Multiresolution Similarity Search in Time Series Data: An Application to EEG Signals

Amalia Charisi[1], Fragkiskos D. Malliaros[2], Evangelia I. Zacharaki[1], Vasileios Megalooikonomou[1,3]

[1]Department of Computer Engineering and Informatics, University of Patras, Greece
[2] Computer Science Laboratory, École Polytechnique, France
[3] Center for Data Analytics and Biomedical Informatics, Temple University, PA, USA

{charisa, vasilis}@ceid.upatras.gr, fmalliaros@lix.polytechnique.fr, ezachar@upatras.gr

ABSTRACT

Time series constitute a prevalent data type that arise in several diverse disciplines (e.g., biomedical data, sensor data, images, video data), and therefore analyzing time series is a significant task with a plethora of important applications. In this paper, we study the general problem of similarity search in time series databases and we propose a novel multiresolution indexing (i.e., representation) and retrieval method for time series similarity search. Our approach is motivated by the idea that if we examine a time series at different resolution levels, we could possibly acquire further insights about the data. The proposed algorithm adopts a combined, two-step pruning (filtering) strategy to further reduce data dimensionality by discarding irrelevant time series (i.e., false alarms). At a first level, the time series are represented by line segments and filtered by the triangular inequality property. Then, a Vector Quantization like scheme is applied to encode data and thus to reduce dimensionality.

We test and demonstrate the performance of the proposed method, analyzing EEG time series data for retrieval of one of the constituent brain waveforms in EEG recordings, the K-complex, but the method can as well be applied for retrieval of other patterns of interest in time series analysis. The automatic detection and categorization of the EEG patterns will allow the advanced correlation analysis of large amounts of data and will lead to advanced decision making capabilities assisting diagnosis by medical professionals.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database Applications—*Data mining*; H.2.4 [**Information Systems**]: Systems—*Multimedia databases*

PETRA '13, May 29-31, 2013, Island of Rhodes, Greece.

General Terms

Algorithms, Experimentation

Keywords

Time series, Similarity search, EEG signals, Assistive Environments

1. INTRODUCTION

A time series is a sequence of real numbers taken by the observation of a variable over time. Time series can be found in several domains including medical, financial and business. For this reason, time series analysis and its applications have received a lot of attention over the last years.

An important application of time series data analysis involves brain and body activity monitoring using a variety of modalities, such as EEG, ECG, EOG, EMG. EEG is the major modality that is used in the literature for several purposes including diagnosis and seizure detection in epilepsy studies. Specifically in such applications, it is also crucial to analyze time series of sleep EEG due to the fact that several sleep factors, mechanisms, or conditions were found to aggravate seizures [12].

Here, we deal with the problem of similarity search in time series databases. A similarity search query (range query) is defined as follows: given a time series database T and a query (q, r), where r represents a threshold, find all the time series $t \in T$ that their distance from q is smaller than r. The easiest but the most expensive way to answer such a query is to scan the database sequentially. To improve the performance of similarity searching, several dimensionality reduction techniques have been suggested in the literature; these techniques, transform time series into a lower dimensional space and then process more efficiently the similarity search task in this reduced space.

In this paper, we propose MR-PVQ (MultiResolution Piecewise Vector Quantization), a representation and similarity search method for time series datasets. In general, it would be helpful if we are able to examine a time series at different resolution levels, in order to acquire further insights about the data. Our method uses line segments to represent time series and then a dimensionality reduction technique based on a vector quantization method. The above scheme is applied in multiple resolution levels, discarding the non-qualifying sequences in every level.

The main contributions of the paper are in respect to the following aspects:

- *Similarity search method*: We propose MR-PVQ, a multiresolution approach for indexing and retrieval of time series data. MR-PVQ is composed by a vector quantization like method for the representation of time series, combined with an enhanced filtering strategy– improving accuracy.

- *EEG signal analysis*: We apply the proposed method to analyze brain recordings from sleep EEG. The sleep microstructure, characterized by the EEG signal building blocks (K-complexes, spindles, microarousals etc.) [11] are known to be involved in mechanisms underlying the expression of seizures and epileptiform discharges. Thus the automatic retrieval of interesting brain patterns, such as the K-complex, can facilitate correlation analysis and lead to new medical knowledge discovery.

The rest of the paper is organized as follows: in Section 2 we present related work. Section 3 provides useful information about the concepts that will be used throughout the paper. Then, in Section 4 we describe the proposed method, and we evaluate its performance in Section 5. Finally, in Section 6 we provide some concluding remarks and discuss interesting extensions for future work.

