General Hierarchical Model (GHM) to Measure Similarity of Time Series*

Xinqiang Zuo
School of Software
Tsinghua University, Beijing, 100084, China
zuoxq04@mails.tsinghua.edu.cn

Xiaoming Jin
School of Software
Tsinghua University, Beijing, 100084, China
xmjin@mail.tsinghua.edu.cn

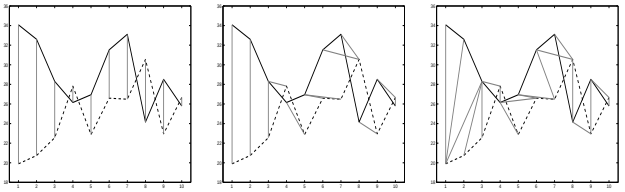
Abstract

Similarity query is a frequent subroutine in time series database to find the similar time series of the given one. In this process, similarity measure plays a very important part. The previous methods do not consider the relation between point correspondences and the importance (role) of the points on the content of time series during measuring similarity, resulting in their low accuracies in many real applications. In the paper, we propose a General Hierarchical Model (GHM), which determines the point correspondences by the hierarchies of points. It partitions the points of time series into different hierarchies, and then the points are restricted to be compared with the ones in the same hierarchy. The practical methods can be implemented based on the model with any real requirements, e.g. FFT Hierarchical Measures (FHM) given in this paper. And the hierarchical filtering methods of GHM are provided for range and *k*-NN queries respectively. Finally, two common data sets were used in *k*-NN query and clustering experiments to test the effectiveness of our approach and others. The time performance comparisons of all the tested methods were performed using the synthetic data set with various sizes. The experimental results show the superiority of our approach over the competitors. And we also give the experimental powers of the filtering methods proposed in the queries.

1 Introduction

In the real-world, time series appears in various applications, e.g. stock market, medicine field, network etc. Unlike exact match in the traditional database, similarity query is frequent in time series database to find the similar time series of the given one. It has attracted a lot of interests to design an effective and efficient similarity measure [1, 2, 3, 4, 5].

Most of the methods compute the sum of the distances between the values on the points of time series as the measuring result. The most popular ones are L_p norms, especially L_2 (Euclidean Distance), which follow the triangle inequality, and have many dimensionality reduction approximations and index methods for fast query [6, 7, 8]. But they are demonstrated to be very brittle for many applications, because they do not support local time shifting in the measuring process. They cannot accommodate the time series that are similar but out of phase, which exist widely in the real-world applications. Other methods have been proposed based on different requirements to solve the problem, e.g. DTW, LCSS, EDR etc. They allow time warping in measuring, as shown in Figure 1. They always determine the point correspondences to minimize the warping cost, only with different cost calculations. However, they do not consider the relation between point correspondences and the importance (role) of the points on the content of time series.



(a) Euclidean (b) DTW (c) LCSS

Figure 1. The point correspondences of Euclidean Distance, DTW, LCSS.

The influences of points on the content of time series, which decides the similarity, are different, so the points in time series should play different roles in similarity measure. Based on this idea, in this paper, we propose a General Hierarchical Model (GHM), which determines the point correspondences in measuring similarity by the hierarchies of points. After giving the hierarchies of points, we put the comparative restriction into the measuring that the points should be compared with the ones in the same hierarchy. Any hierarchical strategies can be adopted based on the actual analysis. In this paper, two practical measures are also given to realize GHM, termed FFT Hierarchical Measures (FHM), which use Fast Fourier Transform (FFT) to partition the points. And the hierarchical filtering methods of GHM are provided for range and *k*-NN queries respectively. In the experimental evaluation, two common synthetic and real data sets were used in *k*-NN query and clustering experiments to test the effectiveness of ours and other major competitors. The time performances of all the methods were compared on the synthetic data set with various sizes. The experimental results show the superiority of our approach over others in terms of effectiveness and efficiency. And finally, we also tested the powers of the filtering methods proposed on the real data set with various parameters.

The rest of the paper is organized as follows. Sect 2 provides the related work. Sect 3 introduces GHM and the instance FHM. In

*The work was supported by the NSFC 60403021 and the 973 Program 2004CB719400.

Sect 4, we give the exhaustive performance tests of ours and other methods in terms of effectiveness and efficiency. Finally, in Sect 5 we offer some conclusion remarks.

