An Effective Indexing and Retrieval Approach for Temporal Cases

David Patterson, Mykola Galushka, Niall Rooney
Northern Ireland Knowledge Engineering Laboratory (NIKEL)
Faculty of Engineering
University of Ulster
Jordanstown
N. Ireland
wd.patterson,, mg.galushka, nf.rooney @ulster.ac.uk

Abstract

In this work D-HS^T, a novel, effective indexing technique for cases with temporal attributes is presented. It is shown to be more competent, and efficient than the widely used F-Index approach based on a series of experiments using 5 synthetically generated case-bases. Additional benefits include its scalability and its suitability for indexing both temporal and non-temporal attributes alike.

Introduction

Case-based reasoning (CBR) has been successfully applied in a wide variety of domains. Most systems focus on time invariant attributes within case knowledge and either ignore temporal attributes or oversimplify them [1]. This is largely due to the difficulty of handling the CBR processes, such as, similarity determination, indexing, adaptation and knowledge maintenance satisfactorily with time related data. However time is an important and pervasive concept in the real world [2] and therefore by default, important to many of the domains CBR can be applied to. The growing importance of handling temporal data is clearly seen by observing the recent increase in the volume of research on temporal CBR (T-CBR) systems [3].

As most temporally orientated domains will likely have a mixture of both temporal and non temporal attributes (e.g. a patient profile may include both non temporal attributes such as age, weight and height and temporal attributes such as ECG and blood pressure trend) it is important that all the CBR processes can handle both types interchangeably. As such, in this paper a novel, domain independent, indexing scheme, based on a matrix structure, called the D-HS^T (Discretised Highest Similarity ^{Temporal}) is proposed, which can effectively index both temporal and non temporal attributes. As most of the research into temporal data has been carried out in the context of temporal data mining, we compare the D-HS^T index with the well known F-index approach [4] used widely within the data mining community. The F-index technique combines Discrete Fourier Transformation (DFT) [5] as a feature reduction technique with an R-tree [6] as an index. This transformation addresses the problem of the curse of dimensionality in the time domain when the number of samples is large i.e. > 25.

The focus of this paper is to investigate the effectiveness of the D-HS^T index with temporal cases. As such, we created 5 synthetic case-bases, each with one time series attribute generated using the random walk algorithm, to serve as a basis for an initial investigation into the effectiveness of the approach. A limitation of R-trees is the number of dimensions which can be effectively indexed (15-20) [4]. This was the reason that only one temporal attribute was synthetically generated as, after DFT, this creates at least 10 dimensions for the R-tree. It should be noted that an advantage of D-HS^T, as well as being able to index non temporal attributes, is that it can index a number of temporal attributes within the one case and is not limited to 1 or 2 like R-trees.

The organization of the rest of the paper is as follows. The next section outlines the methodology, which is followed by the experimental technique and results of the comparison between D-HS^T and the F-index. Finally a conclusion and future work is proposed.

Methodology

The main motivation behind this technique was to create a common indexing space for both time variant and time invariant attributes in a CBR system. A system based on a D-HS indexing structure has previously been proposed [7] for indexing cases with time invariant at-

tributes only and has shown very promising results. Here this original concept was extended for handling time-series attributes. Temporal attributes were represented by sample vectors, where each sample was defined as an instantaneous value of an observed process at a particular moment in time, *t*. DFT was applied to these attributes to transform them from a time domain into a frequency domain. A vector of complex Fourier coefficients was obtained as a result of this transformation. It has been shown that only the first few frequency coefficients need to be considered when building an index [4], as the energy of most real word signals is concentrated within these first few coefficients. These complex frequency coefficients were then split into their respective real and imaginary components. In presenting this methodology, how temporal attributes were indexed in the D-HS^T is initially described, followed by how relevant cases were retrieved from the D-HS^T in response to a query.

