Time-Series Similarity Queries Employing a Feature-Based Approach

R. J. ALCOCK and Y. MANOLOPOULOS

Data Engineering Laboratory,
Department of Informatics,
Aristotle University of Thessaloniki,
54006, Thessaloniki,
GREECE.

1. SUMMARY

Time-series, or time-sequence, data show the value of a parameter over time. A common query with time-series data is to find all sequences which are similar to a given sequence. The most common technique for evaluating similarity between two sequences involves calculating the Euclidean distance between them. However, many examples can be given where two similar sequences are separated by a large Euclidean distance. In this paper, instead of calculating the Euclidean distance directly between two sequences, the sequences are transformed into a feature vector and the Euclidean distance between the feature vectors is then calculated. Results show that this approach is superior for finding similar sequences.

2. INTRODUCTION

Time-series data are sequences showing the value of a parameter, or sometimes parameters, over time. Important time series include stock market prices, interest rates, sales of a product, scientific results, weather readings and medical records. A common query with time-series data is to find all sequences which are similar to a given sequence. In formal terms, given a time series r_i of length n ($1 \le i \le n$), it is required to find all stored time series which are similar to r_i. The problem can be extrapolated to two dimensions, where it is required to find all stored images similar to a given image [1]. When similar sequences have been found, it is then possible to use these sequences to facilitate tasks such as prediction.

In time-series literature, the most common technique for evaluating similarity between two sequences is to calculate the Euclidean distance between them and if this is below a given threshold ε, then the two sequences are said to be similar [2]. The distance D between two sequences r_i and s_i, both of length n, can be calculated by:

$$D = \sum_{i=1}^{n} ((r_i - s_i)^p)^{1/q} \qquad (1)$$

When p=q=2, formula (1) represents the Euclidean distance. Given this definition of similarity, most research in the area has concentrated upon finding similar sequences as quickly as possible.

Whilst it is important to find similar time sequences quickly, it is vital that the found sequences are in fact similar. Many examples can be given where two similar sequences are not close using the Euclidean distance metric. A simple example would be a pattern involving an upward shift at time t_1 (Fig. 1). If this is compared with a second time series with a shift at a later time t_2 ($t_2 >> t_1$) then the Euclidean distance depends upon how much larger t_2 is than t_1. However, the two sequences should be found to be similar, irrelevant of where the step occurs. Many other examples have been given where similar sequences are separated by a large Euclidean distance [3, 4, 5].

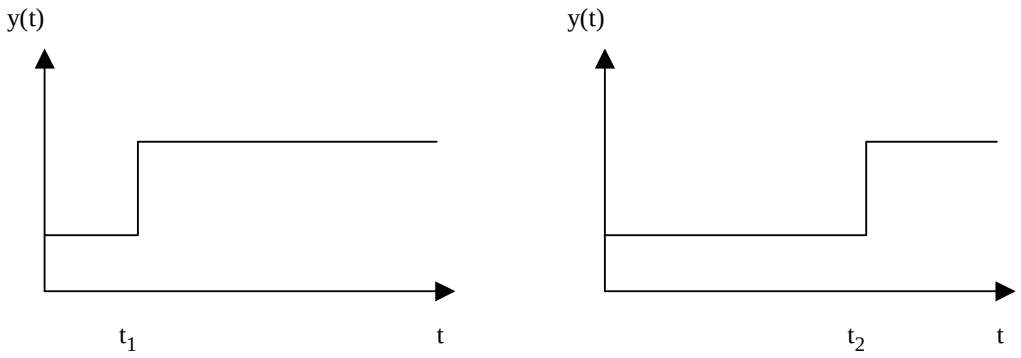

Fig. 1 Similar Sequences Separated by a Large Euclidean Distance

Agrawal et al. [2] proposed that signals could be transformed into the Fourier space using the Discrete Fourier Transform (DFT). As the first few Fourier co-efficients contain most of the information, a signal can be compressed by using just the first few co-efficients to store it. In later work by Rafiei and Mendelzon [6], it was proposed to utilise also the last few Fourier co-efficients. According to Parseval's theorem, the Euclidean distance between two signals is preserved in the Fourier space. Therefore, whilst the DFT is very useful for compression and storage of signals, for time-series similarity, it gives the same results as employing the original signal.

