Two Novel Adaptive Symbolic Representations for Similarity Search in Time Series Databases

Ninh D. Pham [#1], Quang Loc Le [*2], Tran Khanh Dang [*3]
Faculty of Computer Science and Engineering
HCM University of Technology, Vietnam National University of HoChiMinh City, Vietnam
[1]ninhpham@contentinterface.com
[2,3]{locle, khanh}@cse.hcmut.edu.vn

***Abstract*—Since the last decade, we have seen an increasing level of interest in time series data mining due to its variety of real-world applications. Numerous representation models of time series have been proposed for data mining, including piecewise polynomial models, spectral models, and the recently proposed symbolic models, such as Symbolic Aggregate approXimation (SAX) and its multiresolution extension, *i*ndexable Symbolic Aggregate approXimation (*i*SAX). In spite of many advantages of dimensionality/numerosity reduction, and lower bounding distance measures, the quality of SAX approximation is highly dependent on the Gaussian distributed property of time series, especially in reduced-dimensionality literature. In this paper, we introduce a novel adaptive symbolic approach based on the combination of SAX and *k*-means algorithm which we call *a*daptive SAX (*a*SAX). The proposed representation greatly outperforms the classic SAX not only on the highly Gaussian distribution datasets, but also on the lack of Gaussian distribution datasets with a variety of dimensionality reduction. In addition to being competitive with, or superior to, the classic SAX, we extend *a*SAX to the multiresolution symbolic representation called *i*ndexable *a*daptive SAX (*ia*SAX). Our empirical experiments with real-world time series datasets confirm the theoretical analyses as well as the efficiency of the two proposed algorithms in terms of the tightness of lower bound, pruning power and number of random disk accesses.**

I. Introduction

The last decade has seen an increasing level of interest in time series data mining due to its variety of application domains, including financial data analysis, sensor network monitoring, network traffic analysis, medical and biological experimental observations, etc. As with all data mining problems, the key aspect to effective and efficient methodologies is the suitable choice of data representation. Because of this fact, there has been a tremendous growth of algorithms and representations produced in indexing and mining time series. In addition, most of the time series representations proposed in the literature are based on the reduced-dimensionality technique and lower bounding the corresponding distance measures defined on the original time series, such as Discrete Fourier Transform (DFT) [4], Discrete Wavelet Transform (DWT) [2], Discrete Cosine Transform (DCT) [8], Piecewise Aggregate Approximation (PAA) [6], Adaptive Piecewise Constant Approximation (APCA) [7], Chebyshev Polynomials (CHEB) [1], Symbolic Aggregate approXimation (SAX) [9], Indexable Piecewise Linear Approximation (IPLA) [3], *i*ndexable Symbolic Aggregate approXimation (*i*SAX) [12], and so on. However, most of the techniques above are the real valued data-reduced representations that are expensive in terms of processing and storage cost in indexing and querying.

While literally hundreds of papers have introduced new algorithms to discretize time series, there is only SAX approach that allows dimensionality/numerosity reduction, symbolic discretization and the creation of lower bounding distance measures (e.g., Euclidean distance, Dynamic Time Warping). To obtain SAX representation, we firstly transform the normalized time series into PAA (Piecewise Aggregate Approximation) by segmenting them into equal sized frames and recording their mean values. Since most of the normalized time series have the highly Gaussian distribution [9], the "breakpoints" producing the equal-sized areas under Gaussian curve are determined to map the PAA coefficients into SAX symbols. As a consequence, the quality of SAX approximation is highly dependent on the three major factors: (1) the Gaussian distributed property of time series, (2) the number of PAA coefficients (word size) for dimensionality reduction, and (3) the number of equal-sized areas under Gaussian curve (alphabet size) for discretization.

In this paper, we introduce an adaptive symbolic approach, adaptive ***S****ymbolic* ***A****ggregate approX**imation (aSAX)* and show that it can be extended to the indexable multiresolution symbolic representation which we call ***i****ndexable* ***a****daptive* ***S****ymbolic* ***A****ggregate appro**X**imation (iaSAX)*. In particular, our representations are based on the original SAX, but adaptive vector of "breakpoints". These adaptive "breakpoints" are determined by a pre-processing phase with the two parameters, word size controlling the number of approximating elements and alphabet size controlling the granularity of each approximating element, as inputs. In addition to being competitive with the classic techniques, we demonstrate that the adaptive representations produce tighter lower bound, greater pruning power, less random disk accesses and thus higher performance in query by content.

The rest of this paper is organized as follows. In Section 2, we briefly review the background material and related work. Section 3 introduces the two novel adaptive representations, namely *a*SAX and *ia*SAX, and the corresponding distance measures defined on them. In Section 4, we show experimental evaluations of the proposed approaches with real-world datasets. Finally, in Section 5, we conclude and give some future research directions.

