Repeating Patterns as Symbols for Long Time Series Representation

Jakub Sevcech, Maria Bielikova

Faculty of Informatics and Information Technologies,
Slovak University of Technology,
Ilkovičova 2, 842 16 Bratislava, Slovakia
{jakub.sevcech, maria.bielikova}@stuba.sk

Abstract

Over the past years, many representations for time series were proposed with the main purpose of dimensionality reduction and as a support for various algorithms in the domain of time series data processing. However, most of the transformation algorithms are not directly applicable on streams of data but only on static collections of the data as they are iterative in their nature. In this work we propose a symbolic representation of time series along with a method for transformation of time series data into the proposed representation. As one of the basic requirements for applicable representation is the distance measure which would accurately reflect the true shape of the data, we propose a distance measure operating on the proposed representation and lower bounding the Euclidean distance on the original data. We evaluate properties of the proposed representation and the distance measure on the UCR collection of datasets. As we focus on stream data processing, we evaluate the properties and limitations of the proposed representation on very long time series from the domain of electricity consumption monitoring, simulating the processing of potentially unbound data stream.

Keywords:
Time Series Representation, Symbolic Representation, Stream Processing, Lower Bound

This paper is based on J. Sevcech and M. Bielikova, Symbolic Time Series Representation for Stream Data Processing published in the proceedings of the the 1st IEEE International Workshop on Real Time Data Stream Analytics (part of the BigDataSE-15 conference) [1].

1. Introduction

Many different time series representations were proposed over the past years [2]. However, only small portion of them is applicable on stream data processing as most of the transformation procedures are iterative in their nature or they require some sort of statistical information about the whole dataset.

Our primary motivation is to propose a time series representation applicable in stream data processing, in domains where very long (potentially infinite) time series are produced and where repeating shapes are occurring in the course of the time series. The primary application we had in mind when we proposed the representation is forecasting and anomaly detection in data such as counting metrics running on production or consumption data streams, where strong seasonal patterns are occurring. Our prime requirement for such a time series representation is incremental procedure of the data transformation and symbolic representation of reoccurring patterns.

In our work, we are most interested in symbolic representations of equally spaced time series as they enable the application of methods that are not directly applicable on real-valued data [3] such as Markov models, suffix trees or many algorithms from the domain of text processing. An example of such representation is SAX [3] – one of the most widely used time series representations. Similarly to the majority of other representations, however, transformation into the SAX representation is iterative and cannot be directly applicable to stream data processing as it requires statistical information about the whole transformed dataset. Examples of other symbolic time series representation can be found in [3, 4, 5, 6], but they all share the same limitation, stream data cannot be directly transformed into these representations.

The representation we propose is based on the symbolic time series representation used in [4] for rule discovery in time series. Clusters of similar subsequences are used as symbols in the transformation of time series into sequences of symbols. This work was influencing many researchers for several years, but they found its two major limitations:

- It is iterative due to the K-means algorithm used for cluster formation.

- It has been proved that the transformation process produces meaningless clusters that do not reliably reflect the data they were formed from [7].

In our work, we address both of these limitations. To be able to transform potentially infinite data streams into the proposed representation, we use an incremental greedy clustering algorithm creating new clusters every time new sequence, sufficiently distant from all other clusters, occurs. In previous works multiple authors used various techniques to form meaningful subsequence clusters. Most of these methods limit the number of sequences used in the clustering process by using motifs [8] or perceptually important points [9]. All of these works used the K-means algorithm in cluster formation. We hypothesize, that not by limiting the number of formed clusters, but by changing the clustering algorithm, we will be able to form meaningful clusters.

According to the authors of another study [3] many symbolic time series representations were proposed, but the distance measures on these representations show little correlation with the distance measures on original data. To show our representation is not the case, we propose the distance measure *SymD* that returns the minimum distance between time series in the representation and we show it lower bounds the Euclidean distance on the original time series. To evaluate the applicability of time series representation we use the tightness of lower bounds (TLB) [10] as it is the current consensus in the literature [11].

As the majority of existing time series representations focus on processing of static collections of data and we propose our representation to be applicable in stream data processing domain, we evaluate the properties of the proposed representation on static collections of data as well as on very long time series substituting the potentially infinite data streams.