To overcome the shortcomings of using Euclidean distance for similarity queries, many researchers have advocated that the query sequence should be transformed before applying the Euclidean distance [7-11]. Such transformations include shifts (vertical and horizontal), scaling (linear and amplitude) and removal of non-matching parts. However, scaling could be of any real-valued size and shifts could occur at any time during the sequence. Also, non-matching parts could appear at any point in the sequence and be of various sizes. Therefore, the number of transformations that a sequence could undergo is infinite and searching for the correct sequence of transformations would face the combinatorial explosion problem.

Another approach gives two criteria that must be met for two sequences to be considered similar [4]. The first is the Euclidean distance and the second is that the distance between two corresponding points in the two sequences cannot be larger than a given threshold. The problem

is that the two sequences could be similar except for one erroneous point that differs widely between the two sequences. Thus, the two sequences would be found to be dissimilar.

It is proposed here that, instead of calculating the Euclidean distance between two sequences, the sequences are transformed into feature vectors and the Euclidean distance is then calculated between these vectors. If the dimension of the feature vectors is small, the features can be employed as a signature for indexing [12].

3. PROBLEM DEFINITION

One major problem in assessing the effectiveness of techniques for similarity queries is that there is no mathematical definition for similarity. Whilst it is trivial to model equality mathematically, by its nature, similarity cannot be defined exactly. Therefore, if an algorithm finds that two sequences are similar, normally a human observer is required to verify this. Even between different humans, the idea of similarity varies. Thus, this paper proposes the use of time-series data called control chart patterns, which do have some quantifiable similarity [3, 13]. Control chart patterns are time series that show the level of a machine parameter plotted against time. The control chart patterns described in [3, 13] are artificially generated by six equations. Each equation represents a different type of pattern. Two patterns can be considered similar if they are generated by the same equation. The six pattern types are illustrated in Figs 2 to 7 and are described as normal, cyclic, increasing trend, decreasing trend, upward shift and downward shift. The equations used here are based on those given in [3] and are as given below. Each time sequence is of length n and is represented by an array of values y(t), where $1 \leq t \leq n$.

1. Normal pattern: $y(t) = m + rs$ (2)

Where m = 30, s= 2 and r is a random number between ± 3

2. Cyclic pattern: $y(t) = m + rs + aSIN(2\pi t/T)$ (3)

Where a and T take values between 10 and 15 for each pattern

3. Increasing shift: $y(t) = m + rs + gt$ (4)

Where g takes a value between 0.2 and 0.5 for each pattern

4. Decreasing shift: $y(t) = m + rs - gt$ (5)

5. Upward shift: $y(t) = m + rs + kx$ (6)

Where, for each pattern, x takes a value between 7.5 and 20. k = 0 before time t_3 and 1 after this time. t_3 takes a value between n/3 and 2n/3 for each pattern