2. RELATED WORK

A large body of the related literature has focused on the broader problem of time series analysis and more specifically on similarity search in time series databases. Several representation schemes for dimensionality reduction have been proposed, as well as measures for capturing the similarity between time series. Examples of representation techniques include the Discrete Fourier Transform [1, 6], the Discrete Wavelet Transform [4], and the Symbolic Aggregate Approximation [14]. Using such techniques, the time series can be represented with reduced dimensionality. As we will discuss in Section 3, the representation techniques should guarantee the *lower bounding* property, i.e., the relative distance of two objects in the representation space should be preserved.

Other approaches for similarity search include embedding-based [2] and query-by-humming [13] methods. Furthermore, a plethora of similarity (or distance) measures have been introduced and applied, including the Euclidean distance (see definition 3) and the well-known Dynamic Time Warping [9]. A more detailed comparative presentation is given in [5].

In [19], the authors proposed MVQ, a multiresolution vector quantization approach, where the representation of times series is based on the *appearance frequency* of every codeword. While our proposed MR-PVQ method shares some common features with MVQ, the main difference is that we use the *index of the closest codeword*, to encode and represent the time series. Another multiresolution approach for time series retrieval, is the one presented in [16]. The method uses polynomials in several resolutions for the approximation of time series, with main focus to improve the time efficiency of the algorithm.

3. PRELIMINARIES AND BACKGROUND

In this section, we provide the necessary definitions and background that will be used throughout the paper. We begin by defining the basic data type of times series and then we proceed with the description of the two main concepts upon which our approach is built: (i) Generic Multimedia Indexing and (ii) Piecewise Vector Quantized Approximation.

Definition 1 (Time series dataset). We assume that there is a dataset T of M time series (or time sequences):

$$T = \{t_1, t_2, \ldots, t_M\}.$$

Each time series is an ordered collection of k real values:

$$t_m = (t_{m,1}, t_{m,2}, \ldots, t_{m,k}),\ m = 1, 2, \ldots, M.$$

Definition 2 (Subsequence of a Time series). A subsequence $t_{m,(s,w)}$ of length $w < k$ of a time series t_m, is defined as a part of the original time sequence which starts from position s and formed by consecutive elements:

$$t_{m,(s,w)} = (t_{m,s}, t_{m,s+1}, \ldots, t_{m,s+w-1}), \quad \text{where}$$

$$1 \leq s \leq k - w + 1.$$

Definition 3 (Euclidean distance). Given two time series $t_1 = t_{1,1}, t_{1,2}, \ldots, t_{1,k}$ and $t_2 = t_{2,1}, t_{2,2}, \ldots, t_{2,k}$ of the same length k, their Euclidean distance d is defined as:

$$d(t_1, t_2) = \sqrt{\sum_{i=1}^{k} |t_{1,i} - t_{2,i}|^2}.$$

3.1 Generic Multimedia Indexing

The Generic Multimedia Indexing (GEMINI) constitutes a framework for fast similarity search in time series databases (generally, the method can be applied in any domain where data is represented as time series) [1, 6]. Assuming that we are given a time series dataset T and a query time series q, how can we efficiently find those objects from the dataset that are similar to the query?

The first step of GEMINI is to define a similarity (or dissimilarity) measure between two data objects (time series in our case). Given two objects O_1 and O_2 their distance is denoted by $D(O_1, O_2)$. Such a distance measure could be Euclidean distance, which is computed as suggested to Definition 3.

Having defined a distance function, the goal of a similarity search method is to find the objects that are similar to the query. The naive solution of applying sequential scanning, i.e., computing the distance between the query and every object in the database, may not be feasible, because the computational cost of the distance function may be large, which in addition to a large dataset may lead to a computational bottleneck.

In order to overcome the above problem and improve the performance of similarity search, the second step of GEMINI is to define a dimensionality reduction technique for the data. That is, reducing the size (i.e., length) of the data, one is able to efficiently discard many of the time series which are not similar to the query, and therefore efficiently reduce the number of potential "false alarms". This could be achieved by mapping the objects of our database to a new representation space (called feature space), where the distance

function could be computed efficiently. The mapping to the new space should preserve the relative distances between objects in the original space, otherwise "false dismissals" could possibly break down the effectiveness of the similarity search method. In other words, the distance between two objects in the feature space should be underestimated with respect to the corresponding distance in the original space (this property of the distance function is known as *lower bounding*).

3.2 Piecewise Vector Quantized Approximation

In this paragraph, we describe the Piecewise Vector Quantized Approximation (PVQA) [15, 18] method, a dimensionality reduction technique that relies on the widely known Vector Quantization [7] for the encoding of time series. As we will present shortly, our approach extends this method to perform more accurate similarity search.

The PVQA method consists of the following three steps:

- Construction of a codebook.
- Segmentation of each time series into subsequences.
- Encoding each subsequence with the index of the most similar codeword in the codebook.

