2004－ICS－138　(30)
2004／12／6

Performance Evaluation of Similarity Search Module and Application for Cell Simulator

Shohei HIDO † and Hiroyuki KAWANO††

We develop a similarity search module which performs K-nearest search and range query search in large time series databases. To deal with the high dimensionality of time series data, we apply the techniques such as Adaptive Piecewise Constant Approximation (APCA), multi-dimensional index structure (Hybrid-tree) and approximate distance function to our module. In this paper, we evaluate the performance of the module by using the action potential data of a cardiac cell, which is produced by simBio cell simulator. Experimental results show that the module has high performance and applicability.

1. Introduction

In recent years, advanced techniques for data storing, indexing, query processing and analyzing in large databases have become more important. In order to discover some information or knowledge from the flood of data, various approaches of data mining have been proposed and investigated by many researchers. Major mining algorithms are shown in Table 1.

Efficient similar searches for time series data is one of the important research fields in data mining. Time series data is the continuously recorded time and various attribute values of some target objects. Such kind of data is very common and produced almost everywhere; for example, in science, economy and healthcare. Typical data such as population, heart rate, temperature, stock price and exchange rate have different meaning and variant unit, but all of them could be treated as time series data.

Table 1 Examples of processing on data mining

Name	Description
Clustering	generate clusters of similar data
Classification	detect features for classifying data
Association Rule	find hidden rules between multiple data
Motif Discovery	discover frequent patterns in dataset

† Department of Systems Science
Kyoto University, Kyoto, Japan.
E-Mail: hido@sys.i.kyoto-u.ac.jp
†† Department of Information
and Telecommunication Engineering
Nanzan University, Aichi, Japan.
E-Mail: kawano@it.nanzan-u.ac.jp

The fundamental data processing for time series data are based on the computation of similarity between different data, i.e., the function of distance. In the case of calculating the distance between data by considering time series data which consists of n elements as a point in the n-th dimensional space, Euclidean distance is one of the general distance functions. However, when we apply Euclidean distance to all distance calculation in similarity search, the computational cost of it is too expensive. Therefore, various search space reduction methods are proposed and developed by using different index structures[1)~3)] and many waveform approximation techniques[4)~6)]. Moreover, time series decision tree for classification is one of researches on time series data mining[7),8)].

The rest of this paper is organized as follows. In Section 2, we describe the foundation of the approximate distance functions, multi-dimensional index structures and existing methods for data approximation. Then we explain proposed methods, Hybrid-tree[3)] and APCA[4)], which are implemented in our module. Section 3 contains the detailed description of our method for similarity search and its implementation. In Section 4 we utilize the actual data generated by the cell simulator simBio[9)] of the cell/biodynamics simulation project in Kyoto University, then evaluate the performance of our module for K-nearest search and range query search as examples.

Table 2 Comparison of multi-dimensional index structures

Structure	Division	Dimension	Bounding face	overlap	Supplement
R-tree	Data	N	2N	large	performance degradation in higher dimension
KDB-tree	Space	1	1	none	high cost in tree operation
Hybrid-tree	Space	1	1 or 2	small	robust for higher dimension and tree operation

2. Background and Related Work

2.1 Distance between Data

It is difficult to define the general similarity between two data points[6]. In this paper, we adopt the similarity based on Euclidean distance considering time series data as a multi-dimensional vector. Euclidean distance between two n-th dimensional time series data, $A = \{a_1, \cdots, a_n\}$ and $B = \{b_1, \cdots, b_n\}$ is defined as:

$$D_{Euclid}(A, B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \qquad (1)$$

However, it needs about $O(M^2)$ order time to calculate Euclidean distance between all pairs in the dataset that contains M elements. We define approximate distance function D' making use of index structure and waveform approximation mentioned in the following sections. D' should always satisfy $D'(A, B) \leq D_{Euclid}(A, B)$ and its computational cost should be lower than Euclidean distance. When the distance between A and B is compared with the distance between A and C, if $D_{Euclid}(A, C) \leq D'(A, B)$ is fulfilled, it becomes clear that $D_{Euclid}(A, B)$ meets $D_{Euclid}(A, C) \leq D_{Euclid}(A, B)$ without calculating $D_{Euclid}(A, B)$ actually. Therefore, similarity searching can be speeded up.