2 Related Work

L_p norms are popular [6, 7, 8], but with the precondition of the same length. They are simple and fast, but they cannot deal with local time shifting, which is an important evaluation for the feasibilities of measures in real applications. Berndt et al. [1] adopted DTW to measure the similarity of time series, which allows an elastic shifting, but it is sensitive to noise due to its continuous warping path. LCSS is robust to noise, but inaccurate with different gaps existing between similar shapes in time series [2]. Recently, Chen et al. proposed Edit distance with Real Penalty (ERP) and Edit Distance on Real sequence (EDR) in [4, 5] respectively for multi-dimensional data. ERP is also sensitive to noise, and EDR follows a near triangle inequality, which are discussed in [5]. DTW, LCSS, ERP and EDR release the strict restriction of point correspondences in L_p norms, but their corresponding strategies, i.e. minimizing the warping cost, do not take the importance (role) of each point on the content of time series into account, resulting in low accuracy.

To solve the performance problems of the direct distance functions, including effectiveness and efficiency, many representation methods were proposed. Das et al. [9] gave a simple representation that automatically clusters all the subseries in a fixed-window into some classes, and then uses the symbols standing for the classes to replace each subseries. The method may be disabled due to the inaccuracy of the interval boundaries, e.g. a whole shape (or content) might be segmented. In [10], Lin et al. proposed the Symbolic Approximation (SAX) with an approximate distance function that lower bounds the Euclidean Distance. Clipped representation has attracted much interest [11], and it has superior space benefit due to only saving 0 and 1. Recently, in [12], Megalooikonomou et al. adopted a multiresolution symbolic representation, and Hierarchical Histogram Model was used as the distance function. The multiresolution seems similar to our approach, but actually, we have the essential difference that we emphasize the hierarchies of points, and they used the multiresolution segmental windows to solve the inaccuracy using only one fixed-window. Dimensionality reduction is also one kind of representation, representing the time series with a multidimensional vectors. In [6], Agrawal el al. utilized Discrete Fourier Transform (DFT) to perform the dimensionality reduction, and other techniques have been suggested, including Singular Value Decomposition (SVD) [13] and Discrete Wavelet Transform (DWT) [14]. In [3], Keogh et al. considered the relative importance of each individual linear segment using the weight, which seems more corresponding to subjective similarity, but with difficulties to obtain apriori weighted values. Yi and Keogh et al. [8, 15] proposed Piecewise Aggregate Approximation (PAA), which divides time series into k segmentations and uses the mean of each segment to represent each subseries segmented. And in [16], Keogh et al. gave a more effective method Adaptive Piecewise Constant Approximation (APCA) with segments of varying lengths of each time series.

Before measuring similarity, there is a simple but important preprocess: normalization. It removes the impacts of baseline and scaling factor by going beyond the absolute value of time series. The standard normalization is done by computing the mean of the time series and subtracting it from the real value on each point and then dividing each value by its standard deviation. Gavrilov et al. [17] studied Euclidean Distance on clustering stock data using various normalization methods and proposed the piecewise normalization, which can get much better accuracy, and was implemented in our experiments.

3 General Hierarchical Model (GHM)

Our approach measures the similarity of time series using a hierarchical strategy. The intuitional thought can be described as two following points:

1. The importance of each point in time series will affect their roles in measuring similarity process;
2. The points should be compared with the ones with the same (or approximative) roles.

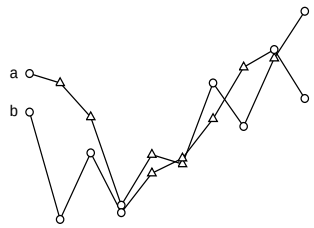


Figure 2. An example to explain the intuitional hierarchical thought.