Indexing

A case-base *D* consisted of a set of cases *d* :

$$D = \{d_j\}_1^N, \quad (1)$$

where *N* was the number of cases in the case-base. Each case d_j consisted of a vector of problem description attributes and a solution field:

$$d_j = \left\{ d_j \in D : d_j = [x_i], \quad i = \overline{1,..,K} \right\}, \quad (2)$$

where elements $x_{1..K-1}$ were problem description attributes and x_K the solution field. Each of these attributes can either be time invariant (numeric or nominal) or temporal (time series) in nature. Due to space restrictions, only the indexing and retrieval of cases consisting of temporal attributes is presented in this work. (How time invariant attributes are managed within the D-HS framework has been demonstrated elsewhere [7]).

The D-HS^T indexing structure ***M*** consisted of a vector of matrices M_{x_i} as shown in Figure 1 and equation 3:

$$\mathbf{M} = [M_{x_i}], \quad i = \overline{1,..,(K-1)}, \quad (3)$$

where *K-1* was the number of matrices in the indexing structure (equal to the number of problem description attributes in a case d_j (2)). Each matrix M_{x_i} provided an indexing space for a related attribute x_i and as now described, actually consisted of two components.

Since the time series attributes were defined as a vector of samples, it was not possible to use them directly in the indexing process. Therefore DFT was used [5] as a dimensionality reduction technique, to transform the time series attributes into the frequency domain, where the data was represented as a vector of Ω - complex Fourier frequency coefficients.

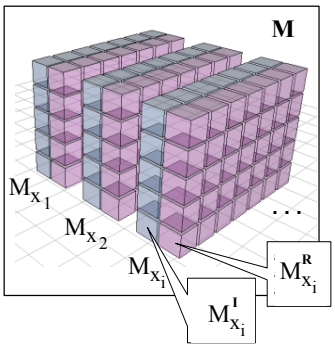

Figure 1. D-HS^T consisting of one matrix per case attribute

Based on the fact that each coefficient was complex (consisting of real and imaginary components), the matrix M_{x_i} (4) was proposed whose schematic view is also shown in detail in figure 2:

$$M_{x_i} = \left[M^R_{x_i}, M^I_{x_i} \right], \quad M^R_{x_i}, M^I_{x_i} = \left[m_{\omega\theta} \right], \quad \omega = \overline{0,..,\Omega}, \quad \theta = \overline{0,..,\Theta} \quad (4)$$

where $m_{\omega\theta}$ was a matrix cell, Θ the number of intervals into which the normalised value of the real and imaginary parts of the Fourier coefficients were split and Ω the number of Fourier coefficients taken into consideration.

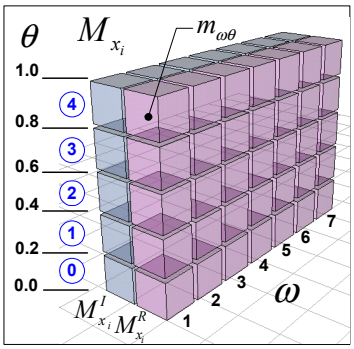

Figure 2. An individual matrix for a transformed time series attribute consisting of real and imaginary components

As can be seen each matrix consists of two components $M^R_{x_i}$ and $M^I_{x_i}$. The matrix $M^R_{x_i}$ relates to case indices constructed based on the real part of the complex Fourier coefficients and $M^I_{x_i}$ relates to case indices which were constructed based on their imaginary part.

In general the process of creating an index can be represented as a *projection* of the case-base ***D*** onto the indexing structure ***M*** (5):

$$I = proj_D \mathbf{M}, \quad (5)$$

where ***I*** is the obtained index. Since the case-base is a set of cases d_i (1) the projection operation (5) becomes:

$$proj_D \mathrm{M} = \left\{ \forall d_j : d_j \in D, \quad proj_{d_j} \mathrm{M} \right\}, \tag{6}$$

where each case is sequentially projected onto ***M***. According to (2), each case is a set of attributes so the case projection operation $proj_{d_j} \mathrm{M}$ from (6) becomes:

$$proj_{d_j} \mathrm{M} = \left\{ \forall x_i : x_i \in d_j, \quad proj_{x_i} M_{x_i} \right\}. \tag{7}$$

where each temporal attribute x_i, belonging to the case d_j, was projected onto the relevant matrix M_{x_i}. The process of indexing a temporal attribute in the D-HS^T is shown diagrammatically in figure 3.