II. Background and Related Work

As with most problems in time series data mining, the well defined and approximated representation is the key aspect for achieving effectiveness and efficiency when managing the time series data mining tasks. A great number of high level time series representations have been proposed, including data adaptive and non-data adaptive approaches. However, the real valued formats of the proposed techniques are expensive in terms of processing and storing cost. It is thus highly desirable to develop a bit level approximation of time series preserving most of the non-trivial features of existing techniques, including dimensionality reduction and lower bounding guarantee. In this literature, SAX [9] and its multiresolution extension, *i*SAX [12] are the two promising approaches allowing bit level indexing and querying with Vector Approximation File (VA-File) [13] and hierarchical *i*SAX tree index structure, respectively.

The basic idea of our approaches in this paper is based on the classic SAX and extended to *i*SAX representation for indexing and mining terabyte sized time series. These two methods are briefly reviewed below in this section.

A. Review of SAX Representation

The SAX approach [9] is the first symbolic representation of time series that allows dimensionality/numerosity reduction and lower bounding distance measures defined on the original series. In SAX representation, a time series T from n dimensions can be represented in a w-dimensional space by a vector of real numbers $\bar{T} = \bar{t}_1, \ldots, \bar{t}_w$. The i^{th} element of $\bar{T}$ is the mean value of i^{th} equal sized segment, which is calculated by the following equation:

$$\bar{t}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} T_j$$

By recording the mean value of each segment, the time series is converted into a data-reduced representation called PAA [6]. The PAA coefficients are mapped into SAX symbols by using a predetermined “breakpoints” that produce a equal-sized areas under Gaussian curve, as shown in Fig. 1.

We represent a time series T converted into a SAX word with the word size w and the alphabet size a as follows:

$$\text{SAX}\ (T,\ w,\ a) = \mathbf{T^a} = \{t_1, t_2, \ldots, t_{w-1}, t_w\},$$

where t_i can be represented by a symbol or a binary number. Note that, with binary representation, we easily build the VA-File as the indexing algorithm. Fig. 2 illustrates a sample time series converted into SAX binary representation.

As noted above, there is a clear tradeoff between the two parameter choices: word size w and alphabet size a. It is difficult to estimate the best tradeoff due to the highly data dependence. In addition, the “breakpoints” determined by producing a equal-sized areas under Gaussian curve are not suitable because (1) the equiprobable feature of SAX symbols produces low performance for non-uniform or correlated time series datasets, (2) the Gaussian distributed property of time series is affected by the dimensionality reduction, and (3) a large amount of medical time series do not have the highly Gaussian distribution.

βi \ *a*	3	4	5	6	7	8
β1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15
β2	0.43	0.00	-0.25	-0.43	-0.57	-0.67
β3		0.67	0.25	0.00	-0.18	-0.32
β4			0.84	0.43	0.18	0.00
β5				0.97	0.57	0.32
β6					1.07	0.67
β7						1.15

Figure 1. SAX “breakpoints” with the alphabet size from 3 to 8.

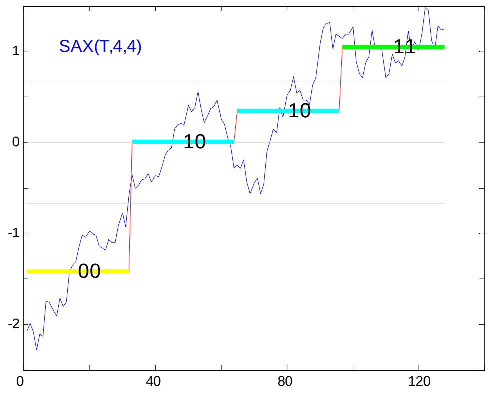

Figure 2. A time series T of length 128, converted into PAA with 4 segments and SAX words of cardinality 4: SAX(T,4,4) = T^4 = {00,10,10,11}.

B. Review of iSAX Representation

The *i*SAX approach [12], a multiresolution symbolic representation, is an interesting extension of the classic SAX which allows the mixed cardinalities of each SAX word. To easily write out the binary strings and its cardinality of each *i*SAX word and to compare the *i*SAX words with the different and mixed cardinalities, we represent an *i*SAX word, such as:

$$i\text{SAX}(T,4,4) = \mathbf{T^4} = \{00, 10, 10, 11\} = \{0^4, 2^4, 2^4, 3^4\},$$

where the *i*SAX word is {0, 2, 2, 3} with cardinality of 4. It is easy to note that the lower resolution *i*SAX can be derived from the higher resolution *i*SAX by ignoring the trailing bits in each symbol of the *i*SAX word. Furthermore, we can apply this promotion trick for mixed cardinality *i*SAX word, such as:

$$i\text{SAX}(T,4,8) = \mathbf{T^8} = \{000, 100, 101, 110\} = \{0^8, 4^8, 5^8, 6^8\}$$
$$= \{000, 10\ , 101, 1\ \ \} = \{0^8, 2^4, 5^8, 1^2\}$$

