A Multi-Metric Index for Euclidean and Periodic Matching

Michail Vlachos[1], Zografoula Vagena[2], Vittorio Castelli[1], and Philip S. Yu[1]

[1] IBM. T.J. Watson Research Center
[2] University of California, Riverside

Abstract. In many classification and data-mining applications the user does not know a priori which distance measure is the most appropriate for the task at hand without examining the produced results. Also, in several cases, different distance functions can provide diverse but equally intuitive results (according to the specific focus of each measure). In order to address the above issues, we elaborate on the construction of a hybrid index structure that supports query-by-example on shape and structural distance measures, therefore lending enhanced exploratory power to the system user. The shape distance measure that the index supports is the ubiquitous Euclidean distance, while the structural distance measure that we utilize is based on important periodic features extracted from a sequence. This new measure is phase-invariant and can provide flexible sequence characterizations, loosely resembling the Dynamic Time Warping, requiring only a fraction of the computational cost of the latter. Exploiting the relationship between the Euclidean and periodic measure, the new hybrid index allows for powerful query processing, enabling the efficient answering of kNN queries on both measures in a single index scan. We envision that our system can provide a basis for fast tracking of correlated time-delayed events, with applications in data visualization, financial market analysis, machine monitoring/diagnostics and gene expression data analysis.

1 Introduction

Even though many time-series distance functions have been proposed in the data-mining community, none of them has received the almost catholic acceptance that the Euclidean distance enjoys. The Euclidean norm can be considered as the most rudimentary *shape-matching* distance measure, but it has been shown to outperform many complex measures in a variery of clustering/classification tasks [3], while having only a fraction of the computational and logical complexity of the competing measures.

Lately however, time-series researchers are also starting to acknowledge certain limitations of shape matching distance measures, and therefore we are gradually experiencing a shift to more *structural* measures of similarity. These new structural measures can greatly enhance our ability to assess the inherent similarity between time sequences and tend to be more coherent with theories governing

the human perception and cognition. Recent work quantifying structurally the similarity between sequences, may take into consideration a variety of features, such as change-point-detection [2], sequence burstiness [7], ARIMA or ARMA generative models [9], and sequence compressibility [4].

In many cases though, there is no clear indication whether a shape or a structural measure is best suited for a particular application. In the presence of a heterogeneous dataset, specific queries might be tackled better using different measures. The distance selection task becomes even more challenging, if we consider that different distance measures can sometimes also provide diverse but equally intuitive search results.

In an effort to mitigate the distance selection dilemma, we present an index structure that can answer multi-metric queries based on both shape and structure, allowing the end user to contrast answer sets, explore and organize more effectively the resulting query matches. The proposed indexing scheme seamlessly blends the Euclidean norm with a structural *periodic* measure. Periodic distance functions were recently presented in [8] and have been shown to perform very effectively for many classes of datasets (i.e., ECG data, machine diagnostics, etc). However, in the original paper no indexing scheme had been proposed. Recognizing that the periodic measure can easily (and cost-effectively) identify time-shifted versions of the query sequence (therefore loosely resembling Time-Warping), we exploit the relationship between the euclidean and the periodic measure in the frequency domain, in order to design an index that supports query-by-example on both metrics. By intelligently organizing the extracted sequence features and multiplexing the euclidean and periodic search we can return the k-NN matches of both measures in a *single* index scan. Both result sets are presented to the user, expanding the possibilities of interactive data exploration, providing more clues as to the appropriate distance function.
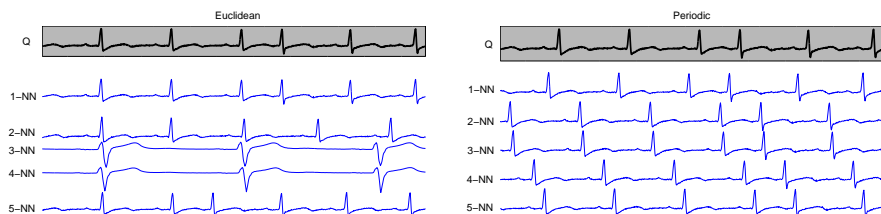

Fig. 1. 5-NN euclidean and periodic matches on an ECG dataset.