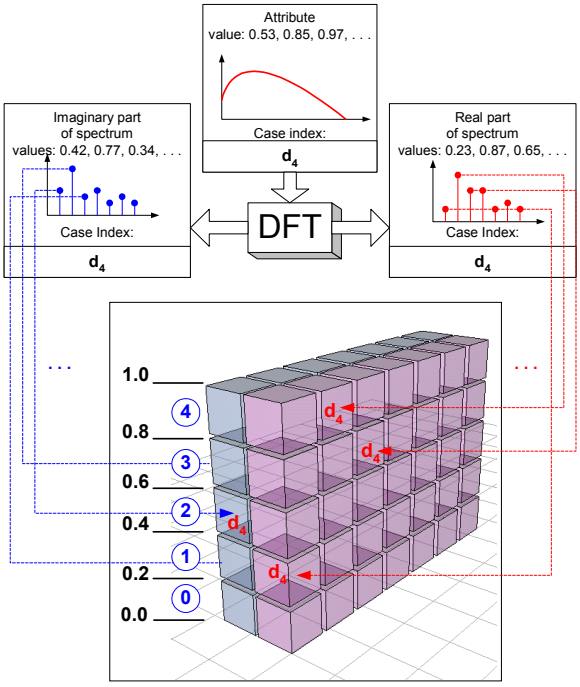

Figure 3. Indexing a temporal attribute

DFT was applied to transform x_i into the set of complex coefficients c_i:

$$c_i = dft(x_i) \tag{8}$$

The number of Fourier coefficients Ω taken into consideration in this work was 5 based on work by Rafiei [4]. However this value could fluctuate depending on the data. According to (8) the projection of a time series attribute onto the matrix M_{x_i} (7) could be rewritten as a projection of its frequency coefficients onto M_{x_i} (9):

$$proj_{d_j} \mathrm{M} = \left\{ \forall x_i : x_i \in d_j, \quad proj_{c_i} M_{x_i} \right\} \tag{9}$$

The interval index θ was calculated by the following expression:

$$\theta(v) = \begin{cases} \Theta - 1 & iff \quad v = 1 \\ \lfloor \Theta v \rfloor & iff \quad v < 1 \end{cases}, \tag{10}$$

where v was the real or imaginary part of complex coefficients and the operation $\lfloor \Theta v \rfloor$ is the integral value of the product of Θ *and* ***v***.

The process of projecting complex coefficients, obtained as a result of DFT transformation, is explained in more detail in figure 4.

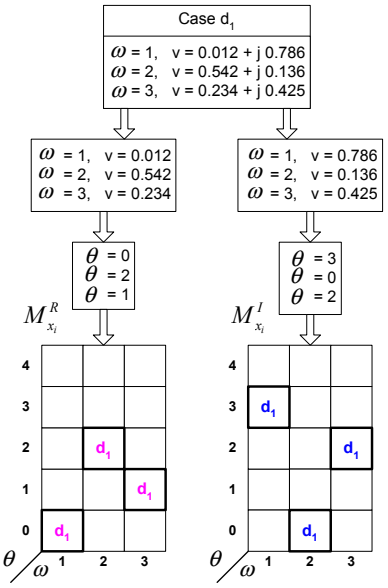

Figure 4. Projecting transformed frequency coefficients onto the indexing structure

Lets suppose that a temporal attribute x_1 belonging to a case d_1 had to be indexed in the matrix M_{x_1}. Lets suppose that only three frequency coefficients ($\Omega = 3$) were taken to consideration and the number of intervals into which the real and imaginary parts were split was 5 ($\Theta = 5$).

The first coefficient $\omega = 1$ has the value v_1= *0.012 + j 0.786*. Its real part is 0.012 and imaginary part is 0.786. According to (10), the index for the real part is calculated as $\lfloor 0.012*5 \rfloor = \lfloor 0.06 \rfloor = 0$ and for the imaginary part $\lfloor 0.786*5 \rfloor = \lfloor 3.93 \rfloor = 3$. So, the temporal attribute x_1 of the case d_1 is indexed in the matrix element $m_{1,0}$ for the real part of the first frequency coefficient and in the matrix element $m_{1,3}$ for the imaginary part of the first frequency coefficient. The second and third coefficients are indexed in ($m_{2,2}$, $m_{2,0}$) ($m_{3,1}$, $m_{3,2}$) the same way.