In order to index *i*SAX representation with different and mixed cardinalities, the time series which have the same *i*SAX mapping are inserted into a file, the name of which encodes the information of *i*SAX. For example, T = $\{0^4, 2^4, 2^4, 3^4\}$ is inserted into the file *0.4_2.4_2.4_3.4.txt*. However, because of the huge skew in the distribution, the largest file contains perhaps 20% of the entire dataset, a threshold *th* was defined to limit the maximum number of time series in a file. The overflow file will be split into 2 files by promoting the cardinality and remapping all of time series in this file to the two new files. Based on this intuition, a hierarchical *i*SAX tree in Fig. 3 was built as the indexing algorithm. The terminal node contains the pointer to an index file on disk with raw time series. The internal node designates a split in SAX space. For approximate matching, we attempt to find a terminal node in the index with the same *i*SAX representation as the query. For exact matching, a combination of approximate search result and lower bounding distance function is proposed to reduce the search space and guarantee no false dismissals.

Likewise, again, the efficiency of *i*SAX representation is highly dependent on the "breakpoints" under Gaussian curve. Furthermore, the hierarchical height-unbalanced *i*SAX tree, especially severely height-unbalanced on the lack of Gaussian distribution datasets (e.g., ECG datasets [12]), produces low performance in terms of both I/O and CPU cost.

C. *Distance Measure*

As noted in [12], there is no measurable difference between Dynamic Time Warping and Euclidean distance in most large datasets. Euclidean distance is thus used as the similarity distance measure in this paper. Given two time series *T* and *S* of the same length *n*, their Euclidean distance is defined as:

$$D(T,S)=\sqrt{\sum_{i=1}^{n}(T_i-S_i)^2}$$

If we further transform time series into the classic SAX representation, we can define a lower bounding approximation to the Euclidean distance by the following equation:

$$MINDIST(T_{SAX},S_{SAX})=\sqrt{\frac{n}{w}}\sqrt{\sum_{i=1}^{w}\left(dist(T_{SAXi},S_{SAXi})\right)^2}$$

where *dist()* function can be implemented based on a table lookup predetermined by the SAX "breakpoints".

For completeness, to compare two *i*SAX words of different cardinalities, we firstly convert them into the same highest cardinality for each *i*SAX symbol and then calculate their lower bound. In this case, the lower bounding MINDIST function is quite loose; however, due to the available query time series, we are free to use its PAA representation to obtain the tighter bounding. The lower bounding MINDIST_PAA _*i*SAX function between a PAA representation and an *i*SAX representation is defined by the following equation:

$$\text{MINDIST\_PAA\_}i\text{SAX}(T_{PAA},S_{iSAX})=\sqrt{\frac{n}{w}}\sqrt{\sum_{i=1}^{w}\begin{cases}(\beta_{Li}-T_{PAAi})^2 & if\,\beta_{Li}>T_{PAAi}\\(\beta_{Ui}-T_{PAAi})^2 & if\,\beta_{Ui}<T_{PAAi}\\0 & otherwise\end{cases}}$$

where T_{PAA} is the PAA representation of time series *T* and S_{iSAX} is the *i*SAX representation of time series *S*, such that $|T| = |S| = n$, $|T_{PAA}| = |S_{iSAX}| = w$. The j^{th} cardinal value of S_{iSAX}

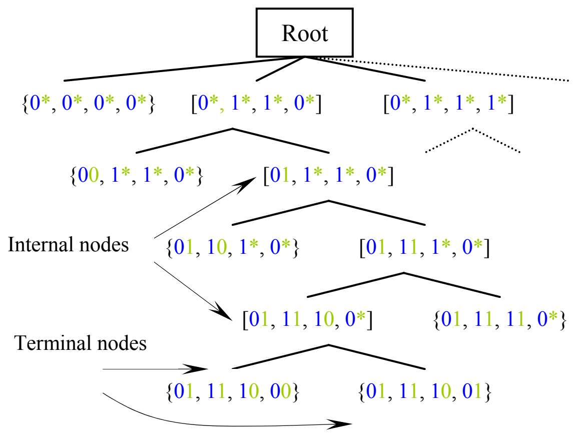

Figure 3. An illustration of a hierarchical *i*SAX index with the base cardinality $b = 2$ and the word size $w = 4$.

derives from a PAA value, β_L and β_U are two lower and upper breakpoints of each *i*SAX symbol, $1 \le j \le w$.

Recall that the two lower bounding functions above can be used for both SAX and *i*SAX representations. In order to solve the ambiguity, we use MINDIST as SAX lower bound and MINDIST_PAA _*i*SAX as *i*SAX lower bound.