A sample output of the proposed index for a database of ECG data is shown in Fig. 1. For the specific query, all instances returned by the periodic measure belong to the same class of sequences and correspond to time-shifted variations of the query sequence. The 1,2,5-Nearest-Neighbor (NN) matches of the Euclidean metric can also be considered similar to the query, however the 3-NN and 4-NN would be characterized as spurious matches by a human. The purpose of this (rather simplistic) example, is to emphasize that in many cases multiple measures are necessary, since each metric can harvest a different subset of answers.

Even though other multi-metric distances have been presented in [10, 6] (but for different sets of distance functions), queries needed to be issued multiple times for retrieving the results of the different measures. Therefore, the presented index has two distinct advantages:

■ It supports concurrent euclidean and periodic matching, returning both sets of Nearest-Neighbor matches in a *single* index scan. So it allows for both rigid matching (euclidean distance), or more flexible periodic matching, by identifying arbitrary time shifts of a query (periodic measure).

■ Performance is not compromised, but is in fact improved (compared to the dual index approach) due to the reduced index size and the intelligent tree traversal.

Given the above characteristics, we expect that the new index structure can provide necessary building blocks for constructing powerful 'all-in-one' tools, within the scope of applications such as decision support, analysis of causal data relationships and data visualization.

2 Background

The periodic measure and the hybrid index that we will describe later operate in the frequency domain, therefore we will succinctly describe important concepts from harmonic analysis.

2.1 Frequency Analysis

A discrete-time signal $\mathbf{x} = [x_0, \ldots, x_{N-1}]$ of length N can be thought of as a period of a periodic signal and represented in terms of its Fourier-series coefficients $\{X_k\}_{k=0}^{N-1}$ by

$$ x_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X_k e^{2\pi j(k/N)n}, \qquad n = 0, \ldots, N-1, $$

where $j = \sqrt{-1}$ is the imaginary unit. The coefficient X_k is defined by

$$ X_k = \rho_k e^{j\theta_k} = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n e^{-2\pi j(k/N)n}, \qquad k = 0, \ldots, N-1, $$

and corresponds to the frequency $f_k = k/N$. Here ρ_k and θ_k are respectively the magnitude and the phase of X_k. Parseval's theorem states that the energy $\mathcal{P}$ of the signal computed in the frequency domain is equal to the energy computed in the Fourier domain:

$$ \mathcal{P}(\mathbf{x}) = \|\mathbf{x}\|^2 = \sum_{k=0}^{N-1} x_k^2 = \mathcal{P}(\mathbf{X}) = \|\mathbf{X}\|^2 = \sum_{k=0}^{N-1} \|X_k\|^2. $$

Many operations are substantially more efficient in the frequency domain than in the time domain. The use of frequency-domain operations is often appealing thanks to the existence of the efficient Fast Fourier Transform, which has computational complexity of $O(N \log N)$.

3 Distance Functions

3.1 Euclidean Distance

Let $\mathbf{x}$ and $\mathbf{y}$ be two time sequences of length N having Discrete Fourier Transform $\mathbf{X}$ and $\mathbf{Y}$, respectively. The Euclidean distance $d(\mathbf{x}, \mathbf{y})$ between $\mathbf{x}$ and $\mathbf{y}$ (i.e., the ℓ_2 norm of $\mathbf{x} - \mathbf{y}$) is defined by $d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y}) \cdot (\mathbf{x} - \mathbf{y})} = \sqrt{\sum_{k=1}^{N} |\mathbf{x}_k - \mathbf{y}_k|^2}$, where $\cdot$ denotes the inner product. Parseval's Theorem ensures that $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{X}, \mathbf{Y})$. We can decompose the Euclidean distance into the sum of the magnitude distance and a non-negative term involving both magnitudes and phases:

$$
\begin{aligned}
[d(\mathbf{x}, \mathbf{y})]^2 &= \sum_{k=0}^{N-1} \|x_k - y_k\|^2 = \sum_{k=0}^{N-1} \|\rho_k e^{j\theta_k} - \tau_k e^{j\phi_k}\|^2 \\
&\overset{(a)}{=} \sum_{k=0}^{N-1} (\rho_k \cos(\theta_k) - \tau_k \cos(\phi_k))^2 + (\rho_k \sin(\theta_k) - \tau_k \sin(\phi_k))^2 \\
&\overset{(b)}{=} \sum_{k=0}^{N-1} \rho_k^2 + \tau_k^2 - 2\rho_k\tau_k \left(\sin\theta_k \sin\phi_k + \cos\theta_k \cos\phi_k \right) \\
&\overset{(c)}{=} \sum_{k=0}^{N-1} (\rho_k - \tau_k)^2 + 2 \sum_{k=1}^{N} \rho_k\tau_k \left[1 - \cos(\theta_k - \phi_k) \right],
\end{aligned}
\tag{1}
$$

where (a) is the Pythagorean theorem, (b) follows from algebraic manipulations and elementary trigonometric identities, and (c) follows by adding and subtracting $2\rho_k\tau_k$ to (b), collecting terms, and using an elementary trigonometric identity. Having expressed the Euclidean distance using magnitude and phase terms, we explore its connection with a periodic measure in the following section.

3.2 Periodic measure

We present a distance measure that can quantify the structural similarity of sequences based on *periodic* features extracted from them. The periodic measure was discussed, together with applications, in [8] and is explicated here for completeness of presentation. In this work we make the connection with euclidean distance in the frequency domain and show how to combine both in an efficient index.
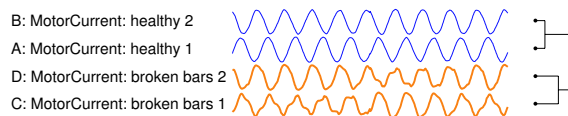

B: MotorCurrent: healthy 2
A: MotorCurrent: healthy 1
D: MotorCurrent: broken bars 2
C: MotorCurrent: broken bars 1

Fig. 2. Dendrogram on 4 sequences using a periodic measure

The introduction of periodic measures is motivated by the inability of shape-based measures such as the Euclidean to capture accurately the rudimentary human notion of similarity between two signals. For example two sequences

that are identical except for a small time shift should be considered similar in a variety of applications (Fig. 2), in spite of the potentially large euclidean distance between them. Therefore, the periodic measure loosely resembles time-warping measures, requiring only linear computational complexity, thus rendering it very suitable for large data-mining tasks.

3.3 Periodic Distance (pDist)

To assess the periodic similarity of two sequences we examine the difference of their harmonic content. We define the periodic distance between two sequences $\mathbf{x}$ and $\mathbf{y}$, with Fourier transforms $\mathbf{X}$ and $\mathbf{Y}$, respectively as the euclidean distance between their magnitude vectors:

$$[pDist(\mathbf{X}, \mathbf{Y})]^2 = \sum_{k=1}^{N} (\rho_k - \tau_k)^2.$$

Notice that the omission of the phase information renders the new similarity measure shift-invariant in the time domain, allowing for global time-shifting in $O(n)$ time (another alternative would be the use of using Time-Warping with $O(n^2)$ complexity).

In order to meaningfully compare the spectral power distribution of two sequences in the database, we normalize them to contain the same amount of energy by studentizing them (thus producing zero-mean, unit-energy sequences):

$$\hat{x}(n) = \frac{x(n) - \frac{1}{N} \sum_{i=1}^{N} x(i)}{\sqrt{\sum_{i=1}^{N} (x(n) - \frac{1}{N} \sum_{i=1}^{N} x(i))^2}}, \quad n = 1, \dots, N$$

For the rest of the paper we will assume that all database sequences are studentized (whether we are considering euclidean or periodic distance). We also remark a property of the periodic distance that will be useful to provide more effective traversal of our indexing structure.

Lemma 1 *The periodic distance is a lower bound to the Euclidean distance: $pDist(\mathbf{X}, \mathbf{Y}) \leq d(\mathbf{X}, \mathbf{Y})$.*

This lemma is proved by noting that the first sum on the RHS Equation 1 is the periodic distance, and that the second sum is non-negative.

4 Lower Bounding & Coefficient Selection

In order to efficiently incorporate a distance measure with an indexing structure, one needs to: (i) compress a sequence (dimensionality reduction) (ii) provide a lower bounding function of the original distance using the compressed object. We will show how both of these can be achieved in an effective way.