Retrieval

Having described how cases with time series attributes were indexed, their retrieval in response to a query

is now explained. DFT transformation (Figure 5A) was applied to the temporal attribute of the target case d^t. The frequency coefficients obtained were split into real and imaginary parts as before. Both parts were then used to extract relevant cases from the D-HS^T (5 B). Extraction of relevant cases involved matching individual frequency coefficients (real and imaginary) from the query case with those already in the D-HS^T. If a query case coefficient and an indexed case coefficient fell into the same cell of the indexing structure, then they were deemed to initially intersect and the case was added to what was known as the initial retrieval set (5 C). The final retrieval set of case indices D^t was obtained, based on a weighted vote of all intersecting coefficients of the cases in the initial retrieval set (5 D). A diagrammatic representation of extraction can be seen in figure 5 and a formal definition of this process is provided in (11):

$$D^t = proj_{d^t} I \,. \qquad (11)$$

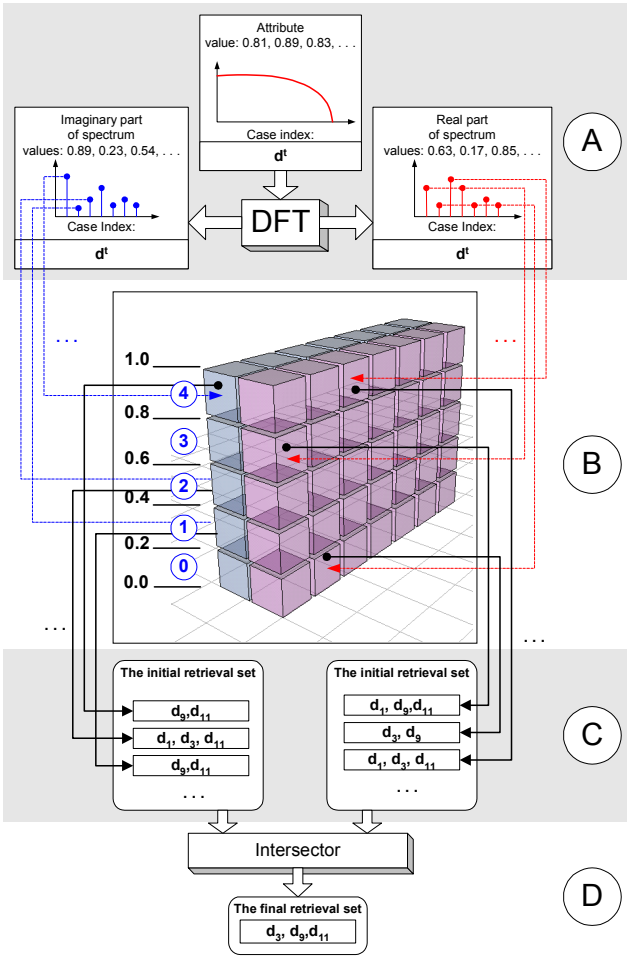

Figure 5. Case retrieval from the indexing structure

A worked example of the retrieval process is shown in figure 6.

Lets suppose that *after* DFT a target case produces 3 frequency coefficients corresponding to the real Matrix elements $m_{1,0}$, $m_{2,2}$, $m_{3,3}$ and the imaginary Matrix elements $m_{1,3}$, $m_{2,4}$, $m_{3,1}$ as indicated by the cells in bold in Figure 6 part A. The set of cases therefore extracted (retrieved) for the initial retrieval set (6 B) from the matrix $M^R_{x_1}$ consists of d_1, d_2, d_3, d_4, d_5 and the set of cases retrieved from the matrix $M^I_{x_1}$ is d_1, d_3, d_4, d_5.