2.2 Index structure

Popular multi-dimensional index structures partition the dataset into the smaller area, the most representative method is R-tree[1]. It stores the dataset as a number of Minimum Bounding Region (MBR) and constructs hierarchical structure. It is necessary to partition the dataset with as small overlap between MBRs as possible in order to improve performance of similarity search in R-tree. However, if a dataset has M elements of data, it needs $O(2^{M-1})$ order time to examine all considerable ways to split the dataset into two MBRs. Though there are some excellent techniques such as Quadratic Split, it becomes hard to avoid overlapping between MBRs in higher dimensional data.

There are also index structures to split the data space into two partitions with the bounding face. In KDB-tree[2], the boundary is defined as a value of a certain dimension. The dataset is divided into two partitions that include the data having larger or smaller value than the bounding value. Thus we can have appropriate divisions by KDB-tree, even with higher dimensional dataset in contrast to R-tree. However, there is a problem that the cycle of partition subdividing needs high computational cost when new data is inserted into the tree because the partitions have no overlap.

Hybrid-tree[3] is a index structure using combinational method of R-tree and KDB-tree. It is essentially based on KDB-tree and using space partitioning, but it alters the way of splitting to use not one but two bounding values and accept that the partitions have some overlap like MBR in R-tree in order to avoid the cyclic subdividing. More than one partition could include the same data but the cost of tree operation becomes much lower.

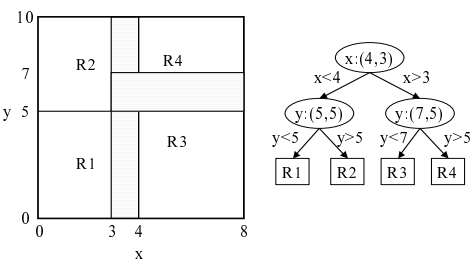

Fig. 1 Image of partitions (left) and its index structure (right)

2.3 Approximation of time series

Well-known waveform processing methods are fourier transform or wavelet transform. In this section, we introduce the discrete wavelet transform, Piecewise Aggregate Approximation (PAA) and Adaptive Piecewise Constant Approximation (APCA) as the discrete approxi-

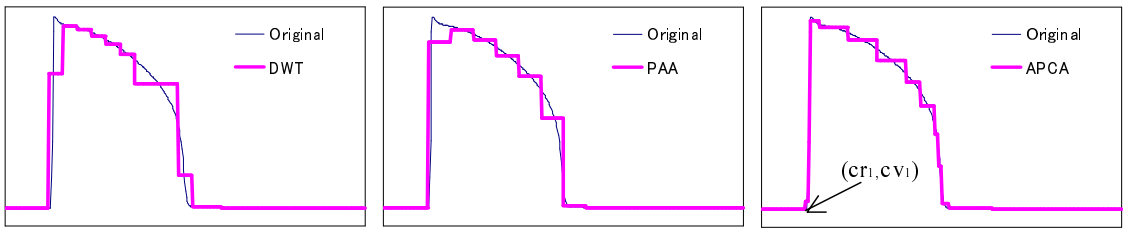

Fig. 2 Examples of waveform approximation

mation method for time series data.

2.3.1 Discrete wavelet transform

Discrete wavelet transform is effective for the data that contains some stable periods with various duration because it can approximate non-periodic uneven waveform. Though there are many mother wavelets, Haar wavelet is the most typical and easy to implement. Bold curve in the left chart of Fig. 2 is an example of Haar wavelet transformation. An transformation algorithm in linear order time and fast approximate distance function is proposed. However, if the data length is not power of two, a preprocessing to add some redundant data until the length reaches power of two and a postprocessing to remove the redundant data is required because this method is applicable only for the data which length is power of two.

2.3.2 PAA

Piecewise Aggregate Approximation separates the original waveform into some periods, whose width is all same, and figure the average value in each region like center of Fig. 2. It is a quite simple and fast technique, and a rapid distance function is also available. The accuracy of approximation depends on the width of period, thus PAA is unsuitable for non-periodic data which contains the stable and unstable region, because the wide period for saving data volume leads to inaccurate approximation. On the other hand, if the width is too narrow, the data amount becomes large unnecessarily.

2.3.3 APCA

Adaptive Piecewise Constant Approximation resolves the issue of PAA by dynamic adjustment of the width of period. The width is extended at stable range and reduced in unstable range. Right-hand graph in Fig. 2 shows better approximation than the others. Similar techniques have been suggested, but Chakrabarti et al made it possible to apply APCA easily to the index structure[4]. APCA data C' is expressed with the pairs of the end point cr_i and the average cv_i of ith period as:

$$C' = \{< cr_1, cv_1 >, \cdots, < cr_K, cv_K >\} \quad (2)$$

3. Techniques for similarity search

3.1 Distance calculation with APCA

3.1.1 Lower bound distance function

We define the fast approximate distance function between the search target Q and the data C in database. In pre-processing, C is transformed into APCA data C' and also stored in database. APCA data Q' is calculated with the condition $qr_i = cr_i$, in order that C' and Q' has the same K periods as shown in Fig. 3.