After a sequence is transformed in the frequency domain, it can be compressed by recording only a small subset of its coefficients. Therefore, $\{X_k\}_{k=0}^{k=N-1} \rightsquigarrow \{X_k\}_{k \in S}, \quad S \subset \{0, \dots, N-1\}, \quad |S| << N$. It is straightforward to show that the euclidean or periodic distance on the compressed vectors will lower bound the original distances, because they are a sum of positive numbers:

$$d(\mathbf{X}_k, \mathbf{Y}_k)_{k \in S} \leq d(\mathbf{X}, \mathbf{Y})$$

$$pDist(\mathbf{X}_k, \mathbf{Y}_k)_{k \in S} = \sum_{k \in S} (\rho_k - \tau_k)^2 \leq pDist(\mathbf{X}, \mathbf{Y})$$

The majority of data-mining work has adapted the selection of the first k coefficients for sequence compression [5], which can provide effective approximation of signals with low frequency content (e.g., stock price movement). Recent work also suggested picking the k coefficients for each sequence that preserve most of its energy [7]. High energy coefficients can provide effective sequence reconstruction, but are not necessarily suitable for data retrieval purposes (i.e. incur the least number of accesses to the original data). Consider a dataset where the k coefficients with the highest energy are the same for all sequences, and the sequences only differ in the low energy coefficients (i.e. there are some small but distinct nuances between each sequence). In this case, one needs to select the coefficients that will *discriminate better* the sequences. Generally speaking, we can capture more effectively the data differences by recording those coefficients that account for most of the data variation. With this observation in mind, we will record the k coefficients that depict that largest variance:

$$arg \max_k var(X_k^{(j)})_{j=1...m}$$

where X_k^j denotes the kth coefficient of sequence j. We compare the performance of various coefficient selection schemes with a comprehensive experiment on 40 datasets (each containing 1000 sequences of length 1024), obtained from the UCR time-series archive [3]. We perform a 1-NN leave-one-out search and estimate the pruning power of each method as given by the ratio: *(examined objects)/(total objects)*. The methods for coefficient selection that we consider are: (i) first k coefficients, (ii) coefficients with maximum energy (iii) coefficients with maximum variance.

The results attest to the superiority of the 'max-variance' method for Nearest-Neighbor retrieval, depicting an improvement in 21 out of the 40 datasets. The average improvement over the next best method is 17.17% (with a maximum of $> 80\%$ in the *darwin* dataset). Eight datasets do not exhibit any change at all, which is observed because the set of coefficients selected by the 3 methods were the same. Finally, 11 datasets depict a deterioration in the k-NN performance which however is almost negligible (never exceeding -0.5%), with the average negative improvement being -0.13%. For the remainder of the paper, we will assume that the coefficients selected from each time-series are the ones exhibiting the largest variance.

5 Index for Euclidean and Periodic distance

Instead of constructing a different index structure for each measure, we can exploit the common representation of the euclidean and periodic measure in the magnitude/phase space, as well as their lower bounding relationship, for designing a metric index structure that can simultaneously answer queries on
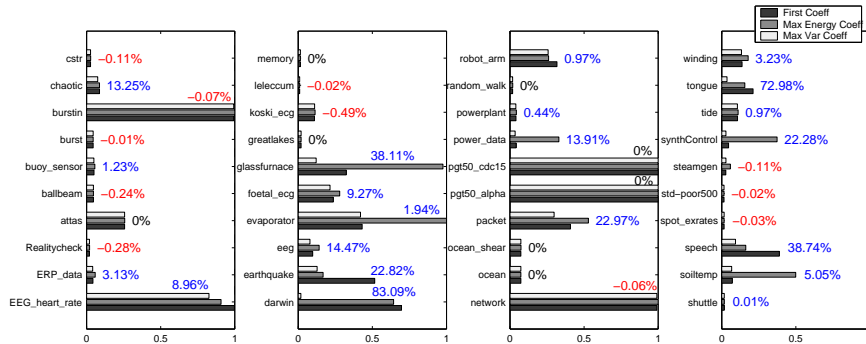
---

[3] http://www.cs.ucr.edu/~eamonn/TSDMA/

Fig. 3. Comparison of coefficient selection (smaller numbers are better). Improvement of *max-variance* method vs second best is reported next to each performance bar.

both measures. Our index structure borrows ideas from the family of metric index structures [1], recursively partitioning the search space into disjoint regions, based on the relative distance between objects. Our indexing approach has three important differences from generic metric trees: (i) only compressed sequences are stored within the index nodes, reducing the index space requirements, (ii) the index uses a different distance measure for data partitioning on each alternating level of the tree, (iii) a novel tree traversal is presented, that can answer euclidean and periodic queries in a single index scan.