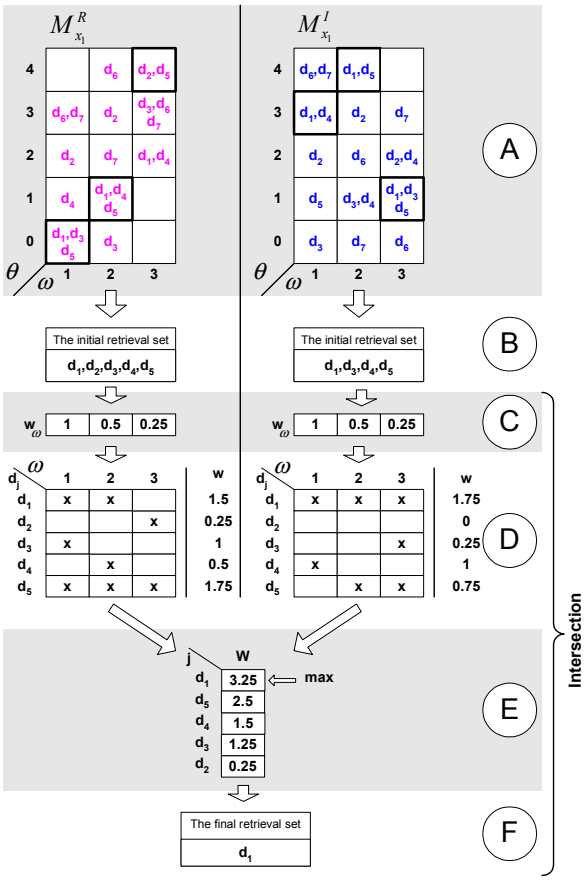

Figure 6. An example of retrieval

Similarity determination between a query and the cases in the case-base is determined as a product of the number of attribute Frequency coefficients that match between the query and retrieved cases and the relative importance of these matching coefficients. Therefore attribute Frequency coefficients were weighted in order of their importance (1, 0.5 and 0.25 in Figure 6). This ensures that a match between the query case and retrieved cases on the first Frequency coefficient of an attribute has more influence on similarity determination than a match in subsequent coefficients (6 C). Weighting was done in this way, as it is known that the initial frequencies have the most importance in reconstructing

the original time series data from the transformed frequency domain.

Therefore in order to determine similarity in the example, for each matching case in the initial retrieval set, an overall similarity score must be determined based on their matching attribute frequency coefficients and corresponding weights (6 D). For example case d_1 matches on the first 2 frequency components for $M^R_{x_1}$. These correspond to weights of 1 and 0.5 respectively, giving an overall similarity score of 1.5 from $M^R_{x_1}$. It matched on all 3 frequency components for $M^I_{x_1}$, giving a similarity of (1.*1) + (1*0.5) + (1*0.25) = 1.75. The overall similarity score for d_1, 3.25, is determined by summing the similarity scores for the real and imaginary components. When the similarity scores of all cases in the initial retrieval set are calculated they are ranked in order of similarity (6 E). The N top cases are selected to form the final retrieval set (6F). In the example in Figure.6 N=1.

Experimental Technique and Results

The technique has been tested successfully on both real world and synthetic case-bases but due to space limitations here we demonstrate the efficacy of the technique based on five synthetically generated case-bases of different sizes (10^2, 10^3, $5*10^3$, 10^4, $5*10^4$ cases). Each case had one temporal attribute, generated by the random walk algorithm. Each temporal attribute contained 100 samples. The case bases were split into training (9/10 used to create index) and test (1/10) sets. Ten-fold cross validation was carried out and the mean absolute distance (MAD) and efficiency for D-HS^T and F-index obtained. For evaluation purposes we converted the retrieved case back into the time domain and calculated the MAD in the following way. The absolute distance (AD) was calculated between the temporal attribute in the target case and selected case by the following expression $AD = \sqrt{\sum_{l=1}^{L} (x_t[l] - x_s[l])^2}$, where L was a number of attribute samples, $x_t[l]$ the attribute's sample l of the target case and $x_s[l]$ the attribute's sample l of the selected case. In these experiments $\Theta = 10$ and $\Omega = 5$.