6. Downward shift: $y(t) = m + rs - kx$ (7)

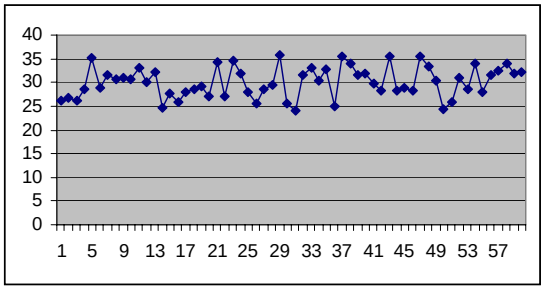

Fig. 2 Example of a Normal Pattern

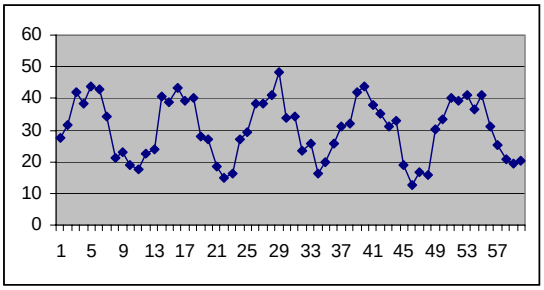

Fig. 3 Example of a Cyclic Pattern

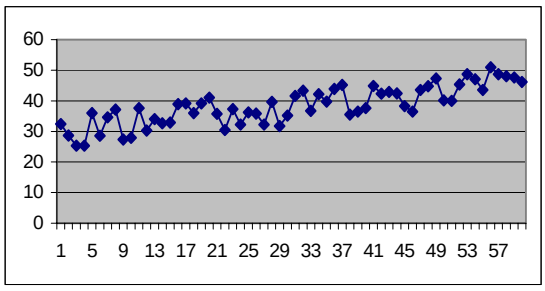

Fig. 4 Example of an Increasing Trend

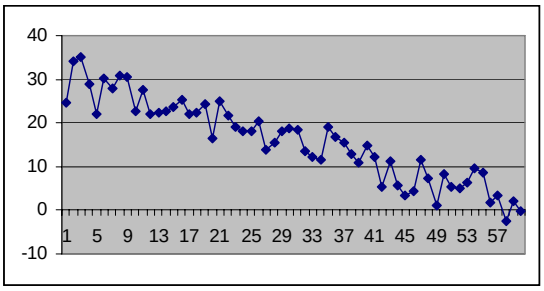

Fig. 5 Example of a Decreasing Trend

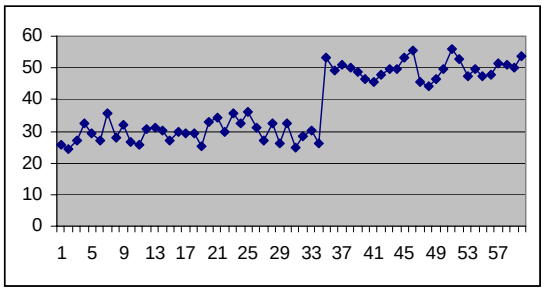

Fig. 6 Example of an Upward Shift

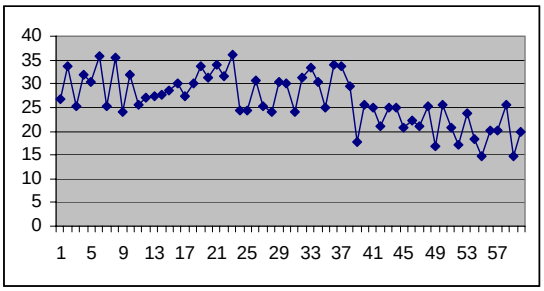

Fig. 7 Example of a Downward Shift

4. FEATURES

As previously mentioned, it is proposed here that a set of features, called a feature vector, is calculated from each time series. The feature vectors are used by the distance function to determine similarity. The motivation to employ features comes from the field of image processing [14].

Many different features have been employed in image processing to classify objects. Features that describe the intensity, or brightness, of the pixels (picture elements) which constitute an object can be divided into first-order features and second-order features. First-order features are based on the actual intensity of pixels and second-order features are based on the difference in intensities of nearby pixels.