As mentioned above, the Vector Quantization (VQ) method [7] is utilized to represent a time series and calculate the similarity among them. VQ was originally applied for data compression based on the principle of block coding. The algorithm uses a codebook C that divides the dataset into encoding regions S_n. The codebook consists of codevectors (or codewords) c_n, where each codevector is used to represent part of the dataset. For the learning process of VQ, a training dataset of time series is used.

3.2.1 Codebook generation

Next, we will briefly describe the main steps of a clustering algorithm, named Generalized Lloyd Algorithm, for producing the optimal codebook [7].

The algorithm starts by generating optimal codebook of size N through a recursive procedure for the training set of time series. It starts by using an initial codebook, where the codeword is the centroid of the training set. The cells as well as the codewords are doubled in every repetition of the algorithm, until the desired number of the codevectors (codewords) are obtained.

The optimal codebook is determined by two conditions and a distortion function (i.e, mean-square error) between each time series and its closest codevector (codeword). The nearest-neighbor condition is the one that finds the encoding region, whereas the codebook is determined by the centroid condition:

- **Nearest Neighbor Condition**:
  Let $C = \{c_1, c_2, \ldots, c_N\}$ represents the codebook, where N is the number of codevectors. The optimal encoding region S_n should consist of all vectors that are closer to c_n than any of the other codevectors.
- **Centroid Condition**:
  For a given encoding region S_n, its optimal codeword c_n is the average of all the vectors that participate in the training set and belong to the corresponding encoding region.

The algorithm stops when the change of the distortion between two consecutive iterations is less than a given threshold.

3.2.2 Time-series encoding

As we mentioned above, the PVQA method splits each time series in multiple segments. Assuming that a time series t_m is partitioned into ℓ segments $t_m = (t_{m,s_1}, t_{m,s_2}, \ldots, t_{m,s_\ell})$, PVQA applies a distance function to find the closest codevector (codeword). To represent each segment of the time series, the index of the most similar codeword is used. If the approximation of the original time series is required, one can concatenate corresponding codevector in the appropriate segment t_{m,s_ℓ}.

4. PROPOSED METHOD

In this section, we introduce MR-PVQ, a multi-resolution indexing and retrieval method for time series similarity searching.

4.1 Overview

Our MR-PVQ (MultiResolution Piecewise Vector Quantization) method, extends the PVQA dimensionality reduction technique presented previously, in *multiple resolutions*. The lower the resolution level is, less number of segments are used to encode time series data. To achieve this, we propose to use a two-level pruning (filtering) strategy in order to decrease the number of objects that will be encoded during the next step. In each level of this strategy, we use an approximation function for the time series. The first filter applies a property that all the indexing schemes require to hold, triangular inequality discarding the non-qualifying objects. Here, the time series data are approximated with first degree polynomials. The second filtering level is based on the lower bounding lemma of the GEMINI algorithm and a VQ technique is used to encode the testing set.

Our work is motivated by the observation that although global information of a time series is kept after the encoding by PVQA in one resolution, important local information of the time series is lost. The idea of using multiple resolution levels, gives us the opportunity to retain both global and local information. Combining this fact with the representation of the time series using polynomials and the application of the previously described filters that enhance the pruning power of the algorithm, can improve substantially the performance of similarity search in time series databases. For example, Fig. 1 depicts the time series returned for a specific query (top fig.), using one (middle fig.) or multiple resolution levels (bottom fig.). As we can observe, the time series retrieved using multiple resolution levels, is more similar to the query.

4.2 Proposed MR-PVQ Method

The proposed MR-PVQ approach, constitutes a multiresolution representation and retrieval method. In each time series, two different representation methods along with their corresponding filters are used, in order to discard time series that correspond to false alarms. The first representation applies line segments to represent the time series; this constitutes a very simple and efficient scheme with interesting properties. It mainly discards the non-qualifying objects of lower resolution and is used as the first level of our pruning

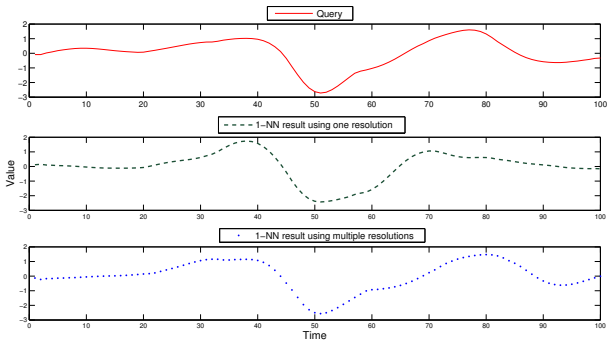


Figure 1: 1-NN results for a specific query (top), using one resolution level where the time series is splitted into 10 segments (middle) and multiple resolution levels (bottom). Observe that the time series returned by the multiresolution approach is more close to the query.

strategy, in order to decrease the number of time series that will be encoded by the PVQA method.