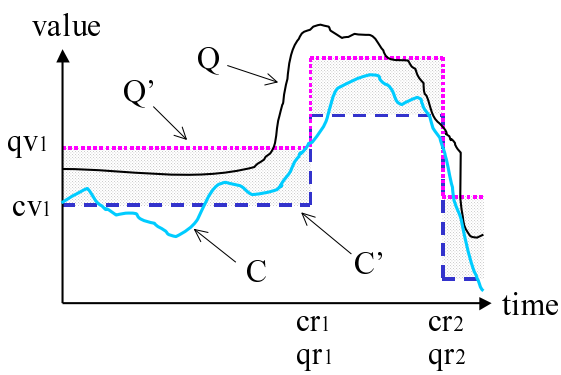

Fig. 3 Example of $D_{LB}(Q', C')$ (shaded region)

(3) is always fulfilled if $D_{LB}(Q', C')$ is defined as shown in (4)

$$D_{LB}(Q',C') \leq D_{Euclid}(Q,C) \quad (3)$$

$$D_{LB}(Q',C') = \sqrt{\sum_{i=1}^{K} (cr_i - cr_{i-1})(qv_i - cv_i)^2} \quad (4)$$

$D_{LB}(Q', C')$ forms the lower boundary of Euclidean distance between Q and C, and can be calculated rapidly. Therefore, as mentioned in Section 2.1, it is possible to save the total cost of the distance calculation in similarity search.

3.1.2 MINDIST function using MBR

We utilize Hybrid-tree as the index structure in our implementation of similarity search module. Each node in Hybrid-tree has MBR figured from the dataset included in the node. An approximate distance function between a time series data and MBR makes it possible to reduce the number of traced node in similarity search.

However, as given in Section 2.3, it is impossible to treat APCA data as a simple multi-dimensional vector because it is constructed as the pairs of the time period and the average value. Consequently we implements modified MBR for APCA data and the distance function between it and time series data.

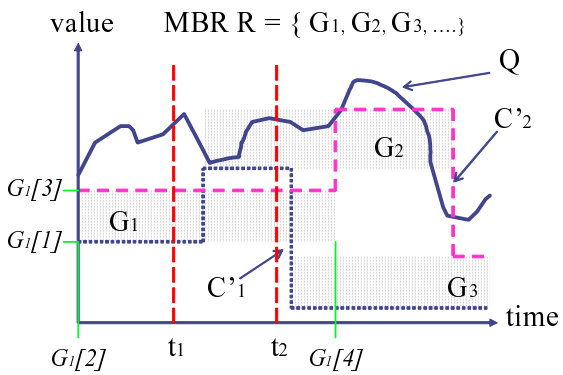

Fig. 4 MBR of APCA data C'_1, C'_2

We define the set of time series data C as U and the set of APCA data C' with K periods as U'. MBR R derived from U is designated with K rectangles $\{ G_i \}$ as:

$$R = \{G_1, G_2, \cdots, G_K\} \tag{5}$$

G_i is expressed with the beginning point $cr_{i-1}+1$, the end point cr_i, the minimum value $cmin_i$ and the maximum value $cmax_i$ of i-th period (Fig. 4)

$$\begin{cases} G_i[1] = min_{(cmin_i)} \\ G_i[2] = min_{(cr_{i-1}+1)} \\ G_i[3] = max_{(cmax_i)} \\ G_i[4] = max_{(cr_i)} \end{cases} (\forall C' \in U') \tag{6}$$

Then MINDIST distance function between R and time series data Q can be given. $D_{min}(Q, G_i, t)$ is the difference between Q and G_i which meets $G_i[2] \leq t \leq G_i[4]$ at time t (7).

$$D_{min}(Q, G, t) = \begin{cases} (G[1]-q_t)^2 & (q_t < G[1]) \\ (q_t-G[3])^2 & (G[3] < q_t) \\ 0 & (\text{otherwise}) \end{cases} \tag{7}$$

In this case $D_{min}(Q, R, t)$ is the minimum value among $D_{min}(Q, G_i, t)$ (8).

$$D_{min}(Q, R, t) = min(D_{min}(Q, G_i, t)) \tag{8}$$

Finally, MINDIST distance function $D_{min}(Q, R)$ is defined as:

$$D_{min}(Q, R) = \sqrt{\sum_{t=1}^{n} D_{min}(Q, R, t)} \tag{9}$$

Because of MINDIST distance function, $D_{min}(Q, R) \leq D_{Euclid}(Q, C)$ is satisfied with all data C in U then the redundant search in the dataset under the node including R could be eliminated.