Figure 2 gives an example to explain our idea. The points in the two time series can be partitioned into two hierarchies: the circle ones are turning, and the triangle ones are interim. They play different roles on the content of time series. According to our idea, the circle (triangle) points in time series a should be corresponding to the circle (triangle) ones in b during the measuring process. For implementing this idea, we design the following three steps to generate the General Hierarchical Model (GHM):

Step 1: Adopt a hierarchical strategy to partition the points into several hierarchies;

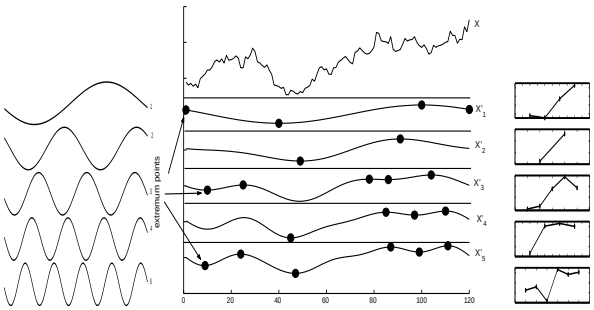
Step 2: Measuring similarity is executed restricting in each hierarchy using a distance function selected or designed;

Step 3: The sum of the results in all the hierarchies is the final measuring distance.

According to the three steps, different hierarchical methods can be designed. The first step is the most important one in the whole process. The strategy determines the representation of the points in each hierarchy and the distance function, which needs selecting or designing. The design should consider different effects of points on the content of time series. GHM can be defined as follows:

Definition 1. Given two series X and Y, and their points have been partition into h hierarchies. $R_X(i)$ and $R_Y(i)$ stand for the representations of points of X and Y in i-th hierarchy respectively, then GHM distance of them can be obtained by the following formula:

$$\mathrm{GHM}(X,Y) = \sum_{i=1}^{h} \mathrm{D}(R_X(i), R_Y(i)) \quad (1)$$



(a) (b) (c)

Figure 3. Generating process of the representation in FHM-1: Fourier waves (a), linear combination of waves X_i' and time points in first 5 hierarchies (b), and the representations in each hierarchy $R_X(i)$ (c).

where D is a distance function to compute the distances in each hierarchy.

3.1 Practical Measures of GHM-FHM

In this subsection, we give two practical hierarchical measures based on GHM. They adopt FFT to implement the model, termed FFT Hierarchical Measures (FHM).

Hierarchical Strategy: Given a time series X and a hierarchy parameter h, we use FFT to generate first h waves, as shown in Figure 3(a). Then the linear combination of the first i Fourier waves X_i' is generated, like the five curves in Figure 3(b). The time points with extremum in X_i' are partitioned into i-th hierarchy, i.e. the black points in Figure 3(b). Then they form the representation $R_X(i)$, which is defined as follows:

Definition 2. Unsequential Subseries: An unsequential subseries is formalized by $X(S)$ where $S = S(1), S(2), ..., S(m)$ is the ordered subset of natural number with the restriction $1 \leq S(1) < S(2) < ... < S(m) \leq |X|$. Each value in $X(S)$ can be got by the formula $X(S)_i = X[S(i)]$. The unsequential subseries is the representation in each hierarchy, i.e. $R_X(i)$, as shown in Figure 3(c).

Actually, there are two selections to generate X_i' according to the order of the Fourier waves:

1. increasing frequency of the corresponding Fourier waves, i.e. the original generating order;
2. decreasing amplitude of the corresponding Fourier waves.

So two methods can be designed, labelled FHM-1 and FHM-2. Then a distance function need designing to compute the distances of the unsequential subseries in each hierarchy.

Distance Function: After the hierarchical partition, we can compute the distance between the unsequential subseries in each hierarchy using any distance function that can deal with different lengths, e.g. DTW, LCSS and EDR. We take DTW as D to analysis the time

```
Input: X, Y, h, D; Output: distance
1.  X_f = FFT(X)  Y_f = FFT(Y)
2.  generate the first h waves series of X and Y
3.  distance=0
4.  for i=1: h
5.      X_i', Y_i'=combination of the first i waves of
        X and Y respectively.
6.      construct R_X(i) and R_Y(i) with the extremum
        points sets of X_i' and Y_i' respectively
7.      distance=distance+D(R_X(i), R_Y(i))
8.  end
9.  return distance
```

Figure 4. FHM (FHM-1 and FHM-2) algorithm.

performance of FHM. Given two time series X with length of n and Y with m, then DTW(X,Y) consumes $O(mn)$ time complexity, but FHM combines the "smaller" DTW processes in each hierarchy to decrease time consumption. After giving D, the complete algorithm of FHM is illustrated formally in Figure 4.