4.1 First-Order Features

The most commonly-used first-order features are the statistical features, mean (μ), standard deviation (σ), skewness (SKEW) and kurtosis (KURT). The equations for these are:

$$\mu = \frac{\sum_{t=1}^{n} y(t)}{n} \tag{8}$$

$$\sigma = \sqrt{\frac{\sum_{t=1}^{n} (y(t) - \mu)}{n}} \tag{9}$$

$$SKEW = \frac{\sum_{t=1}^{n} (y(t) - \mu)^3}{n\sigma^3} \tag{10}$$

$$KURT = \frac{\sum_{t=1}^{n} (y(t) - \mu)^4}{n\sigma^4} - 3 \tag{11}$$

4.2 Second-Order Features

The most common second-order features in image processing are co-occurrence features which have been shown to perform better than other second-order features [14]. Here, their description relates to how co-occurrence features can be calculated for a one-dimension time series. First, the data is quantised into Q levels. For example, if the sequence 1, 2, 3, 4 is quantised into 2 levels (Q=2) then it becomes 1, 1, 2, 2. Second, a two dimensional matrix c(i, j) is constructed ($1 \leq i,j \leq Q$). Point (i, j) in the matrix represents the number of times that a point in the sequence with level i is followed, at a distance d_1, by a point with level j. Finally, five co-occurrence features (energy, entropy, correlation (COR), inertia and local homogeneity (LH)) are calculated using the following equations:

$$Energy = \sum_{i=1}^{n} \sum_{j=1}^{n} c(i, j)^2 \tag{12}$$

$$Entropy = \sum_{i=1}^{n} \sum_{j=1}^{n} c(i, j).\log(c(i, j)) \tag{13}$$

$$COR = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} (i - \mu_x)(j - \mu_y) c(i, j)}{\sigma_x \sigma_y} \tag{14}$$

Where:

$$\mu_x = \frac{\sum_{i=1}^{n} i \sum_{j=1}^{n} c(i,j)}{n} \tag{15}$$

$$\mu_y = \frac{\sum_{j=1}^{n} j \sum_{i=1}^{n} c(i,j)}{n} \tag{16}$$

$$\sigma_x^2 = \frac{\sum_{i=1}^{n} (i-\mu_x)^2 \sum_{j=1}^{n} c(i,j)}{n} \tag{17}$$

$$\sigma_y^2 = \frac{\sum_{j=1}^{n} (j-\mu_y)^2 \sum_{i=1}^{n} c(i,j)}{n} \tag{18}$$

$$Inertia = \sum_{i=1}^{n} \sum_{j=1}^{n} (i-j)^2 c(i,j) \tag{19}$$

$$LH = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{1+(i-j)^2} c(i,j) \tag{20}$$

4.3 New Second-Order Features

New second-order features for time-series data are proposed here. These features have the advantages that they do not require quantisation and are relatively simple to calculate. First, a one-dimensional array is calculated called D(t):

$$D(t) = y(t+d_2) - y(t) \quad 1 \le t \le n-d_2 \tag{21}$$

Where d_2 is the distance between the points being compared. Second, from this array, four statistical features are calculated: mean, standard deviation, skewness and kurtosis. These will be referred to as μ_2, σ_2, SKEW2 and KURT2.

5. RESULTS

For the experiments, 600 patterns were generated, 100 of each type with n equal to 60 as in [3, 13]. Each pattern was taken in turn and compared against the other 599. Features were extracted from the patterns and then scaled between 0 and 1 because each of the features has a different range. Then, the Euclidean distances between the feature vectors were calculated. The performance measure employed was:

$$\text{Performance} = (T - F) / M \tag{22}$$

where T is the number of patterns found to be similar which actually are similar, F is the number of patterns incorrectly found to be similar and M is the maximum value which T can take.

Therefore, optimal performance is achieved when T equals M and F is zero. In this case, the overall performance is 1. As it is not yet known how to determine ε optimally, a global search was performed for the best value to use simultaneously for all 600 patterns.

Three sets of experiments were carried out. First, the performance was determined of simply calculating the Euclidean distance between the original time series. Second, the Euclidean distance between the first-order and co-occurrence features was calculated. Third, the first-order and new second-order features were employed to differentiate the signals. Each experiment was run three times and the average performance taken.