The rest of the paper is organized as follows. Section 2 introduces the symbolic time series representation. Section 3 defines the distance measure on the proposed representation and provides the proof it lower bounds the Euclidean distance on the original data. An experimental evaluation of properties of the proposed representation and distance measure on the number of datasets is presented in section 4. We conclude by summarizing obtained results and by hints on future work.

2. The Symbolic Representation

As a base for our time series representation we use an assumption presented in [12]. The authors state that frequent patterns extracted from time series data are more stable than the time series itself. We use this assumption to form the main idea of our representation as to represent time series data as a sequence of reoccurring patterns. We search for reoccurring similar subsequences in the course of the whole data stream by clustering subsequences. We transform them into sequences of symbols where every subsequence cluster identifier is transformed into a symbol similarly to the representation proposed in [4]. For the purpose of our work, we will refer the proposed representation as to *Incremental Subsequence Clustering* (*ISC*).

The transformation of stream data to the *ISC* representation can be divided into three steps:

1. Split incoming data into overlapping subsequences using running window.
2. Cluster z-normalized subsequences by their similarity.
3. Use cluster identifiers as symbols, subsequences are transformed to. In connection with normalization coefficients, these symbols approximate the original data.

As the processed time series may contain some levels of noise and trend, the preprocessing step may be introduced into the transformation. To remove the noise present in the formed subsequences and to highlight important parts of the data, some level of smoothing can be applied before the symbol formation as the introduction of smoothing before the symbols are created can produce more stable alphabet of symbols. To find the correct level of smoothing, one could use a framework such as the one presented in [13], based on Minimum Description Length principle [14]. In the evaluation of the proposed representation presented in this paper however, we did not use any smoothing as we did not want to introduce any error by omitting minor changes in the shape of the processed time series.

The ISC representation is inspired by the representation presented in [4], with two important differences:

- we use overlapping symbols and
- we don't use K-means algorithm in symbol formation.

The redundancy contained in overlapping symbols could be used to improve the reconstruction accuracy when transforming data back to their raw form, and to some extent it is used in the similarity measure on the data transformed into ISC (presented in later parts of the paper). The main motivation to introduce the overlapping symbols however, is to support one of intended applications of the time series representation - short term time series forecasting. If time series is transformed into a symbolic representation with overlapping symbols incrementally, in every moment at least the length of the overlapping part of two symbols could be used to search for similar shapes in alphabet of symbols. The last part of the processed time series could be simply compared to early parts of symbols in the alphabet. The later part of the most similar symbol from the alphabet can be then used to forecast the rest of the symbol's length. Of course, this would be just the simplest method which could be extended by employing other similar symbols or sequence of symbols occurring earlier in the transformed time series.

The main difference of the proposed *ISC* representation to the representation Das et al. used [4] is the clustering algorithm we use for symbol formation. They used K-means, which is iterative in its nature and requires the number of formed clusters to be specified in advance. As shown in [7], this results in meaningless cluster formation as the cluster centre does not reflect the data, cluster is formed from, but transforms into a shifted sinusoidal shape regardless the shape of the transformed data. We chose different approach to symbol formation by not using K-menas clustering algorithm.

We use incremental greedy algorithm not limiting the number of clusters but limiting the maximal distance of subsequences from the cluster centre. The algorithm assigns subsequence into the cluster if its distance from the cluster centre is smaller than the predefined threshold (referenced as limit distance). The algorithm forms new cluster with the subsequence in its centre if no cluster with the distance to the processed subsequence lower than the maximal distance exists.

The pseudo-python code for the described clustering algorithm and the transformation is as follows:

```
clusters = [] # Symbol alphabet

# Transforms time series into sequence of overlapping
# symbols of defined size
```

```
def transform(series, size, overlap, limit_dist):
        symbols = []
        windows = split_windows(size, overlap, series)
        for window in windows:
                cluster = get_cluster(window, limit_dist)
                symbols.append(cluster.id)
        return symbols

# Finds or creates a cluster in the cluster alphabet.
# First cluster within limit distance from
# the subsequence is returned or new one is created
def get_cluster(window, limit_distance):
        cluster = None
        for c in clusters:
                if dist(c.centre, window) < limit_dist:
                        cluster = c
                        break

        if cluster == None:
                cluster = create()
                cluster.centre = window
                clusters.append(cluster)

        return cluster
```

As clusters are not updated and the first subsequence used to form the cluster is used as its representative, the cluster centres do not degrade into a shape not representing the data used in the transformation as seen in the Das' representation [4, 7]. By limiting the distance of subsequences within the cluster we are able to guarantee maximal distance the transformed time series can drift from its original shape, which is used to guarantee the lower bounding property of the distance measure on the ISC representation of time series (presented later in this paper).