III. The Adaptive Symbolic Representations

In the classic SAX approach, there is an assumption that most of the time series have the highly Gaussian distribution because each dimension is discretized into symbols based on the predetermined "breakpoints" under Gaussian curve. Also, the discretization is often applied on the reduced-dimensionality data (e.g., PAA representation); however, no specific algorithm for that option was proposed. Finally, each PAA coefficient is mapped to symbols with equiprobability, which is the simple partitioning that coincides for the highly Gaussian distributed datasets. In this section, we propose our approaches that lead to more efficient and effective searching using the classic symbolic representations. For concreteness, we introduce the two novel adaptive representations, namely *a*SAX and *ia*SAX, based on the classic representations but with the adaptive "breakpoints". These representations produce the greater performance in indexing and querying time series on both the highly Gaussian distribution datasets and the lack of Gaussian distribution datasets.

A. *Adaptive SAX (aSAX)*

Our proposed approach, ***a**daptive **S**ymbolic **A**ggregate approX**imation (aSAX)* is based on the classic SAX but with the adaptive "breakpoints". These adaptive "breakpoints" are predetermined by a pre-processing phase based on Lloyd's algorithm [10], which is nothing but the *k*-means algorithm [11], [5] for clustering, specialized to one dimension. In particular, we use a small set of the normalized time series as the training set, the alphabet size of symbols as parameter *k* in *k*-means algorithm and the "breakpoints" under Gaussian curve as the clever initialization of cluster intervals. We note that the original normalized time series can be used for training but for the best performance we suggest the PAA representations of training time series.

Given a training set, we first convert all time series into PAA representations and store them in a PAA array. Denote by t_n the value of the nth PAA point in PAA array, $[\beta_i, \beta_{i+1})$ as the given set of interval under Gaussian curve, and r_i as the representative value for the interval i for $i = 0,\ldots,a-1$, $\beta_0 = -\infty$, $\beta_a = \infty$. Set $\Delta = \infty$, and fix $\gamma > 0$. The training algorithm to achieve the "adaptive" breakpoints for *a*SAX representation is derived based on *k*-means algorithm as follows:

1. For $i = 0,\ldots,a-1$, compute the new representative value as the center of mass of all PAA points in the interval $[\beta_i, \beta_{i+1})$, by $r_i = \frac{1}{N_i} \sum_{t_n \in [\beta_i, \beta_{i+1})} t_n$, where N_i is the total number of PAA points in the interval $[\beta_i, \beta_{i+1})$.
2. Compute the new intervals by $\beta_i = (r_{i-1} + r_i) / 2$, for $i = 1,\ldots,a-1$.
3. Compute the total representation error as $\Delta' = \sum_{i=1}^{k} \sum_{t_n \in [\beta_i, \beta_{i+1})} (t_n - r_i)^2$. If $\frac{\Delta - \Delta'}{\Delta} < \gamma$, then STOP. Otherwise, set $\Delta = \Delta'$, and go to step 1.

The above algorithm is guaranteed to converge because of the clever initialization with the equiprobability of the classic SAX symbols under Gaussian curve. The approximation quality of the symbols is improved in each step of algorithm. By doing so, we obtained a resultant vector of new "breakpoints" for the adaptive approach, as shown in Fig. 4. After the third step, the error introduced by approximating the PAA points by the corresponding symbols is further reduced. Since the representation quality of each symbol directly affects the quality of symbolic representation, this leads to better both tightness of lower bounding function and pruning power. The improvement by the proposed *a*SAX technique is much more apparent for the lack of Gaussian distribution datasets, especially by the analysis of the performance results of each technique.

The astute reader will have noticed that we can easily achieve the more adaptive "breakpoints" by clustering each PAA coefficient separately. That means we have each "break points" for each PAA coefficient, thus suffer more processing and storage cost but tighter lower bound and greater pruning power. Furthermore, the set of adaptive "breakpoints" above has the approximately multiresolution feature. If we scale to achieve the multiresolution feature, we can easily apply them on *i*SAX representation without reducing performance. In the next section, we formalize this intuition and provide details on algorithm to achieve the multiresolution feature of the adaptive "breakpoints" for the multiresolution symbolic approach.

βi \ a	3	4	5	6	7	8
β1	-0.51	-0.82	-0.98	-1.09	-1.23	-1.29
β2	0.52	0.00	-0.32	-0.54	-0.72	-0.81
β3		0.79	0.32	0.00	-0.24	-0.40
β4			0.98	0.54	0.24	0.00
β5				1.11	0.73	0.42
β6					1.24	0.82
β7						1.35

Figure 4. *a*SAX "breakpoints" of the random walk dataset of length 256 with the word size of 8 and the alphabet size from 3 to 8.

B. Indexable Adaptive SAX (iaSAX)

As noted before, the set of adaptive "breakpoints" can be applied to *i*SAX, a multiresolution representation of SAX for mining the truly massive datasets. This approach is called ***i**ndexable **a**daptive **S**ymbolic* Aggregate *approXimation (iaSAX)*. Although the set of adaptive "breakpoints" predetermined does not have the multiresolution property, it is easy to achieve by scaling them from the low resolution to the high resolution. In this section, we introduce a simple heuristic algorithm to produce a set of multiresolution breakpoints from the adaptive ones.