The first set of experiments, calculating the Euclidean distance between the original signals, gave a performance of 0.371. This confirms that the approach of determining sequence similarity based on the pure Euclidean distance between sequences is far from perfect.

The results of the second set of experiments are shown in Table 1. The best performance was achieved when the first and second-order features were employed together with parameter values Q=3 and D_1=1. The effect of varying Q and D_1 can also be seen in Table 1. The best performance obtained was 0.452, which shows that the feature-based approach is better than the simple Euclidean distance approach. However, the co-occurrence features, when employed without the first-order features, gave a poor performance (0.173). Simple visual analysis of the feature values showed that skewness might not be very useful in discriminating between the patterns. Indeed, the results show that skewness adds very little extra benefit when employed with the other seven features.

Features employed									Parameters for second order features		
μ	σ	Skew	Kurt	Energy	Entropy	COR	Inertia	LH	Q	D_1	Average performance (3 runs)
√	√	√	√	√	√	√	√	√	3	1	0.452
√	√		√	√	√	√	√	√	3	1	0.451
√	√		√						-	-	0.422
√	√								-	-	0.393
√	√	√	√						-	-	0.371
√	√	√	√	√	√	√	√	√	4	1	0.358
				√	√	√	√	√	3	1	0.173
				√	√	√	√	√	4	1	0.158
				√	√	√	√	√	3	2	0.081
				√	√	√	√	√	8	1	0.000

Table 1 Experiments Using Co-occurrence Features

Table 2 shows the performance of the third set of experiments. As with the second set of experiments, the best performance (0.473) was achieved when the first and second-order features were employed together. The main conclusion when comparing Tables 1 and 2 is that the second-order features proposed in this paper perform better than co-occurrence features for time-series similarity queries. When the new second-order features are employed without the first-order features, the performance (0.463) is higher than any result achieved with the co-occurrence features. It can be seen from Table 2 that KURT2 adds little extra benefit when added to the

other seven features. Also, μ_2 is a vital second-order feature, as the other three second-order features alone cannot discriminate between the signals at all.

Features employed									
μ	σ	Skew	Kurt	$\mu 2$	$\sigma 2$	Skew2	Kurt2	D_2	Average performance (3 runs)
√	√	√	√	√	√	√	√	7	0.473
√	√	√	√	√	√	√		7	0.471
				√	√	√		7	0.463
				√	√	√	√	7	0.428
				√	√		√	7	0.417
				√	√	√	√	5	0.414
				√	√	√	√	9	0.375
				√	√	√	√	3	0.321
				√		√	√	7	0.279
				√	√	√	√	1	0.049
					√	√	√	7	0.000

Table 2 Experiments Using the New Second-Order Features

6. CONCLUSION

Most previous work in time-series similarity analysis has used the Euclidean distance between the signals as the basis for similarity. However, many cases can be found where similar sequences are separated by a large Euclidean distance. This paper has adopted a feature-based approach to similarity queries. Common features employed in image processing were utilised as the basis for the experiments. These were first-order statistical features and second-order co-occurrence features. This feature-based approach gave a superior performance than the traditional approach. Also, new second-order features were developed and it was found these gave a better performance than co-occurrence features.

Future work could be performed in many areas to improve upon the achieved performance. First, simple preprocessing could be tried on the signals, such as scaling and smoothing. Second, new features could be developed and the optimum features determined to employ in similarity queries. Third, different distance functions could be tested. A possibility would be to utilise different values for p and q in formula (1). Fourth, experiments could be carried out to determine automatically the optimal threshold for ε. Fifth, work could be performed on the matching of subsequences, as proposed in [15]. Finally, the developed techniques should be used on real data sequences. Any real patterns used need to be put manually into type categories to enable the performance measure to be calculated. Patterns which could be employed include stock market prices and electrocardiograms (ECGs) [16].