To generate the final answer set, the algorithm begins from lower resolution and continues to higher ones until the resolution levels run out. Lower resolution means that less segments are used to split a time series. As a result, less codewords are used to encode a time series and only some basic global information about the time series remains. As the resolution level increases, more segments are used to split the original time series and more local information (details) remains after its encoding by several codewords. Our multiresolution approach speeds-up the similarity search, discarding the objects that have basic differences with the query from the beginning (lower resolution) and sending only a subset of the dataset to the next level. The outline of the MR-PVQ method is described as follows:

For each resolution level i:

1. Split the time series into ℓ segments.
2. Represent each segment with a first degree polynomial.
3. Discard the time series that are not close to the query using the above representation (first level pruning strategy).
4. Encode the remaining time series using PVQA.
5. Discard the time series that correspond to false alarms (second level pruning strategy).
6. Move on to the next resolution level and repeat steps 1-6.

A schematic representation is also shown in Fig. 2.

4.2.1 *First level pruning strategy: degree one polynomial segment*

As mentioned above, our method applies two filters to enhance its pruning power. The first filter proposed by [16], comprises by a meaningful combination of the triangular inequality property with the representation of a time sequence as a line segment.

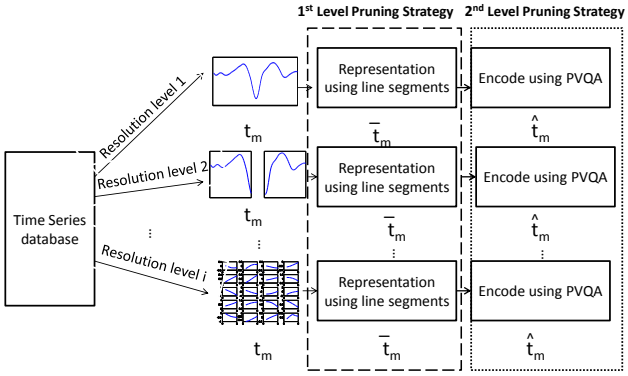


Figure 2: Representation scheme applied in MR-PVQ.

In more detail, we use polynomials of first degree to approximate a time series t_m in several resolutions. Let $\mathbf{R}^k$ be the k-dimensional space of a time series. The time series is partitioned into ℓ segments each of length s, and a polynomial of first degree is used to represent them. The approximation error between the polynomial and the segment of the original time series is minimized. The time series $\bar{t}_m$ that is produced by the projection of the points of all the segments of time series t_m on the line, is the approximation of the original time series.

To discard the non-qualifying objects, an exclusion condition is applied. Let q be a query and its threshold r. $\bar{q}$ and $\bar{t_m}$ are the best approximations of q and t_m, respectively, where t_m is a time series in the dataset. Let d be the distance function used, between two time series. Here, we used Euclidean distance as a similarity function. By applying the triangular inequality property obtain:

$$d(\bar{q}, t_m) \leqslant d(q, t_m) + d(\bar{q}, q), \ \forall t_m \in T \tag{1}$$

Because $\bar{t}_m$ and $\bar{q}$ are the best approximation for the time series t and q respectively, the distance between the original time series and its representation is minimized. We can conclude the following:

$$d(\bar{q}, t_m) > d(\bar{t}_m, t_m) \tag{2}$$

$$d(\bar{q}, t_m) > d(q, \bar{q}) \tag{3}$$

For each of the above relations, Eq. (1) can be written as follows:

$$d(\bar{t_m}, t_m) < r + d(q, \bar{q}) \tag{4}$$

$$d(q, \bar{q})) < r + d(\bar{t}_m, t_m) \tag{5}$$

obtaining the below relation:

$$|d(q, \bar{q})) - d(\bar{t}_m, t_m)| < r \tag{6}$$

The time series that are not obey inequality (6) should be discarded. Extending the above relation in several resolution

levels, we can obtain our first filter. Let $\bar{q}$ and $\bar{t}_m$ be the approximated time series in every resolution level of q and t_m respectively, our first filter is given by:

$$|d(q, \bar{q}) - d(\bar{t}_m, t_m)| > r \quad (7)$$

4.2.2 Second level pruning strategy: representation and dimensionality reduction

Since dimensionality reduction has always been an important component for fast similarity search in large time series databases, we propose to apply the Piecewise Vector Quantized Approximation for our second level pruning strategy; that way, the dimensionality of the time series can be reduced in several resolution levels. Here, the dataset of the initial time series is divided in two sets, the training and the testing set. Both the datasets are splitted into segments; the higher the resolution level, the more segments are used to split the initial time series. The training set is used for the codebook generation process whereas, the testing set is used in the query process.

For the codebook generation, we apply the GLA algorithm and we produce a different codebook for every resolution level. The generated codewords have length that vary in every level (according to the length of a segment of this resolution). The number of levels used is determined by the length of the original time series. In every resolution level, the length of the segment used to split the original time sequence is reduced by half.