The proposed representation forms an alphabet of symbols (clusters) which grows with the amount of data processed. We adopt the already mentioned assumption about frequent pattern stability presented by [12] and we assume the speed of growth of the alphabet of symbols will decrease with the amount of data processed. The experiments supporting this claim are presented in section 4.

The alphabet of symbols represents the main difference between the proposed *ISC* representation and SAX. The symbols formed by SAX represent equiprobable intervals of PAA coefficients [10] which in turn are results of an aggregate function (mean) performed on a sliding window of a time series. In the case of our representation, individual symbols represent repeating shapes and the alphabet of symbols represents an alphabet of all shapes occurring in the course of the time series. As these symbols represent frequent patterns occurring in course of the time series, we can see the transformation as a form of motif discovery [15] event though we are interested in all repeating patterns of specific length.

The transformation uses three parameters: symbol length (size of the running window), step between two consecutive windows (typically equal to a fraction of symbol length), maximal distance of cluster centre and a subsequence in the cluster. Every symbol in the alphabet of symbols is represented by z-normalized subsequence forming the centre of the cluster and the cluster identifier. The transformed time series is formed by a sequence of triplets: cluster identifier, mean and standard deviation of the original subsequence as illustrated on Figure 1. Using these attributes in connection with the alphabet of symbols, we are able to approximately reconstruct the original time series.

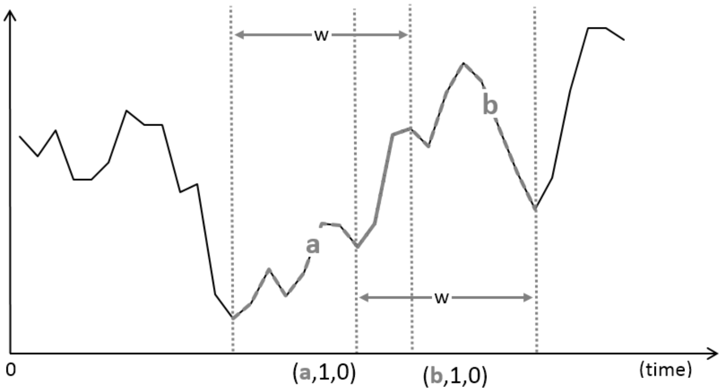

Figure 1: Sliding window (of length w) splits the time series into overlapping symbols. A sequence of symbol identifiers and normalization coefficients is used to represent the time series.

The reconstruction of the raw form of the time series from ISC representation is composed of three steps (the last step is not necessary if no overlap between symbols was used during the transformation process):

1. For every cluster identifier in transformed sequence of symbols, find associated cluster.
2. Use mean and standard deviation of the symbol to denormalize z-normalized cluster centre and use it to replace the symbol.
3. If overlap was used during the transformation and thus multiple data points (from multiple consecutive overlapping symbols) are to be positioned in place of one original data point, use their mean value instead.

As individual symbols are represented by z-normalized subsequences used as centres of clusters of similar sequences, the time series reconstruction is not exact, but small amount of error is introduced (the amount depends on the limit distance of subsequences associated to a cluster). By averaging overlapping parts of symbols some of the variability introduced by approximative representation is decreased.

As the transformation process produces ever increasing number of symbols, one could argue, that when processing unbound streams of data, the symbol alphabet could become too big to be usable. Due to the ever growing alphabet of symbols the computational complexity of the transformation is not constant as the time necessary to search for closest cluster in the alphabet of symbols grows with logarithm of its size. This is the biggest obstacle in application of the transformation on unbounded streams of data. To make the transformation applicable in thorough time restrictions of incremental stream data processing, we would need to limit the growth of the alphabet. We see the solution in the assumption that the most recent and most frequent parts of the time series are most important and should be represented with greater accuracy than the rest of the time series [16]. This leads us to the idea of alphabet symbol management using various amnesic functions [17], where old, unused symbols could be forgotten [18], merged or replaced by a supplement. If the same principle would be applied to most recent and frequent symbols, this could be used to increase the reconstruction accuracy of the representation and to reduce the size of symbol alphabet.