In addition, there are four important explanations to implement FHM (FHM-1 and FHM-2) in the real applications: a) The values of the unsequential subseries is the ones of the original series, not the generating ones X_i'. b) FHM does not usually use all of the points according to the parameter h. Mostly, it is more accurate with larger h, but with more time consumption, i.e. there is always a tradeoff between effectiveness and efficiency. So h can be selected according to the particular requirements on the performances. We performed the experiments to test the performances of ours and others, given in Sect 4. c) If a time point has been in the higher hierarchy, it will not be added in the lower, even if it is also the extremum point in any lower hierarchy. d) There might be the case that the unsequential subseries in i-th hierarchy is not existed, because the extremum points in i-th hierarchy are also with extremum in k-th hierarchy ($k < i$). We use the unsequential subseries in the $(i-1)$-th hierarchy instead of that of the i-th hierarchy in this case.

In FHM, we do not use the extremum points of the original time series directly as the measuring features. Because they may be not the important points reflecting the content, e.g. some noise points etc. We emphasize that point correspondences in similarity measure should be determined by the importance (role) of each point on the content of time series, which decides the similarity, not to restrict the warping path, like the method proposed in [18].

3.2 Hierarchical Filtering of GHM in Similarity Queries

Similarity measure is mostly used in similarity queries in time series database, including range and k-NN queries. The former is to find the time series whose distance with the query Q is below a predefined threshold ε, and the latter searches the most similar k matches in database. Filtering the unmatched time series is widely adopted for better efficiency. In GHM, we use the sum of the distances in all the hierarchies as the measuring result, so the sum of distances of any partial hierarchies can be regarded as the lower bounding of GHM to implement filtering in sequential scan. We

```
Input:  Q,h,D,ε;
Output:  Near(Q)={X | GHM(X,Q) ≤ ε}
1.  initialize  Near(Q)=all time series in database
2.  for each time series  X in database
3.      distance=0
4.      for  i=1:h
5.          distance=distance+D(R_X(i),R_Q(i))
6.          if( distance>ε)
7.              remove  X from Near(Q)
8.              break
9.          end //if, line-6
10.     end //for, line-4
11. end //for, line-2
12. return  Near(Q)
```

Figure 5. GHM filtering algorithm for range query.

assume that the database has stored the representations in each hierarchy for each time series. The filtering algorithms of GHM for range and k-NN queries are shown in Figure 5 and Figure 6 respectively.

4 Experimental Evaluation

In this section, we give the effectiveness tests of ours and others on both synthetic and real data sets. In the experiments, DTW was selected as the distance function D in our approach FHM (FHM-1 and FHM-2), because it is widely used. The competitors included Euclidean Distance, DTW, LCSS and EDR etc. Because EDR is more robust than ERP analyzed in [5], ERP was not included in our experiments. Finally, we give the efficiency comparisons and filtering power tests in Sect 4.3 and Sect 4.4 respectively.

4.1 Experimental Setup of Effectiveness Tests

Two common data sets were used in the experiments. The first one is Synthetic Control Chart Time Series (SYNDATA) data set which was downloaded from UCI KDD archive[1]. It contains 600 examples of synthetic control charts belong to 6 different classes, and each class consists of 100 time series. The length of each time series is equal to 60. The other real data set is the Standard and Poor 500 index (S&P) historical stock data from Mar. 27, 2004 to Mar 26, 2005[2]. We chose the opening price as the experimental data. We only used the stocks whose length is 252 (if the company is removed from the index, the length is smaller than 252,). Based on the official S&P clustering information, the stock data were divided into the classes. Finally, 50 classes contain 442 stock data were used in the experiments by removing the classes which contain only one stock.

Two objective tests, including k-NN query and clustering, were implemented on all methods to compare their effectiveness. Given a query Q, the set of time series which belong to the same class as Q is taken as the standard set std(Q), and the results by different methods are marked knn(Q). The accuracy related to Q can be