Given the set of adaptive "breakpoints" from the training phase, denote by BP_i as the "breakpoints" of the ith resolution, $i = 1,\ldots,n$. That means BP_i is a vector of $2^i + 1$ breakpoints, including $\beta_0 = -\infty$ and $\beta_{2^i} = \infty$. The heuristic algorithm to achieve the adaptive multiresolution "breakpoints" is described as follows:

1. The lowest resolution "breakpoints" need not be scaled. In fact, we can choose the lowest resolution as the base cardinality b used in the hierarchical *i*SAX tree index.
2. The higher resolution "breakpoints" BP_{i+1} should be scaled based on the lower resolution "breakpoints" BP_i. In particular, the $(2j)$th value of BP_{i+1} must be modified equally to the jth value of BP_i and the $(2j+1)$th value of BP_{i+1} is achieved by the combination of BP_i and the *k*-means algorithm as the *a*SAX approach or by the mean of $(2j)$th value and $(2j+2)$th value of BP_{i+1}.
3. Go to step 2 until we achieve the highest resolution "breakpoints".

The heuristic algorithm above can simply achieve the set of multiresolution adaptive "breakpoints" as shown in Fig. 5. It is noted that the symmetric feature of Gaussian "breakpoints" is not guaranteed in the adaptive "breakpoints"; however, it does not matter because of the non-symmetric probability distribution of most of real-world datasets.

It is noted that the proposed adaptive representations are based on the classic representations but with the adaptive "breakpoints". Therefore, the equations defined for lower bounding approximations are the same. For concreteness, we have two lower bounding functions: (1) MINDIST of two *a*SAX representations and (2) MINDIST_PAA_*ia*SAX of a PAA representation and an *ia*SAX representation.

Recall that due to the clustered feature of the *ia*SAX "breakpoints", the number of nodes of the hierarchical *ia*SAX tree, especially the terminal nodes, is reduced. This

a / βi	3	4	5	6	7	8
β1	-0.51	-0.82	-0.98	-1.09	-1.23	-1.29
β2	0.52	0.00	-0.32	-0.51	-0.72	-0.82
β3		0.79	0.32	0.00	-0.24	-0.40
β4			0.98	0.52	0.24	0.00
β5				1.11	0.73	0.42
β6					1.24	0.79
β7						1.35

Figure 5. *ia*SAX "breakpoints" of the random walk dataset of length 256 with the word size of 8 and the alphabet size from 3 to 8.

leads to better both I/O cost and CPU cost in query processing. In Section 4, our experimental results will confirm the theoretical analyses as well as the efficiency of the two adaptive approaches in indexing and querying time series databases.

IV. Empirical Evaluation

In this section, we illustrate through extensive empirical experiments the query performance of the adaptive symbolic representations with the two classic techniques, SAX and *i*SAX, in terms of the tightness of lower bound, pruning power and number of random disk accesses. In our indexing experiments, we tested on a variety of 13 benchmark datasets and 1 synthetic random walk dataset. In particular, 13 benchmark datasets contain data from a wide range of applications and with different characteristics, where each dataset has 2500 time series of length 256. In order to test the efficiency of query processing, we use large datasets, Koski ECG and Random Walk dataset, each of which contains about 100K time series of length 256.

Throughout our experiments, we target at the nearest neighbor queries and study the efficiency and effectiveness of our proposed approaches. All experiments are conducted by Matlab on a Pentium IV PC 1.8 GHz with 1 GB memory.

A. *Tightness of Lower Bounds*

It is important to note that our adaptive approaches are the improvements of the classic ones, not indexing structure such as R*-tree, Hybrid tree, etc. We therefore begin with a simple experiment to compare the tightness of lower bound (TLB) of *a*SAX and *ia*SAX with other lower bounding time series representations, including *i*SAX, SAX and PAA. TLB is calculated as:

$$TLB = \frac{Lower\ Bound\ Dist(T,\ S)}{True\ Euclidean\ Dist(T,\ S)}$$

We first measure TLB on Koski ECG dataset. We sampled randomly T and S 1000 times for each combination of parameters. We vary the time series length [480, 960, 1440, 1920] and the number of bytes per time series available to the dimensionality reduction approach [16, 24, 32, 40]. With the assumption of 4 bytes per real valued coefficient, we use [4, 6, 8, 10] coefficients for PAA representations. For the sake of fair comparisons, we hard code the cardinality to 16 for the bit representations (*ia*SAX, *i*SAX, *a*SAX and SAX), resulting in [32, 48, 64, 80] symbols per word.

Recall that TLB is in the range [0, 1] and larger values are better. A value of 1 would allow a constant time search algorithm, and a value of 0 would force the indexing structure to degrade to sequence scan.