After the codebook generation process, each segment of the testing set is encoded by its closest codeword. To guarantee that our method is not affected by false dismissals, we propose the use of a lower bounding distance for computing the similarity between two encoded time series. Following the GEMINI approach, those time series that their distance in the original space is larger than the one in the feature space, are discarded. For the distance computation Euclidean distance is used.

Let $\hat{q}$ and $\hat{t}_m$ are the approximation of q and t_m respectively, in every resolution level. The lower bounding distance between a query $\hat{q}$ and a time series $\hat{t}_m$ is defined as the distance between the corresponding codevectors of $\hat{t}_m$ and $\hat{q}$[1]:

$$d_{MR\_PVQ} = \sqrt{\sum_{j=1,\ldots,\ell} \hat{d}(\hat{t}_{m_j}, \hat{q}_j)}. \quad (8)$$

The second filter is given by the following relation:

$$d_{MR\_PVQ} > r. \quad (9)$$

4.2.3 Query time

In query time, our algorithm begins with the lower resolution level representing the time series using polynomials and applying the first filter. Then the time series of the testing set that do not satisfy the first exclusion condition, are encoded using PVQA and their lower bounding distance is calculated. The second exclusion condition is calculated and the time series that are not discarded move on to the encoding of the next resolution level. The algorithm recursively proceeds to the next higher resolution level following the above procedure and stops when the resolution levels run out. A schematic illustration is depicted in Fig. 3.

[1]The proof of the lower bounding property can be found in [18].

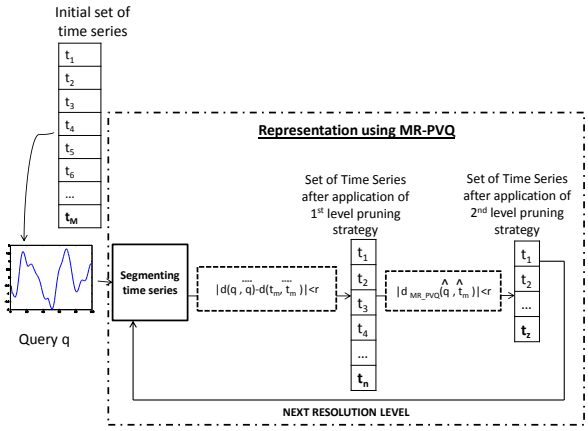


Figure 3: Query answering by MR-PVQ.

5. EXPERIMENTS

5.1 Dataset Description

The method has been applied to the detection of K-complexes that are important constituent brain waveforms in electroencephalography (EEG). The K-complexes have been suggested to engage in information processing, sleep protection [8] and memory consolidation [3] and also they are key features for sleep scoring [17].

5.1.1 EEG Data

The data used in this work were acquired during a whole-night sleep EEG of a healthy volunteer without a history of neurological or psychiatric disorder, or sleep disorder. Nocturnal sleep was recorded using 58 EEG tin electrodes positioned according to the extended international $10-20$ system on an electrode cap, ear lobe referenced. For the current study a single channel (the FZ electrode position) was used. Manual cursor marking offered by Scan software was used in order to place time-markers over the events under study. The K-complex was visually identified as a > 500 ms well-delineated negative sharp wave usually followed by a positive phase that stands out of the EEG background [8]. Singular (without another K-complex or slow wave activity immediately preceding or following) generalized (distinguishable in the EEG across all the midline electrodes) spontaneously occurring KCs were selected and visually marked at the peak of the negative phase [10].

5.1.2 Preprocessing

The raw EEG recordings were first resampled at a sampling frequency of 200Hz and then low-pass filtered to remove high frequency noise. We used a Dolph-Chebyshev window (with a filter order of 100) and a cut-off frequency at 5Hz to include the frequency range of K-complexes. Subsequently baseline correction was performed. Specifically, EEG signals are undergoing slow shifts over time during the recording, such that the zero level (DC component) might differ considerably across channels. The DC component was extracted by calculating the mean signal in overlapping seg-

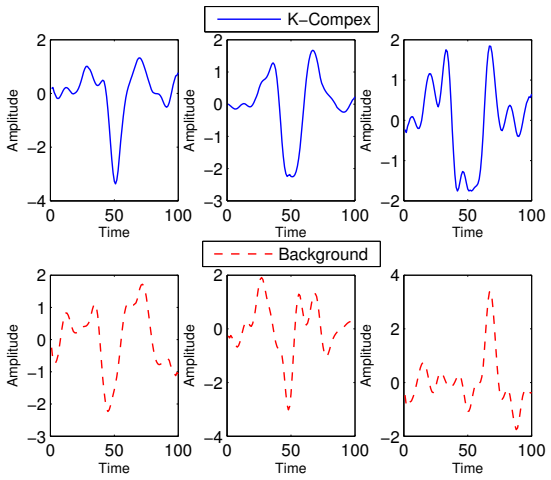


Figure 4: Examples of K-compex (top) and background (bottom) in EEG signals.

ments and then smoothing this stepwise constant DC component by using a moving average filter. The smoothed DC component was afterwards subtracted from the original signal. The KC training samples have been created by extracting the signal in a ± 1sec interval around the expert-defined markers. The training samples for the negative class have been extracted randomly from the background not in close proximity (> 2 sec) to KC markers. All samples have been downsampled to form a 100-dimensional feature vector.