[1]The UCI KDD Archive, http://kdd.ics.uci.edu

[2]S&P500, http://kumo.swcp.com/stocks/

```
Input:  Q,h,D,k;
Output:  GHMKNN(Q)={the nearest  k matches of  Q}
/* GHMKNN(Q) is an ordered array according to
decreasing the similarity with  Q */
1.  GHMKNN(Q)={arbitrary  k time series }
2.  initialize  dknn as an array of size  k
    /* dknn saves the distances between  Q and the
    elements in  GHMKNN(Q) */
3.  for  i=1:k
4.      dkkn[i]=GHM(Q,GHMKNN(Q)[i])
5.  end //if( i<j), dknn[i] ≤dknn[j]
6.  for each of other time series  X in database
7.      distance=0
8.      for  i=1:h
9.          distance=distance+D(R_X(i),R_Q(i))
10.         if( distance ≥dknn[k])
11.             break
12.         end //if, line-10
13.     end //for, line-8
14.     if( distance<dknn[k])
15.         remove  GHMKNN(Q)[k] from GHMKNN(Q)
16.         insert  X into GHMKNN(Q)
17.     end //if, line-14
18. end //for, line-6
19. return  GHMKNN(Q)
/*note that: during the operations in  GHMKNN(Q),
the order of its elements must be maintained*/
```

Figure 6. GHM filtering algorithm for k-NN query.

computed, as [12]:

$$\text{Accuracy}(Q) = \frac{|\text{knn}(Q) \cap \text{std}(Q)|}{k} \tag{2}$$

In our experiment, we set the number of time series belong to the same class as the query as the value of k. And each time series is treated as a query. The average of the accuracies is calculated as the final result.

Hierarchical Agglomerative Clustering (HAC) with the complete distance was used to realize the clustering, as in [5, 15, 12, 17]. We computed the clustering accuracy using the method, which is adopted in many applications [12, 17]. Given the standard clustering result $C = C_1, C_2, ..., C_k$ from the apriori classification information and the clustering result using each method $C' = C'_1, C'_2, ..., C'_k$, the accuracy can be computed by the following formulas:

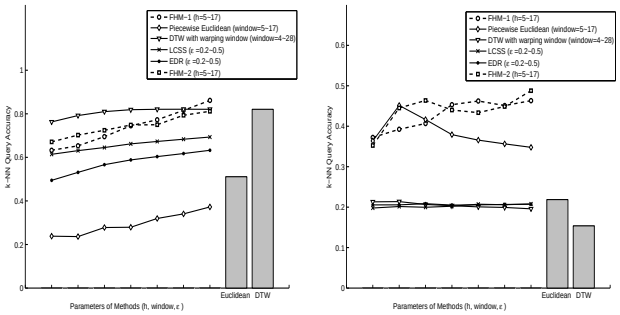
$$\text{Accuracy} = (\text{sim}(C, C') + \text{sim}(C', C))/2 \tag{3}$$

$$\text{sim}(C, C') = (\sum_i \max_j \text{sim}(C_i, C'_j))/k \tag{4}$$

$$\text{sim}(C_i, C'_j) = 2|C_i \cap C'_j|/(|C_i| + |C'_j|) \tag{5}$$

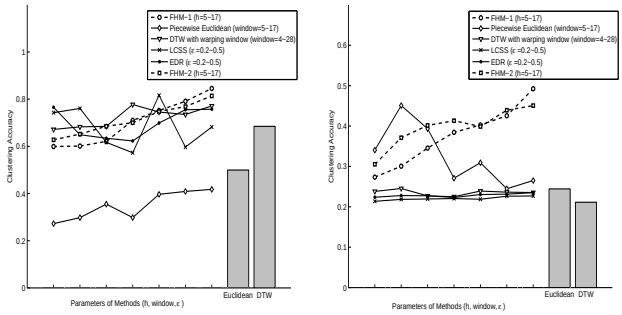
$\text{sim}(C', C)$ above can be calculated similarly as $\text{sim}(C, C')$ in Eq 4. We computed both $\text{sim}(C', C)$ and $\text{sim}(C, C')$, because they are not symmetric. The clustering numbers used in HAC were set to 6 and 50 on SYNDATA and S&P data sets respectively as same as their class numbers.

We also realized DTW with various warping window sizes [18] and the piecewise normalization in Euclidean Distance, which splits



(a) SYNDATA (b) S&P

Figure 7. k-NN query accuracies of the methods.



(a) SYNDATA (b) S&P

Figure 8. Clustering accuracies of the methods.

time series into window, and performs normalization separately within each window [17]. ε in LCSS and EDR were set from 0.2 to 0.5 (one value every 0.05).