It is clear from the Fig. 6 that the adaptive representations (*ia*SAX and *a*SAX) achieve tighter lower bounds than the classic ones (*i*SAX and SAX), especially *a*SAX competitive with SAX in some parameter choices. This is because *ia*SAX and *a*SAX are converted based on the adaptive "breakpoints" which minimized the approximating error of each symbol, as confirmed in Section 3.A.

B. *Pruning Power*

To be free of the possibility of implementation bias, we achieve by comparing the pruning power of the various approaches. The pruning power P is defined as the fraction of the database that must be examined before we can guarantee that we have found the nearest match to a 1-*nn* query.

$$P = \frac{Number\ of\ objects\ that\ must\ be\ examined}{Number\ of\ objects\ in\ database}$$

To calculate P with the sake of fair comparisons, we do the following. We remove 10% of the dataset (a contiguous subsection) for randomly taken queries and build the VA-File index with the remaining 90% for all techniques, including *ia*SAX, *i*SAX, *a*SAX and SAX. Note that, the key difference of such techniques is the "breakpoints" that directly affect the lower bounding functions and thus the pruning power. For query algorithm, we perform the Simple Search VA-File algorithm [13] in order to guarantee no false dismissals because of the reduced-dimensionality data of the symbolic representations. For each reported on a particular dataset, for the dimensionality reduction 32:1, the cardinality of 16 and time series of length 256, we average the results of 100 experiments for nearest neighbor queries. Fig. 7 shows a competitive result of 4 techniques, including *ia*SAX, *i*SAX, *a*SAX and SAX.

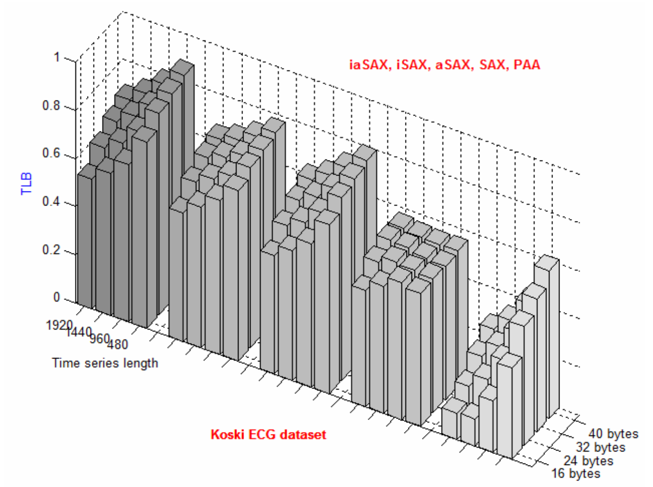

Figure 6. The tightness of lower bounds for various time series representations on Koski ECG dataset.

In general, the results show that the pruning power of *ia*SAX and *i*SAX is perhaps similar; however, *a*SAX representation greatly outperforms or at least is comparable to SAX approach, expressing the superiority in the ECG and Power datasets. The reason is that our representations are based on the clustering feature of data instead of the equal population of data of the classic ones. Furthermore, we would like to emphasize that our results here are obtained in the dimensionality reduction 32:1, that means we use 32 bytes for a time series of length 256 with the assumption of 4 bytes per real value. If we increase the available bytes for a time series, the results will be much improved.

Fig. 8 and Fig. 9 further show the stronger results of the adaptive representations in some lack of Gaussian distribution datasets, including Muscle Activation and Foetal ECG datasets. The experiments were similar to those of Fig. 7, except that the available bytes for a time series ranging from 8 to 64 bytes.

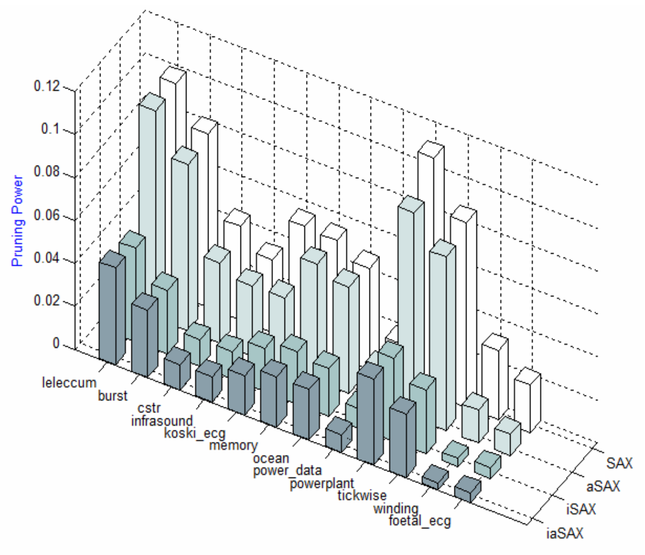

Figure 7. The pruning power of *ia*SAX, *i*SAX, *a*SAX and SAX approaches for the dimensionality reduction 32:1 on 12 real datasets.