5.1 MM-Tree Structure

We introduce a hybrid metric structure, the MM-Tree (Multi-Metric Tree). Similar to VP-trees, each node of the index contains a reference point (or vantage point), which is used to partition the points associated with this node into two distinct and equal sized clusters. Vantage points are chosen to be the sequences with the highest variance of distances to the remaining objects. The distances of the node objects to the reference point (sequence) are calculated, distances are sorted and the median distance μ is identified. Subsequently, any sequence associated with the examined node, is assigned to a *left* or a *right subtree*, depending on whether its distance from the vantage point is smaller or larger than the median distance. The index tree is constructed by recursively performing this operation for all subtrees.

The unique structure of the MM-Tree derives from the fact that it uses a *different* distance function to partition objects at each alternating tree level. Even depth levels (root is zero) are partitioned using the periodic distance, while odd levels utilize the euclidean (Fig. 4). We follow this construction for providing a good partitioning on both distance measures, since usage of a single distance function during the tree construction would have impacted the search on the other domain (potentially leading to search on both left and right subtrees).

Unlike other metric indexing structures, intermediate tree nodes contain the *compressed* representation of a vantage point (in addition to the median distance μ). For example, in the tree of Figure 4 the vantage points of nodes at even
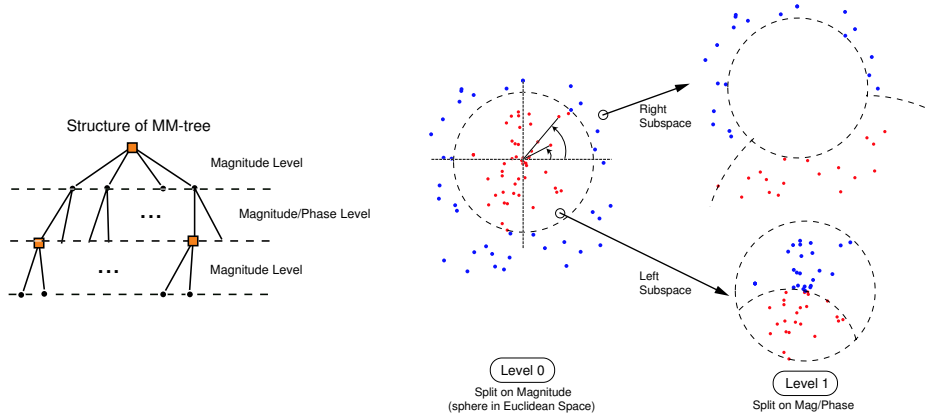
Fig. 4. MM-tree structure and space partitioning. Dotted circles/arcs indicate median distance μ.

depth are represented by the magnitudes of the preserved coefficients (and are called "P-nodes"), while those of nodes at odd depth are represented by both magnitude and phase (and are called "E-nodes"). Finally, leaf nodes contain both magnitude and phase information of the compressed data sequences.

5.2 Multiplexing Search for Periodic and Euclidean Distances

We now describe how the MM-Tree can efficiently multiplex searches in the euclidean and periodic spaces and simultaneously return the nearest-neighbors in both domains. The key idea of the search algorithm is to identify, in a single index traversal, the union of the necessary index nodes for both queries. In figure 5 we provide a pseudocode of the multiplexed search.

The combined search employs two sorted lists, $BEST_p$ and $BEST_e$, that maintain the current k closest points using periodic and euclidean distances, respectively. The algorithm also records a state, depicting whether a visited node is marked for search in the euclidean, or in the periodic domain, or both. The root node is marked for search in both domains.