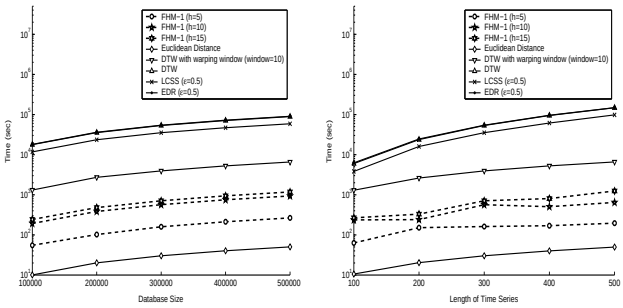
4.2 Effectiveness Experimental Results

k-NN query results on the data sets are given in the Figure 7. From Figure 7(a), we can observe that DTW get the best results on SYNDATA, and FHM-1 and FHM-2 are comparable to it. LCSS and EDR show better capabilities, however, Euclidean Distance has poor accuracies. So it can concluded that SYNDATA is not sensitive to time warping, and suitable to the methods, which allow time warping. FHM can be regarded as an "unsequential piecewise" time warping algorithm, and it can get better efficiency than others. The results on S&P in Figure 7(b) are contrary to that on SYNDATA. But both FHM-1 and FHM-2 also perform the highest accuracies, as Piecewise Euclidean Distance, which are larger than the results of DTW, LCSS, EDR and the plain Euclidian Distance.

Clustering results are listed in Figure 8. The results on both two data sets are similar to that of k-NN query. The superiorities of Piecewise Euclidean Distance than DTW, LCSS and EDR on S&P in Figure 7(b) and Figure 8(b) suggest that the stock data is sensitive to time warping. But the accuracies of FHM with time warping are the best ones on the data due to its hierarchical strategy. And the results also show that FHM-1 and FHM-2 are both good choices to realize the GHM algorithm.

4.3 Time Performance Experiment

In this subsection, we tested time performances of the methods using 1-NN queries with sequential scan, which is data-independent, on the assumptive random walk data set with various database sizes and lengths of time series. We assume that each time series had been preprocessed for each method. The experiment were conducted on the machine with CPU of Celeron 1.70Ghz and 512 MB of physical memory, running Microsoft Windows Server 2003. We only counted the time consumption of distance calculation for exact comparison except accessing the disk. For each query, we tested 100 times and took the sum as the result. In the experiment, the data



(a) Database Size ($length$=300) (b) Length of Time Series ($database\ size = 300000$)

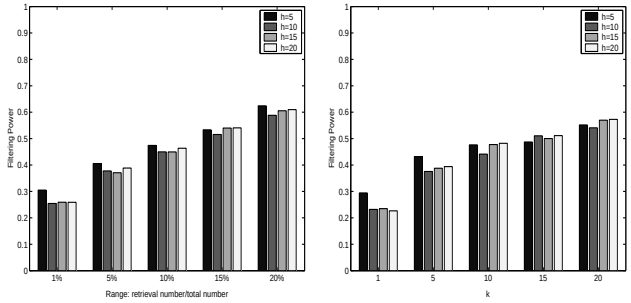
Figure 9. Performance Comparisons of the methods.

size was set from 100000 to 500000 ($length = 300$), and the length of time series was set from 100 to 500 ($database\ size = 300000$). And FHM-1 and FHM-2 are similar, so we only implemented FHM-1 as our approach to compare with others. h in FHM-1 was set to 5, 10 and 15, and ε in LCSS and EDR were set to 0.5. And the warping window size in DTW was set to 10.

In Figure 9, we give the results, which show that LCSS, EDR, DTW are much shower than others. The time consumption of FHM-1 increases linearly along with h. Though Euclidean Distance is faster than ours, considering the effectiveness and efficiency as a whole, our approach is superior to it.

4.4 Filtering Power Tests

Besides the time performance experiment, we also tested the powers of the filtering methods for range and k-NN queries, shown in Figure 5 and Figure 6 respectively. Because the filtering power depends on the data itself, so this test was performed on the real data S&P for approximating the real applications. Each time series in data set was selected as a query. The filtering power is defined as the ratio between the time consumption of 442 queries of the filtering methods and that of the sequential scan. And FHM-1 with



(a) Range query (b) k-NN query

Figure 10. Hierarchical filtering powers in the two kinds of queries.

h =5, 10, 15 and 20 was implemented as the instance of GHM. In the range query, the threshold ε is difficult to set a suitable one, so we set the values to make the ratio between the number of the retrieval series in 442 queries and the total number in the searching space (442×442) equal to 1%, 5%, 10%, 15% and 20% respectively. And k was set to 1, 5, 10, 15 and 20 in k-NN query.