We conducted experiments with 555 time series of the above dataset. The time series belong to two classes: K-complex and background. To statistically analyse the dataset, we used 5-fold cross validation. This is a technique, where the dataset is divided into five subsets. Each time, one of the five subsets is used as a test set and the remaining four subsets are put together to form the training set. Then the average result of all the five trials is computed.

To avoid the effects of scaling and shifting in analysis, the dataset was also preprocessed using zero-mean normalization before we apply the proposed method. Each time series t_m is normalised $t_m = \frac{t_m - \bar{t}_m}{\sigma(t_m)}$, where $\bar{t}_m$ is the mean value of t_m and $\sigma(t_m)$ is the standard deviation of t_m. In the training phase, we used the value 0.01 as a threshold on the fractional drop of the distortion for the GLA algorithm. In our method, we used polynomials of first degree to approximate the time series. As queries, we used the time series of the testing set. The r is chosen properly to give as a final answer set the number of requested objects. In the results, we report an average of the returning accuracies for each query.

MR-PVQ uses as parameters the number of resolution levels rl and the number of codewords n that compose the codebook C. If we consider k the length of the original time series that participate in the dataset, the resolution level rl could be chosen as $rl = \log k$. The codeword length (ℓ) for each level is chosen as follows: At the first level, each time series is treated as a whole, at the second level is splitted in two segments, at the third level is divided in four segments and in the next resolution level, the length of the segments is the half of the previous one $\ell_{rl} = \frac{\ell_{rl-1}}{2}$.

5.2 Results

In time series analysis, best match retrieval is the one of the most common and important applications. Here, we conducted experiments to evaluate the effectiveness of our method. We address the following issues:

(a) How accurate MR-PVQ is?

(b) How does it perform (i) compared to another state-of-the-art multiresolution approach, MVQ and (ii) in the case of using only a single level of MR-PVQ?

As k-nearest neighbor search, we can define the following: given a query sequence q, find the best k matches in the database. We will use the following evaluation metric to measure the performance of different approaches:

$$Accuracy = \frac{|retrieved\_seq(q) \cap std\_class(q)|}{k} * 100\%. \quad (10)$$

To define the above equation, we will use as *std_class*, the class that the query belongs to. The accuracy is defined as the percentage of the requested objects that fall in the query's class. Higher percentage of requested sequences from the same class should be considered as a better retrieval result. In our experiments five resolution levels are used. For each resolution level, the length of the time series is shown in Table 1.

Level	Codeword length
1	100
2	50
3	25
4	10
5	5

Table 1: Length of codewords.

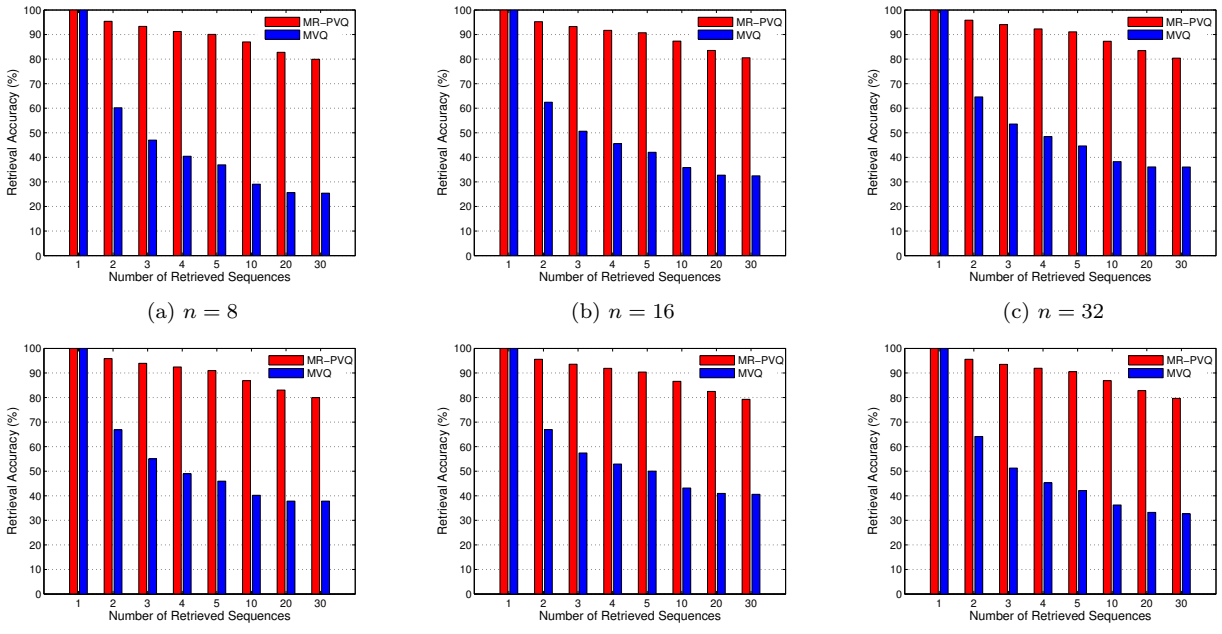
We conducted several experiments using the number of codewords n that compose the codebook.