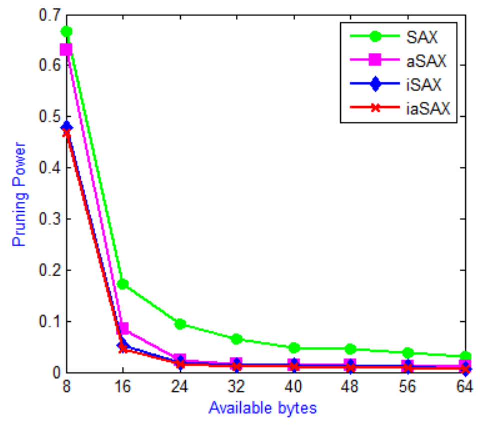

Figure 8. The pruning power on Foetal ECG dataset with the available bytes ranging from 8 bytes to 64 bytes.

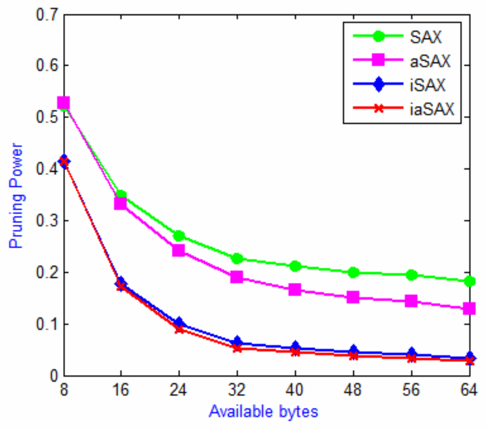

Figure 9. The pruning power on Muscle Activation dataset with the available bytes ranging from 8 bytes to 64 bytes.

The results impressively showed that the adaptive representations performed exceptionally well on the lack of Gaussian distribution datasets. Again, these experiments indicate that *ia*SAX and *a*SAX techniques have fewer false alarms, hence greater pruning power in query by content.

C. *Number of Random Disk Accesses*

It is noted that disk I/O tends to be a bottleneck for any data mining task since the datasets encountered by data miner typically don't fit in main memory. For completeness, we compared different representations, *ia*SAX and *i*SAX, in terms of random disk accesses during the process of the nearest neighbor retrieval of a query time series. In particular, we use large datasets, Koski ECG as the lack of Gaussian distribution dataset and random walk as the highly Gaussian distribution dataset, each of which contains about 100K time series of length 256. It is easy to note that the hierarchical tree index is constructed given base cardinality b, word length w, and threshold th. For the sake of fair comparisons, we hard code $b = 4$, $w = 8$ for all time series of length 256 and scale the threshold parameter th for particular datasets to estimate the performance in the high resolutions of the indexing structure. Because of the different probability distribution of the two datasets, we use $th = [30, 10]$ for random walk dataset and $th = [100, 50]$ for Koski ECG dataset to achieve the highest cardinality of tree $a = [64, 128]$.

From Fig. 10 to Fig. 13 show the number of random disk accesses for the nearest neighbor queries on random walk dataset and Koski ECG dataset with lower bounding the Euclidean distance, using the two threshold values $th = [30, 10]$ and $th = [100, 50]$, respectively. Although the lines are not smooth because of the random sampling queries, it is clear that *ia*SAX representation significantly outperforms *i*SAX representation in both the highly Gaussian distribution datasets (i.e., random walk dataset) and the lack of Gaussian distribution datasets (i.e., Koski ECG dataset). Likewise, again, we would like to notice that we should refer to the correlated or clustered feature more than equiprobable feature of the real-world datasets.

V. Conclusions

In this paper, we introduced two novel adaptive techniques, *a*SAX and *ia*SAX, based on the clustered feature of real datasets. We have shown that our representations are competitive with, or superior to, the classic representations in indexing and querying large time series databases in terms of the tightness of lower bound, pruning power and number of random disk accesses. Note that we can easily improve the heuristic algorithms to achieve the better "breakpoints" by clustering each PAA coefficient separately. That means we have each "breakpoints" for each PAA coefficient, thus suffer more processing and storage cost but tighter lower bound and greater pruning power.

There are a number of challenging time series data mining tasks such as motif discovery, discord discovery, classification and clustering which we intend to extend in future work. Furthermore, we plan to consider the relationship of the word size, the alphabet size of the symbolic approaches and the probability distribution of real datasets to achieve the better efficiency when managing the time series data mining tasks.

References

[1] Y. Cai, and R. T. Ng, "Indexing spatio-temporal trajectories with chebyshev polynomials," Proc. of ACM SIGMOD, 2004, pp. 599-610.

[2] K. Chan, and A. Fu, "Efficient time series matching by wavelets," Proc. of ICDE, 1999, pp. 126-133.

[3] Q. Chen, L. Chen, X. Lian, Y. Liu, and J. X. Yu, "Indexable PLA for efficient similarity search," Proc. of VLDB, 2007, pp. 435-446.