3.2 Indexing with Hybrid-tree

We store the APCA dataset in the root node in Hybrid-tree, then subdivide the partition with the dimension which has the largest difference between the maximum value and the minimum value until all leaf nodes contain a less number of data than preset value.

Assuming that the APCA data has K periods, we compare the widths of K time periods and K value ranges in order to decide the dimension to split. But, because the unit of value is arbitrary, there is an issue that the difference between the scales of time and value could cause the unfair selection of split dimension. Moreover, Hybrid-tree performs better with equal dimensional width[3]. Thus we invoke the coefficient to equalize the averages of the width of time periods and value ranges in pre-processing.

4. Implementation of similarity search module and performance evaluation

4.1 System architecture

The software and hardware environment which is used for the implementation and the performance evaluation is listed below (Table 3).

Table 3 System architecture

CPU	Intel Pentium4 3.00GHz
Memory	1GB
OS	Debian GNU/Linux (Woody)
Language	Java (Sun Microsystem J2SE 1.4.2)
Database	PostgreSQL 7.2.1

4.2 Data generation

simBio is a biological model simulator developed as Java package by Sarai et al[10]. The biological model is composed as a network of the Reactors (functions) and the Nodes (variables), and its initial parameters are given in XML format. The simulator can solve the ordinary differential equations of models by 4th dimensional Runge-Kutta method and show the charts of variation in the parameters such as membrane potential.

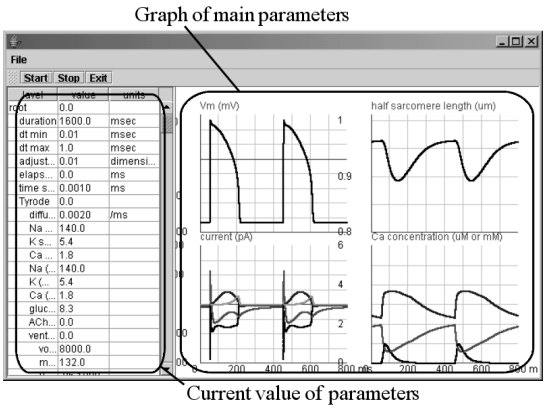

Fig. 5 Execution screen of simBio

Then we added two features to the simulator as preparatory for similarity search in large database. One of them is the capability to store time series data in relational database, and the other is the function to iterate the simulation many times with changing the initial parameters gradually according to the assignment in another XML. Due to these new features, we could obtain the action potential as an example of time series data, using a physiological model of guinea pig cardiac cell called Kyoto model[10),11)] and accumulate a massive amount of the data in database.

4.3 Experiment

At first we stored a large number of waveforms, which consist of the values of the action potential for every millisecond from 0ms to 400ms, in PostgreSQL database with the use of the technique for data generation described in Section 4.2. Next, all of the original waveforms were transformed into APCA data and saved in the database separately. Then we constructed index structures based on Hybrid-tree with from 5000 to 40000 data under the stipulation that all leaf nodes have less than 500 data. After that, K-nearest searches and range query searches were performed. In addition, we compared the result with that of the brute force approach which need to calculate Euclidean distance between the target and all data.

4.4 Result

We measured the computational time of 10-nearest Search to seek the 10 nearest data from the target in the dataset. Fig. 6 is a chart of the experimental result. The vertical axis indicates the execution time and the horizontal axis represents the number of data.

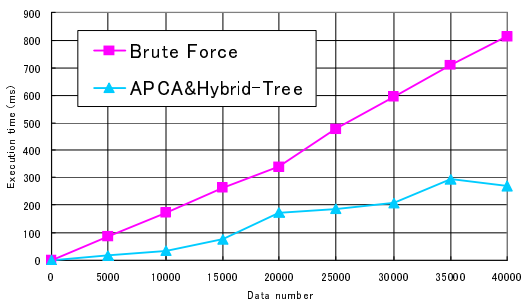

Fig. 6 Evaluation of K-nearest search

In the same way, we execute the range query search to find all data which its distance from the target is less than 10 (out of consideration of the unit). The result is shown in Fig. 7.