Figure 7 shows how the MAD varied for both the D-HS^T and F-index as the number of cases in the case-base was increased from 100 to 50000. Results shown are an average for the 5 case-bases investigated. From this it can be seen that when there are relatively few cases in the case-base both techniques perform comparatively producing a MAD of around 0.45. When the number of cases is increased to 1000 both techniques improve in competency with D-HS^T providing a more substantial increase (MAE of 0.27) over the F-index (MAD of 0.33). The addition of more cases up to 50000, has little effect on the competency of the F-index as it stabilises around a MAD of 0.33, whereas the D-HS^T continues to improve in competency in a linear fashion as more cases are added, to provide a MAD of 0.19 when the size of the case-base is 50000 (42% more competent).

Additionally from the gradient of the MAD line the competency of the D-HS^T should improve even further with the addition of more cases thus showing it to be a scaleable technique with can be used with very large case-bases.

Figure 8 shows the average efficiency of the F-index, the D-HS^T with $\Theta = 10$ and $\Theta = 50$ as the number of cases is increased. Recorded efficiencies include the time to create the index from the raw data in the case-base and carry out retrieval using 10 fold cross validation.

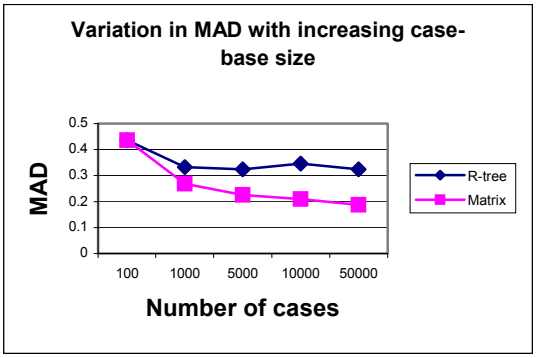

Figure 7 Variation in MAD with increasing case-base size

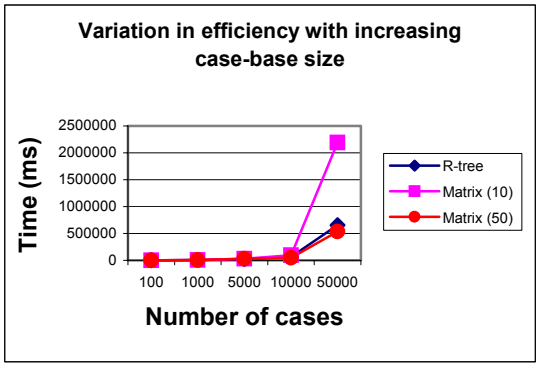

Figure 8 Variation in efficiency with increasing case-base size

From this it can be seen that both the F-index and D-HS^T(10) are comparable in terms of efficiency when the case-base size is less than 10000. As the case-base grows from this point, the efficiency of D-HS^T markedly deteriorates in comparison to F-index. The reason for this is simple. The least efficient part of retrieval is the extraction from D-HS^T of cases for the initial retrieval set. There is a linear relationship between the time taken to determine the final retrieval set and the number of cases in the initial retrieval set. As the case-base grows the computation required to identify cases for the initial retrieval set increases substantially, thus slowing the process down. In order to

investigate this further an additional experiment was carried out using the 5 case-bases consisting of 50000 cases each. In this experiment the number of intervals in D-HS^T Θ was varied between 10 and 100 and the effects on MAD and efficiency noted. By increasing the number of intervals the density of D-HS^T cells was reduced, which in turn reduced the number of cases in the initial retrieval set thus making the process more efficient. Results can be seen in Figure 9 where the ratios are shown compared to D-HS^T with 10 intervals (the original experiment) for both MAD and efficiency as the number of D-HS^T intervals increases. (Values >1 indicate that D-HS^T with 10 intervals is superior).