The filtering powers in range and k-NN queries on S&P are shown in Figure 10. The powers of the filtering methods can decrease about 40%∼75% computing time. The results suggest that the effects of h in our approach is little on the filtering powers. And the powers in both of the queries decrease along with increasing of the retrieval number.

5 Conclusions

In this paper, we introduce a new similarity measure model with better effectiveness and efficiency considering the relation between point correspondences and the importance (role) of the points on the content of time series, named General Hierarchical Model (GHM). The key idea of GHM is that any point should be compared with the ones of other time series with same importance (role). We give the general description of GHM and two practical measures based on the model using FFT, termed FFT Hierarchical Measures (FHM includes FHM-1 and FHM-2). And the hierarchical filtering methods of GHM for range and k-NN queries are also proposed to improve the efficiency in the real applications. Finally, k-NN query and clustering experiments on SYNDATA and S&P data sets were used to evaluate the effectiveness of our approach comparing with others. The results demonstrate that ours is more accurate to measure the similarity of different kinds of data. And the time performance tests were also performed using the synthetic data set with various sizes. The results show that ours is slower than Euclidean Distance, but much faster than DTW etc. And we also tested the powers of the filtering methods in the queries. The results present their superior capabilities.

6 References

[1] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *AAAI Workshop on Knowledge Discovery in Database*, pages 359–370, 1994.

[2] Gautam Das, Dimitrios Gunopulos, and Heikki Mannila. Finding similar time series. In *PKDD '97*, pages 88–100.

[3] Eamonn Keogh and M. Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *KDD '98*, pages 239–241.

[4] Lei Chen and Raymond Ng. On the marriage of lp-norms and edit distance. In *VLDB '04*, pages 792–803.

[5] Lei Chen, M. Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *SIGMOD '05*, pages 491–502.

[6] Rakesh Agrawal, Christos Faloutsos, and Arun N. Swami. Efficient similarity search in sequence databases. In *FODO '93*, pages 69–84.

[7] Dina Q. Goldin and Paris C. Kanellakis. On similarity queries for time-series data: Constraint specification and implementation. In *CP '95*, pages 137–153.

[8] Byoung-Kee Yi and Christos Faloutsos. Fast time sequence indexing for arbitrary lp norms. In *VLDB '00*, pages 385–394.

[9] Gautam Das, King-Ip Lin, Heikki Mannila, Gopal Renganathan, and Padhraic Smyth. Rule discovery from time series. In *KDD '98*, pages 16–22.

[10] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *DMKD '03*, pages 2–11.

[11] A. J. Bagnall and G. J. Janacek. Clustering time series from arma models with clipped data. In *KDD '04*, pages 49–58.

[12] Vasileios Megalooikonomou, Qiang Wang, Guo Li, and Christos Faloutsos. A multiresolution symbolic representation of time series. In *ICDE '05*, pages 668–679.

[13] Flip Korn, H. V. Jagadish, and Christos Faloutsos. Efficiently supporting ad hoc queries in large datasets of time sequences. In *SIGMOD '97*, pages 289–300.

[14] Kin pong Chan and Ada Wai-Chee Fu. Efficient time series matching by wavelets. In *ICDE '99*, pages 126–133.

[15] Eamonn Keogh and Michael J. Pazzani. Scaling up dynamic time warping for datamining applications. In *KDD '00*, pages 285–289.

[16] Eamonn Keogh, Kaushik Chakrabarti, Michael Pazzani, and Sharad Mehrotra. Locally adaptive dimensionality reduction for indexing large time series databases. In *SIGMOD '01*, pages 151–162.

[17] Martin Gavrilov, Dragomir Anguelov, Piotr Indyk, and Rajeev Motwani. Mining the stock market: which measure is best? In *KDD '00*, pages 487–496.

[18] Chotirat Ann Ratanamahatana and Eamonn Keogh. Everything you know about dynamic time warping is wrong. In *In Third Workshop on Mining Temporal and Sequential Data*, 2004.