Searching a P-node node. If the node is marked for search only in the euclidean domain, both subtrees are searched in the euclidean domain only. Otherwise, the algorithm computes the lower bound $LB_p(q,v)$ of the periodic distance between the vantage point of the node and the query sequence. Let r_p be the periodic distance to the farthest entry in $BEST_p$ to the query. Noting that:

$$median < LB_p(q,v) - r_p \Rightarrow median < pDist(q,v) - r_p,$$

where $pDist(q,v)$ is the periodic distance of the corresponding uncompressed sequences, we conclude that the algorithm should search the left subtree only in the euclidean domain (but not in the periodic) if $median < LB_p(q,v) - r_p$.

Searching an E-node node. If the node is marked for search in both domains, all subtrees are searched using both measures. Otherwise, the algorithm computes the lower bound $LB_e(q,v)$ of the euclidean distance between the vantage

```
/* perform 1-NN search for query sequence Q */
1NNSearch(Q) {
   // farthest results are in Best_P[0] and Best_E[0]
   Best_P = new Sorted_List(); // Modified by searchLeaf_Periodic
   Best_E = new Sorted_List(); // Modified by searchLeaf_Euclidean
   search_Node(Q, ROOT, TRUE);
}

search_Node(Q, NODE, searchPeriodic) {
   if (NODE.isLeaf) {
      search_Leaf(Q, NODE, searchPeriodic);
   } else {
      search_Inner_Node(Q, NODE, searchPeriodic);
   }
}

search_Inner_Node(Q, NODE, searchPeriodic) {
   add_Point_To_Queue(PQ, vantagePoint, searchPeriodic);
   if (NODE.E_NODE) { /* E-Node */
     if (searchPeriodic) {
       search_Inner_Node(Q, NODE.LEFT, searchPeriodic);
     } else { /* only search in euclidean space */
       if (LowerBoundEuclidean(Q, vantagePoint) - Best_E[0] < median)
         search_Inner_Node(Q, NODE.LEFT, searchPeriodic);
     }
   } else { /* P-Node */
     if (searchPeriodic) {
       if (LowerBoundPeriodic(Q, vantagePoint) - Best_P[0] < median)
         search_Inner_Node(Q, NODE.LEFT, searchPeriodic);
       else
         search_Inner_Node(Q, NODE.LEFT, FALSE);
     } else { /* only search in euclidean space */
       search_Inner_Node(Q, NODE.LEFT, searchPeriodic);
     }
   }
   search_Inner_Node(Q, NODE.RIGHT, searchPeriodic);
}

search_Leaf(Q, NODE, searchPeriodic) {
   if (searchPeriodic) search_Leaf_Periodic(Q, NODE);   // update Best_P
   search_Leaf_Euclidean(Q, NODE); // update Best_E
}
```

Fig. 5. Multiplexing euclidean and periodic search on the MM-Tree

point of the node and the query sequence. Let r_e be the euclidean distance of the farthest entry in $BEST_e$ to the query. Then, if $LB_e(q, v) - r_e > median$ the left subspace is discarded. It is also important to note that, for both types of nodes, since the vantage point is in compressed form and we use lower bounds to distances, we do not have sufficient information to discard the right subtree, unless we load the uncompressed representation of the vantage point.

A global priority queue PQ, whose priority is defined by the lower bounds of the periodic distances, is employed, in order to efficiently identify the query results. In particular, whenever the compressed representation of a data sequence v is accessed, the lower bound of the periodic distance $LB_p(q, v)$ between v and the query sequence is computed and the pair $(LB_p(q, s), s)$ is pushed into PQ. When a sequence s is popped from the PQ, the associated lower bound of the periodic distance $LB_p(q, v)$ is compared against the current $BEST_p[0]$ and $BEST_e[0]$ values. If $LB_p(q, v)$ is larger than both of those values, the sequence is discarded. However, if it is smaller than $BEST_e[0]$, the lower bound of the euclidean distance $LB_e(q, v)$ is computed and if it is larger than the $BEST_e[0]$ value,

the sequence can still be safely discarded. In all other cases, the uncompressed sequence is loaded from disk and the actual periodic and euclidean distances are computed to determine whether it belongs to any of the $BEST_p$ or $BEST_e$ lists.