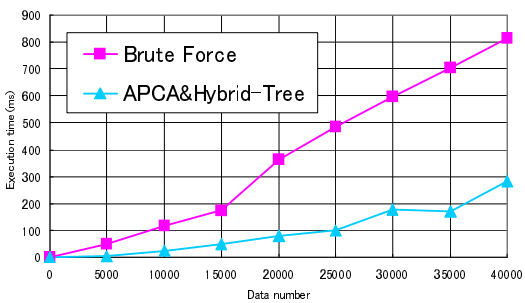

Fig. 7 Evaluation of range query search

4.5 Consideration

In brute force approach, the processing time increases significantly in proportion to the number of data. On the other hand, the growth of time using APCA and Hybrid-tree is slightly increased. Above results indicate that APCA can avoid the redundant calculations of Euclid distance, we can refine the searching space by using the index structure of Hybrid-tree and it is possible to reduce the extension of computational time in similarity search.

Even if the processing cost of the transformation to APCA and the construction of the index is considered, it is still useful to implement our proposed methods in the module for acceleration of similarity searches.

5. Conclusion

Now we can conclude that the combination of APCA and Hybrid-tree could speed up the similarity search such as K-nearest search and range query search. Furthermore, the implementation in Java language which is generally slower than C language shows sufficient performance.

It is believed that there are many way of utilization of this module because of the applicability for any kind of time series data in relational database.

Developed module and accessory is now used in the cell/biodynamics simulation project in Kyoto University. It is embedded in a parameter optimization system to decide the appropriate initial parameters for estimation of the drug effects on living thing during drug development[12].

Acknowledgments This research is an achievement related to the cell / biodynamics simulation project in Kyoto University (project leader: Prof. A.Noma). We received tremendous cooperation and support from Dr. N. Sarai and concerned parties. The part of work was supported by Grant-in-Aid for Scientific Research (16016248,13680482) of the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan.

References

1) A. Guttman. "R-trees: A Dynamic Index Structure for Spatial Searching". In *Proceedings of the ACM SIGMOD*, pages 47–57, 1984.
2) J. T. Robinson. "The K-D-B-tree: A Search Structure for Large Multidimensional Dynamic Indexes". In *Proceedings of the ACM SIGMOD*, pages 10–18, 1981.
3) K. Chakrabarti and S. Mehrotra. "The Hybrid Tree: An Index Structure for High Dimensional Feature Spaces". In *Proceedings of the IEEE ICDE*, pages 440–447, 1999.
4) K. Chakrabarti, E. Keogh, S. Mehrotra, and M. Pazzani. "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases". *ACM Trans. Database Syst.*, 27(2):188–228, 2002.
5) M.Vlachos, M.Hadjieleftheriou, D.Gunopulos, and E. Keogh. "Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures". In *Proceedings of the 9th ACM SIGKDD*, pages 216–225, 2003.
6) E. Keogh. "Indexing and Mining Time Series Data". In *Tutorial at the IEEE ICDM*, 2001. http://www.cs.ucr.edu/%7Eeamonn/tutorial_on_time_series.ppt.
7) Y. Yamada, H. Suzuki, H. Yokoi, and K. Takabayashi. "Experimental Evaluation of Time-series Decision Tree". In *Proceedings of the 2nd International Workshop in Active Mining*, pages 98–105, 2003.
8) Y. Yamada, E. Suzuki, H. Yokoi, and K. Takabayashi. "Decision-Tree Induction from Time-Series Data Based on A Standard-example Split Test". In *Proceedings of the 20th ICML*, pages 840–847, 2003.
9) N. Sarai, A. Amano, S. Matsuoka, T. Matsuda, and A. Noma. "Object-Oriented Cardiac Cell Model Composed of Functional Modules". In *Proceedings of the IEEE EMBS Asian-Pacific Conference on Biomedical Engineering*, 2003.
10) Y. Okamoto. *"The Physiome of the Heart"*, chapter Chapter 2. Cell level. Morikita Syuppan, Japan, 2003.
11) S. Matsuoka, N. Sarai, S. Kuratomi, K. Ono, and A.Noma. "Role of Individual Ionic Current Systems in Ventricular Cells Hypothesized by A Model Study". *Jpn J Physiol*, 53(2):105–123, 2003.
12) S.Hido, N.Sarai, K.Asakura, T.Itoh, K.Koyamada, H. Kawano, and A. Noma. "Extrapolation method of the drug effects on cardiac action potential by the combination of models and experiments". Poster presentation on the 5th ICSB, 2004.