5.2.1 *Fixed number of codewords in each resolution level*

In this section we report results from an experiment we conducted on how the retrieval accuracy is affected by the number of codewords that compose the codebook. We compare the accuracy to that of an state-of-the-art multiresolution approach, MVQ that uses VQ for the encoding of time series.

Figures 5 (a)-(e) show the results for different number of codewords used to compose the codebook. The codeword number is fixed in each resolution level. As we can see, the retrieval accuracy decreases as the number of requested sequences increases, while the accuracy is independent of the number of codevectors used by VQ to divide the Euclidean space. As a result, we do not need a large number of codewords to extract an accurate result.

In comparison to MVQ – which encodes a time series with the appearance frequency of every codeword regardless the order of the segment in the time series – our method achieves higher accuracy. This mainly happens due to the fact that, the proposed encoding scheme does not affect the order of the individual elements within the time series; the relative position of each element $t_{m,i},\ i = 1, \ldots, k$ is retained.



(a) $n = 8$ (b) $n = 16$ (c) $n = 32$

(d) $n = 64$ (e) $n = 128$ (f) Varying n

Figure 5: Retrieval Accuracy for different values of requested sequences ($rs = 1, 2, 3, 4, 5, 10, 20, 30$) for different number of codewords n.

Another reason is that, in every resolution level we discard the time series that are not close to the query.

5.2.2 *Varying the number of codewords*

Here, we demonstrate the retrieval results when the number of codewords changes as the resolution number increases and compare it to MVQ. Table 2 shows the codewords used in each resolution level.

Level	Number of Codewords
1	8
2	16
3	32
4	64
5	128

Table 2: Number of codewords in each resolution level.

Figure 5 (f) shows the results for a varying number of codewords. As we can see, the retrieval accuracy decreases as the number of requested sequences increases. Furthermore, the comparison with MVQ shows that our method outperforms, for reasons similar to the ones that have been reported above.

One can observe that the results from this experiment do not differ significantly from the ones reported on the previous experiment. This happens because there is no loss of information during the process of codebook generation. Although in every resolution level the training set is splitted into multiple segments, all the produced segments are used to generate the codebook. Therefore, the resulted codewords constitute a representative set of key-sequences for the whole dataset. In both experiments, the number of the produced codewords remains relatively small with respect to the number of the training set.

5.2.3 *Varying the resolution levels*

In this section we report results from an experiment we conducted on how the retrieval accuracy is affected by the the resolution levels used. In this experiment, the number of codewords varies in every resolution level as it is shown in Table 2. We compared the multiresolution scheme with the case where a single resolution level is used for the experiment. The level [10000] (i.e., level 1) is the binary form of the resolution level that is involved to the distance calculation for the case where a single level of MR-PVQ method is used. To conduct the experiment, we followed the same process as in the MR-PVQ, where in every resolution level, the two representation methods and the two filters are used to encode the time series and discard the non-qualifying objects accordingly. The results are illustrated in Table 3.

Level	$rs = 1$	$rs = 5$	$rs = 10$	$rs = 20$
[10000]	0.88	0.86	0.80	0.78
[01000]	0.95	0.81	0.77	0.75
[00100]	1.00	0.83	0.76	0.70
[00010]	1.00	0.89	0.85	0.81
[00001]	1.00	0.89	0.85	0.82
[11111]	1.00	0.90	0.86	0.83

Table 3: Retrieval accuracy in each resolution level.

As it can be seen from the results, when all the resolution levels participate in the result the accuracy is slightly better than using only one resolution level. The production of the codebook is an important factor for the final answer set.