6 Experiments

We demonstrate the performance and meaningfulness of results of the MM-tree for answering simultaneously queries on both euclidean and periodic distance measures. The experiments reveal that the new index offers better response time and reduced storage requirements compared to the alternative approach of using two dedicated indices, one for each distance measure.
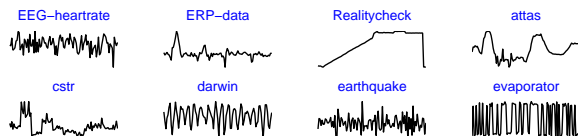

Fig. 6. Sample from the Mixed-Bag dataset

For our experiments we used the same mixture of 40 datasets that was utilized in the coefficient selection section, a sample of which is depicted in figure 6 (*MIXEDBAG* dataset). We used this dataset to create larger datasets with increasing data cardinalities of 4000, 8000, 16000 and 32000 sequences, in order to quantify the index storage requirements and its scalability. All used datasets can be obtained by emailing the first author.

6.1 Matching Results

We depict the meaningfulness of results returned by the MM-tree index, when searching in both euclidean and periodic spaces. Using the *MIXEDBAG* dataset we retrieve the 5-NN of various queries and the results are plotted in Figure 7. It is immediately apparent that the periodic measure always returns sequences with great structural affinity (i.e., belong to the same dataset). The euclidean measure returns meaningful results only when the database contains sequences that are very similar to the query (queries 1 & 3). In such cases, the periodic measure can meaningfully augment the result set of the purely euclidean matches, by retrieving time-shifted variations of the query. In the cases where there are no direct matches to the query (queries 2 & 4), the euclidean measure simply returns spurious matches, while the periodic measure can easily discover instances of the query that belong in the same class of sequence shapes.

6.2 Index Size

The MM-tree presents also the additional advantage of having reduced space requirements, compared to the alternative of maintaining 2 separate indices. Construction of two index structures (one on magnitude and the other on magnitude and phase) results in higher space occupancy, because the magnitude component of each preserved coefficient is stored twice. This is better illustrated in Figure 8, where we plot the total size occupied by the proposed MM-tree, as well as the total disk size occupied by two dedicated metric trees. As expected
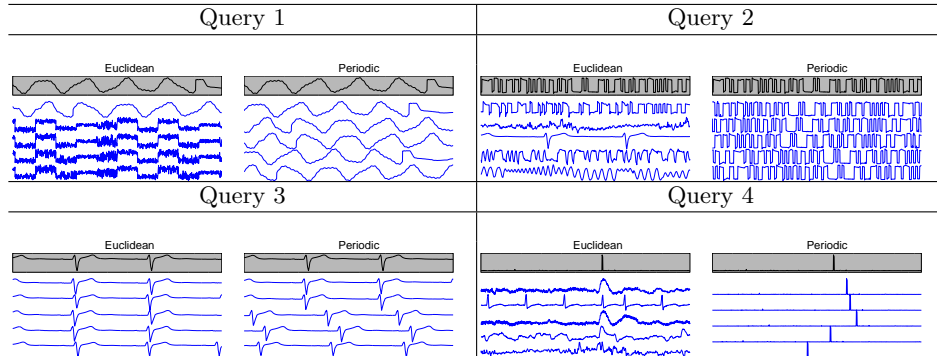
Query 1	Query 2

Query 3	Query 4

Fig. 7. 5-NN euclidean & periodic matches using the MM-tree (*MIXEDBAG* dataset)

MM-tree only requires 2/3 of the space of the dual index approach. Moreover, as shown in the next section, the information compaction that takes place during the MM-tree construction, can lead to a significant performance boost of this new hybrid index structure.
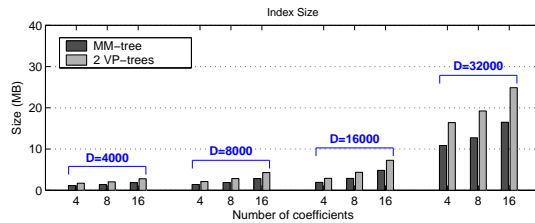

Fig. 8. Index size of MM-tree vs two index structures (euclidean & periodic)