[4] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time series databases," Proc. of ACM SIGMOD, 1994, pp. 419-429.

[5] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and AE. Abbadi, "Vector approximation based indexing for non-uniform high dimensional data sets," Proc. of CIKM, 2000, pp. 202-209.

[6] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," Journal of Knowledge and Information System, 2000, pp. 263-286.

[7] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra, "Locally adaptive dimensionality reduction for indexing large time series databases," Proc. of ACM SIGMOD, 2001, pp. 151-162.

[8] F. Korn, H. V. Jagadish, and C. Faloutsos, "Efficiently supporting ad hoc queries in large datasets of time sequences," Proc. of ACM SIGMOD, 1997, pp. 289-300.

[9] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," *Data Mining Knowledge Discovery*, 2007, pp. 107-144.

[10] S. P. Lloyd, "Least squares quantization in PCM," Proc. of IEEE Transaction on Information Theory, 1982, pp. 129-137.

[11] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281-297.

[12] J. Shieh, and E. Keogh, "*i*SAX: Indexing and mining terabyte sized time series," Proc. of ACM SIGKDD, 2008, pp. 623-631.

[13] R. Weber, H.-J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," Proc. of VLDB, 1998, pp. 194-205.

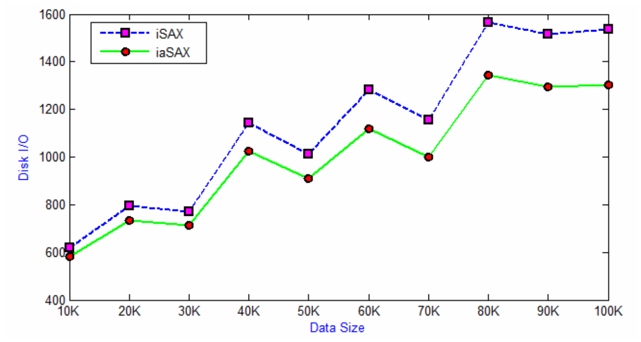

Figure 10. Number of random disk accesses for 1-*nn* queries on the random walk dataset with b = 4, w = 8, th = 30, using *ia*SAX and *i*SAX representations on the hierarchical tree index.

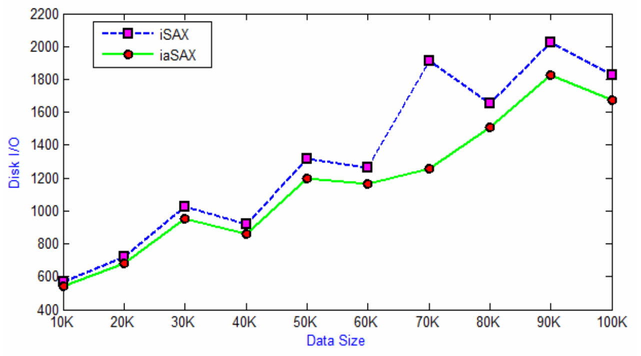

Figure 11. Number of random disk accesses for the nearest neighbor queries on the random walk dataset with b = 4, w = 8 and th = 10, using *ia*SAX and *i*SAX representations on the hierarchical tree index.

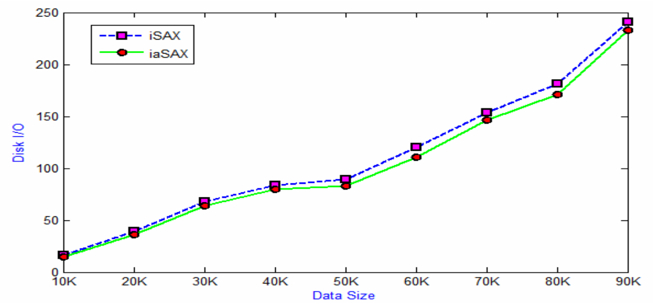

Figure 12. Number of random disk accesses for the nearest neighbor queries on the Koski ECG dataset with b = 4, w = 8 and th = 100, using *ia*SAX and *i*SAX representations on the hierarchical tree index.

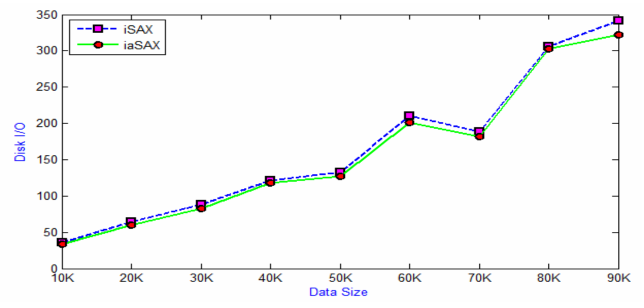

Figure 13. Number of random disk accesses for the nearest neighbor queries on the Koski ECG dataset with b = 4, w = 8 and th = 50, using *ia*SAX and *i*SAX representations on the hierarchical tree index.