ACKNOWLEDGEMENT

The authors would like to thank the European Commission for funding the stay of Robert Alcock at Aristotle University under the Chorochronos project (EC ref: FMRX960056).

REFERENCES

[1] R. Weber, H.J. Schek and S. Blott, A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. Proc. 24th Int. Conf. on Very Large Databases (VLDB), New York. pp. 194 – 205. (1998)

[2] R. Agrawal, C. Faloutsos and A. Swami, Efficient Similarity Search in Sequence Databases. Proc. 4th Int. Conf. on Foundations of Data Organization and Algorithms, Chicago. pp. 69 - 84. (Also in Lecture Notes in Computer Science 730, Springer Verlag). (1993)

[3] D.T. Pham and A.B. Chan, Control Chart Pattern Recognition Using a New Type of Self-Organising Neural Network. Proc. Instn. Mech. Engrs. Vol. 212, No. 1, pp. 115 – 127. (1998)

[4] S.K. Lam and M.H. Wong, A Fast Projection Algorithm for Sequence Data Searching. Data and Knowledge Engineering. Vol. 28, No. 3, pp. 321 - 339. (1998)

[5] D. Rafiei, On Similarity-Based Queries for Time Series Data. Proc. 15th Int. Conf. on Data Engineering. Sydney, Australia. pp. 410 – 417. (1999)

[6] D. Rafiei and A.O. Mendelzon, Efficient Retrieval of Similar Time Sequences Using DFT. Proc. Int. Conf. on Foundations of Data Organisations and Algorithms. Kobe, Japan. (1998)

[7] R. Agrawal, K. Lin, H.S. Sawhney and K. Shim, Fast Similarity Search in the Presence of Noise, Scaling, and Translation in Time-Series Databases. Proc. 21st Int. Conf. on Very Large Databases, Zurich, Switzerland. pp. 490 – 501. (1995)

[8] D.Q. Goldin and P.C. Kanellakis, On Similarity Queries for Time-Series Data: Constraint Specification and Implementation. Proc. 1st Int. Conf. on Principles and Practice of Constraint Programming. Cassis, France. pp. 137-153. (1995)

[9] H.V. Jagadish, A.O. Medelzon and T. Milo, Similarity-Based Queries. Symposium on Principles of Database Systems. San Jose, California. pp. 36 – 45. (1995)

[10] G. Das, D. Gunopulos and H. Mannila, Finding Similar Time Series. Proc. 1st European Symposium on Principles of Data Mining and Knowledge Discovery. Trondheim, Norway. pp. 88 – 100. (1997)

[11] C. Faloutsos, H.V. Jagadish, A.O. Mendelzon and T. Milo, Signature Technique for Similarity-Based Queries. SEQUENCES 97, Positano-Salerno, Italy. (1997)

[12] H. Andre-Jonsson and D.Z. Badal, Using Signature Files for Querying Time-Series Data. Proc. 1st European Symposium on Principles of Data Mining and Knowledge Discovery. Trondheim, Norway. pp. 211 – 220. (1997)

[13] D.T. Pham and E. Oztemel, Control Chart Pattern Recognition Using Learning Vector Quantization Networks. Int. J. Prod. Res, Vol. 32, No. 3, pp. 721-729. (1994)

[14] R.J. Alcock, Techniques for Automated Visual Inspection of Birch Wood Boards. PhD thesis, School of Engineering, Cardiff University, UK. (1996)

[15] C. Faloutsos, M. Ranganathan and Y. Manolopoulos, Fast Subsequence Matching in Time-Series Databases. Proc. ACM SIGMOD Int. Conf. on Management of Data. Minneapolis, Minnesota. pp. 419 – 429. (1994)

[16] B.K. Yi, H.V. Jagadish and C. Faloutsos, Efficient Retrieval of Similar Time Sequences Under Time Warping. Proc. 14th Int. Conf. on Data Engineering. pp. 201 – 208. (1998)