6.3 Index Performance

Finally, we evaluate the performance of the multi-query index traversal on the MM-tree, which returns euclidean and periodic matches in a single scan. In Figure 9 we illustrate the performance gain that is realized by this novel tree traversal, for various coefficient cardinalities and kNN index searches, as captured by metrics such as the pruning power (*examined sequences/total sequences*) and the running time. The results compare the MM-tree with the dual index approach (i.e. total cost of executing one euclidean and one periodic query, each on a dedicated metric VP-tree index). Performance comparisons are conducted with VP-trees, since they have been shown to have superior performance than other metric structures, as well as R-trees [1]. The index performance charts are reported as a fraction of the cost incurred by the sequential scan of the data for the same operation. For sequential scan, the data are traversed just once while maintaining 2 priority queues, each one holding the kNN neighbors of the specific distance function. From the graph it is apparent that the performance of the MM-tree supersedes the dual index execution. The greatest performance margin is observed when retaining the largest number of coefficients per sequence, where the speedup of the MM-tree can be 20 faster than the sequential scan, while the individual metric trees are 16.5 times faster.

This final experiment displays the full potential of the proposed hybrid structure. The MM-tree due to its unique structure outperforms the dedicated metric tree structures when answering both distance queries at the same time, because it can collect the results of both distance measures in a single tree traversal.
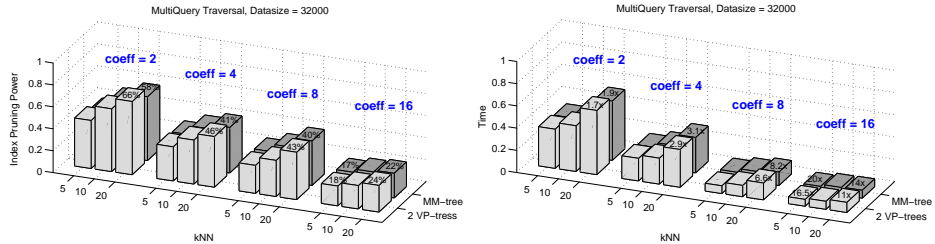

Fig. 9. Performance charts: MM-tree vs 2 VP-trees. Improvement over sequential scan is reported on top of each bar. *(Left)* Pruning power, *(Right)* Running time.

7 Conclusion

We have presented a hybrid index structure that can efficiently multiplex queries on euclidean and periodic spaces. The new index allocates disk space more judiciously compared to two dedicated index structures, and its unique structure allows for more effective tree traversal, returning k-NN matches on two distance measures in the single index scan. We hope that our system can provide the necessary building blocks for constructing powerful 'all-in-one' tools, within the scope of applications such as decision support, analysis of causal data relationships and data visualization.

References

1. A. Fu, P. Chan, Y.-L. Cheung, and Y. S. Moon. Dynamic VP-Tree Indexing for N-Nearest Neighbor Search Given Pair-Wise Distances. *The VLDB Journal*, 2000.
2. T. Ide and K. Inoue. Knowledge discovery from heterogeneous dynamic systems using change-point correlations. In *Proc. of SDM*, 2005.
3. E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. In *Proc. of SIGKDD*, 2002.
4. E. Keogh, S. Lonardi, and A. Ratanamahatana. Towards parameter-free data mining. In *Proc. of SIGKDD*, 2004.
5. D. Rafiei and A. Mendelzon. On Similarity-Based Queries for Time Series Data. *In Proc. of FODO*, 1998.
6. M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh. Indexing Multi-Dimensional Time-Series with Support for Multiple Distance Measures. In *Proc. of SIGKDD*, 2003.
7. M. Vlachos, C. Meek, Z. Vagena, and D. Gunopulos. Identification of Similarities, Periodicities & Bursts for Online Search Queries. In *Proc. of SIGMOD*, 2004.
8. M. Vlachos, P. Yu, and V. Castelli. On periodicity detection and structural periodic similarity. In *SIAM Datamining*, 2005.
9. Y. Xiong and D.-Y. Yeung. Time series clustering with arma mixtures. In *Pattern Recognition 37(8)*, pages 1675–1689, 2004.
10. B.-K. Yi and C. Faloutsos. Fast Time Sequence Indexing for Arbitrary Lp Norms. *In Proceedings of VLDB, Cairo Egypt*, Sept. 2000.