.C.: A case-based approach to gene finding. In: Proceedings of the Fifth International Conference on Case-Based Reasoning Workshop on CBR in the Health Sciences. (2003) 19–28
4. Pekalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition. World Scientific (2006)
5. Schaaf, J.: Fish and Shrink. A Next Step Towards Efficient Case Retrieval in Large-Scale Case Bases. In Smith, I., Faltings, B., eds.: European Conference on Case-Based Reasoning (EWCBR'96, Springer (1996) 362–376
6. Beygelzimer, A., Kakade, S., Langford, J.: Cover trees for nearest neighbor. In: Proceedings of 23rd International Conference on Machine Learning (ICML 2006). (2006)
7. Kullback, S., Leibler, R.A.: On information and sufficiency. Annals of Mathematical Statistics **22** (1951) 79–86
8. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision **40** (2000) 99–121
9. Li, M., Chen, X., Li, X., Ma, B., Vitányi, P.M.B.: The similarity metric. IEEE Transactions on Information Theory **50** (2004) 3250–3264
10. Li, M., Badger, J.H., Chen, X., Kwong, S., Kearney, P.E., Zhang, H.: An information-based sequence distance and its application to whole mitochondrial genome phylogeny. Bioinformatics **17** (2001) 149–154
11. Levenshtein, V.: Binary codes capable of correcting deletions, insertions, and reversals. Problems in Information Transmission **1** (1965) 8–17
12. Saigo, H., Vert, J.P., Akutsu, T.: Optimizing amino acid substitution matrices with a local alignment kernel. BMC Bioinformatics **7** (2006) 246
13. Mallows, C.L.: A note on asymptotic joint normality. Annals of Mathematical Statistics **43** (1972) 508–515

14. Breiman, L.: Random forests. Machine Learning **45** (2001) 5–32
15. Weston, J., Leslie, C.S., Ie, E., Zhou, D., Elisseeff, A., Noble, W.S.: Semi-supervised protein classification using cluster kernels. Bioinformatics **21** (2005) 3241–3247
16. Keogh, E.J., Lonardi, S., Ratanamahatana, C.: Towards parameter-free data mining. In Kim, W., Kohavi, R., Gehrke, J., DuMouchel, W., eds.: KDD, ACM (2004) 206–215
17. Chen, X., Kwong, S., Li, M.: A compression algorithm for DNA sequences and its applications in genome comparison. Proceedings of RECOMB **107** (2000)
18. Smyth, B., Keane, M.T.: Adaptation-guided retrieval: Questioning the similarity assumption in reasoning. Artif. Intell. **102** (1998) 249–293
19. Vert, J.P., Saigo, H., Akutsu, T.: Local alignment kernels for biological sequences. In Schölkopf, B., Tsuda, K., Vert, J.P., eds.: Kernel Methods in Computational Biology. MIT Press (2004)
20. Esmeir, S., Markovitch, S.: Anytime induction of decision trees: An iterative improvement approach. In: AAAI, AAAI Press (2006)
21. Greene, D., Tsymbal, A., Bolshakova, N., Cunningham, P.: Ensemble clustering in medical diagnostics. In: CBMS, IEEE Computer Society (2004) 576–581
22. Lange, T., Roth, V., Braun, M.L., Buhmann, J.M.: Stability-based validation of clustering solutions. Neural Computation **16** (2004) 1299–1323
23. Tsymbal, A., Pechenizkiy, M., Cunningham, P.: Dynamic integration with random forests. In Fürnkranz, J., Scheffer, T., Spiliopoulou, M., eds.: ECML. Volume 4212 of Lecture Notes in Computer Science., Springer (2006) 801–808
24. Chapelle, O., Weston, J., Schölkopf, B.: Cluster kernels for semi-supervised learning. In Becker, S., Thrun, S., Obermayer, K., eds.: NIPS, MIT Press (2002) 585–592
25. Richter, M.M.: Introduction. In Lenz, M., Bartsch-Spörl, B., Burkhard, H.D., Wess, S., eds.: Case-Based Reasoning Technology. Volume 1400 of Lecture Notes in Computer Science., Springer (1998) 1–16
26. Delany, S., Cunningham, P., Smyth, B.: ECUE: A spam filter that uses machine learning to track concept drift. In Brewka, G., S, C., Perini, A., Traverso, P., eds.: 17th European Conference on Artificial Intelligence (ECAI06), IOS Press (2006) 627–631
27. Bell, T.C., Witten, I.H., Cleary, J.G.: Text Compression. Prentice Hall (1990)
28. Lenz, M., H.-D.Burkhard, Brückner, S.: Applying case retrieval nets to diagnostic tasks in technical domains. In Smith, I.F.C., Faltings, B., eds.: EWCBR. Volume 1168 of Lecture Notes in Computer Science., Springer (1996) 219–233
29. Lenz, M., Burkhard, H.D.: Case retrieval nets: Basic ideas and extensions. In: KI - Kunstliche Intelligenz. (1996) 227–239
30. Smyth, B., McKenna, E.: Footprint-based retrieval. In Althoff, K.D., Bergmann, R., Branting, K., eds.: ICCBR. Volume 1650 of Lecture Notes in Computer Science., Springer (1999) 343–357