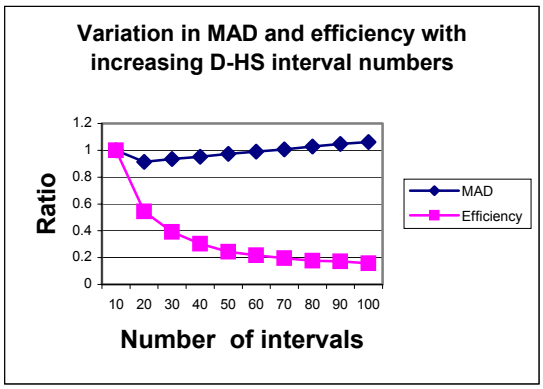

Figure 9 Variation in MAE and efficiency with increasing D-HS^T interval numbers

From this it can be seen that by increasing the number of intervals from 10 to 60 improves efficiency by almost 5 fold. Increasing the number of intervals also has an effect on the MAD, where it initially improves with 20 intervals but rises steadily from here to a point between 60 and 70 intervals where the MAD is comparable to 10 intervals. After this the MAD is worse than with 10 intervals indicating that the D-HS^T cells are becoming too sparsely populated and relevant cases are not being retrieved in response to queries. Therefore it can be seen that there is a trade off between efficiency and competency with around 50 intervals providing the optimal number in terms of efficiency and competency for the case-bases investigated in this study. At 50 intervals D-HS^T is still producing a much more competent indexing and retrieval scheme compared to the F-index (43.8% more competent) and at this point it is also 18.6% more efficient, as can be seen from Figure 8.

Conclusions

It is proposed that the D-HS^T is an ideal approach for indexing and retrieving temporal cases. When compared to the commonly used approach of F-index it was seen to be as competent for small case-bases (100 cases) but up to 42% more competent for larger case-bases (50000). Its efficiency was also seen to be superior for case-bases of less than 10000. Efficiency deteriorated after this point due to the number of intervals in D-HS^T being too small (10). Once this value was increased the efficiency improved greatly to a point at around 50 intervals where efficiency improved almost 5 fold for case-bases of 50000 cases. At this point it outperformed the F-index competency by a factor of 44% and was18.6% more efficient.

Although here we have only demonstrated the results of initial experiments on 1 temporal attribute (due to facilitating a comparison to F-index) and shown D-HS^T to be more competent, efficient and scalable, we have also found it to be similarly effective with cases consisting of numerous temporal cases attributes (results not shown due to space). Therefore we believe it is more scalable in terms of the number of attributes than F-index whose effectiveness deteriorates when the dimensionality increases beyond 15-20 (note 1 temporal attribute = 10 dimensions). Additionally another attractive benefit of this approach is that it can be used to index and retrieve hybrid cases consisting of temporal and non temporal attributes within the same indexing framework. Future work includes investigating, intelligently determining the optimal number of intervals for D-HS^T based on the size of the case-base and determining optimal weights for the frequency coefficients.

References

[1] Dørum, M., Aamodt, A. and Skalle, P. 2002. Representing Temporal Knowledge for Case-Based Prediction. Advances in case-based reasoning; 6th European Conference, 174-188. Aberdeen, LNAI 2416, Springer.

[2] Combi, C. and Shahar, Y. Temporal reasoning and temporal data maintenance in medicine: issues and challenges. Computers in Biology and Medicine (in press).

[3] Workshop "*Applying CBR to Time Series Prediction*". 2003. Int'l. Conference. On CBR, 319-228.

[4] Rafiei, D. and Mendelzon, A. 1998. Efficient retrieval of similar time sequences using DFT. In Proceedings of the 5th Intl. Conf. on Found. of Data Org. and Alg. (FODO '98), Kobe, Japan.

[5] Oppenheim, A. and Schafer, R. 1975. Digital Signal Processing. Prentice-Hall, Englewood Cliffs, N.J.

[6] Guttman, A. 1984. R-trees: a dynamic index structure for spatial searching. In Proc. ACM SIGMOD Int. Conf. on Management of Data, 47-57. Boston, MA.

[7] Patterson, D., Rooney, N. & Galushka, M. 2002. Efficient Similarity Determination and Case Construction Techniques For Case-Based Reasoning. Proceedings of the 4th European Conference on Case-Based Reasoning (ECCBR-02), 292-305. Aberdeen, Scotland.