During the training phase, there is no loss of information because all the segments of the time series of the training set participate for the generation of the codebook. As a result,the query is classified correctly and even with only one resolution level our method extracts accurate results.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a method for time series similarity searching using a multi-resolution scheme of indexing and retrieval. The proposed MR-PVQ approach is an extension of the dimensionality reduction technique, PVQA. Our method uses two representation schemes for its data. The dataset is processed in multiple resolution levels, where in each level a different number of segments is used to split the original time series. The first level of representation uses first degree polynomials to represent a time series and triangular inequality property to discard the time series that are not close to the query. The second level of the proposed technique uses PVQA in several resolutions to reduce the dimensionality introducing a lower bounding distance to avoid false dismissals.

The experiments we performed on a real dataset to demonstrate the utility and the efficiency of our method, showed that it generally compares favorably to previously proposed multiresolution techniques that use VQ for the encoding of time series. Besides good performance, MR-PVQ also has the ability to encode the time series with a representative subset of key-sequences in every resolution level.

We applied the proposed PR-PVQ similarity search method over EEG waveforms that contain K-Complexes and background information. The advanced similarity analysis on such data allows the integrative interpretation and can bring better understanding of the available patient information and can facilitate the differential diagnosis of epilepsy or related disorders, as well as treatment evaluation.

Here, we demonstrated an application where the data is stored in one place (e.g., a healthcare center, a hospital etc.). However, in real world settings, similar data is geographically distributed across several healthcare institutions. The developments in networking and computing technologies provide the challenge to extend such a method (i.e., retrieving similar objects from geographically dispersed time series databases) in a distributed manner, that potentially could allow clinicians to facilitate diagnosis in a more effective way. Furthermore, as another possible future direction, we plan to extend our method for *real-time* detection of patterns of interest in EEG data (e.g., K-Complexes, spindles).

7. ACKNOWLEDGMENTS

This study was partially funded by a University of Patras "Karatheodori" grant and by the European Commission under the Seventh Framework Programme (FP7/2007-2013) with grant ARMOR, Agreement Number 287720.

8. REFERENCES

[1] R. Agrawal, C. Faloutsos, and A. N. Swami. Efficient similarity search in sequence databases. In *FODO*, pages 69–84, 1993.

[2] V. Athitsos, P. Papapetrou, M. Potamias, G. Kollios, and D. Gunopulos. Approximate embedding-based subsequence matching of time series. In *SIGMOD*, pages 365–378, 2008.

[3] S. Cash, E. Halgren, N. Dehghani, and et al. Human K-Complex Represents an Isolated Cortical Down-State. *Science*, 324:1084–87, 2005.

[4] K.-P. Chan and A. W.-C. Fu. Efficient time series matching by wavelets. In *ICDE*, pages 126–133, 1999.

[5] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proc. VLDB Endow.*, 1(2):1542–1552, 2008.

[6] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *SIGMOD*, pages 419–429, 1994.

[7] A. Gersho and R. M. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, Norwell, MA, USA, 1991.

[8] P. Halász. K-complex, a reactive EEG graphoelement of NREM sleep: an old chap in a new garment. *Sleep Med Rev.*, 9(5):391–412, 2005.

[9] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7:358–386, 2005.

[10] V. Kokkinos and G. Kostopoulos. Human non-rapid eye movement stage II sleep spindles are blocked upon spontaneous K-complex coincidence and resume as higher frequency spindles afterwards. *J Sleep Res.*, 20(1 Pt 1):57–72, 2011.

[11] V. Kokkinos, A. Koupparis, M. Stavrinou, and G. Kostopoulos. The hypnospectrogram: An eeg power spectrum based means to concurrently overview the macroscopic and microscopic architecture of human sleep. *Journal of Neuroscience Methods*, 185(1):29–38, 2009.

[12] G. Kostopoulos. Brain mechanisms linking epilepsy to sleep. *Encyclopedia of Basic Epilepsy Research*, 7:1327–1336, 2009.

[13] A. Kotsifakos, P. Papapetrou, J. Hollmén, D. Gunopulos, and V. Athitsos. A survey of query-by-humming similarity methods. In *PETRA*, pages 5:1–5:4, 2012.

[14] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Min. Knowl. Discov.*, 15(2):107–144, 2007.

[15] V. Megalooikonomou, G. Li, and Q. Wang. A dimensionality reduction technique for efficient similarity analysis of time series databases. In *CIKM*, pages 160–161, 2004.

[16] M. M. Muhammad Fuad and P.-F. Marteau. Multi-resolution approach to time series retrieval. In *IDEAS*, pages 136–142, 2010.

[17] A. Rechtschaffen and A. Kales. A manual of standardised terminology and scoring system for sleep stages in human subjects. *U.S. Government Printing Office, Washinghton, DC*, 1968.

[18] Q. Wang and V. Megalooikonomou. A dimensionality reduction technique for efficient time series similarity analysis. *Inf. Syst.*, 33(1):115–132, 2008.

[19] Q. Wang, V. Megalooikonomou, and C. Faloutsos. Time series analysis with multiple resolutions. *Inf. Syst.*, 35(1):56–74, 2010.