3. Lower Bounding Similarity Measure

Having defined the symbolic time series representation, we now define the similarity measure on the transformed data and we prove it lower bounds the Euclidean distance on the original data. As the distance measure for the *ISC* representation we adapt the representation introduced in [3] where the authors proposed an adaptation of Euclidean distance called *MINDIST*. *MINDIST* uses table of distances between individual symbols in the SAX representation of the data to calculate the overall distance. In this representation, the distance table depends solely on the number of symbols used in the transformation process. As the *ISC* representation does not use stable alphabet of symbols and the distance between symbols depends on the shape of the data they are formed from, we have to calculate the distance table from the symbol alphabet. We define the symbolic distance measure (*SymD*) as an adaptation of *MINDIST* distance measure that returns the minimum distance between time series in the *ISC* representation.

The proposed distance measure builds on the most common time series distance measure - Euclidean distance. Eq. (1) shows the formula for Euclidean distance of two time series, Q and C of the length n.

$$ED(Q,C) = \sqrt{\sum_{i=1}^{n}(q_i - c_i)^2} \tag{1}$$

We show the lower bounding property of *SymD* by introducing an auxiliary distance measure as transition from Euclidean distance to the presented *SymD* distance measure. Among these distance measures we demonstrate the lower bounding property and transitively we extend the proof to the proposed *SymD* distance measure on the *ISC* representation (Eq. (2)). The auxiliary distance measure we introduce (for the explanation sake named *OverED*) is described in the following paragraphs.

$$SymD(\hat{Q},\hat{C}) \leq OverED(\overline{\overline{Q}},\overline{\overline{C}}) \leq ED(Q,C) \tag{2}$$

In Eq. (2), Q and C refers to two compared time series in their raw representation. $\overline{\overline{Q}}$ and $\overline{\overline{C}}$ refer to time series split into overlapping subsequences of length w and shift s. $\hat{Q}$ and $\hat{C}$ refers to time series in *ISC* representation.

The distance measure *OverED* refers to the adapted Euclidean distance, where we split the time series into overlapping subsequences of equal length w and shift s between two consecutive subsequences. The distance between two subsequences is calculated using Euclidean distance.

An illustration of time series transformed to overlapping subsequences is presented on Figure 2.

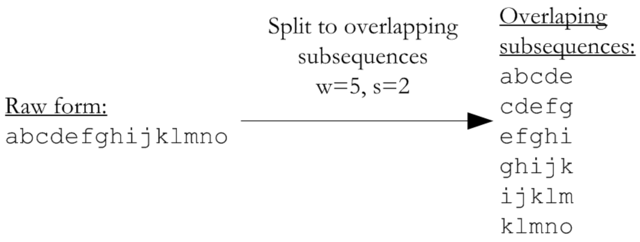

Figure 2: Example of sequence split into overlapping subsequences.

Figure 2 shows a sequence of values in a time series **abcdefghijklmno** where every symbol refers to a different value. *OverED* operates on the time series split into overlapping subsequences of length w and shift s. We choose in our example $w = 5$ and $s = 2$ and we split the sequence.

As we can see from the example, some values are represented repeatedly in the transformed data (eg. **c, d, e** ...) and some are represented only once or with different frequencies (eg. **a, b, n** and **o**). The contribution of the time series value to the overlapping representation depends on its position in the processed time series. None of these values however is repeated more than $\lceil \frac{w}{s} \rceil$ times. We define the *OverED* as sum of squared distances between subsequences (similarly to Euclidean distance) divided by the maximal number of occurrences of individual values in the transformed representation. Eq. (3) shows the definition of *OverED* where $\overline{\overline{q_i}}$ and $\overline{\overline{c_i}}$ are i-th subsequences of time series $\overline{\overline{\overline{Q}}}$ and $\overline{\overline{\overline{C}}}$, n is the total length of time series, w is the subsequence length, s is the shift between two subsequences and $\lceil \frac{n-w}{s} \rceil$ is the total number of symbols in the transformed representation.

$$OverED(\overline{\overline{\overline{Q}}}, \overline{\overline{\overline{C}}}) = \sqrt{\frac{\sum_{i=1}^{\lceil \frac{n-w}{s} \rceil} ED(\overline{\overline{q_i}}, \overline{\overline{c_i}})^2}{\lceil \frac{w}{s} \rceil}} \tag{3}$$

An alternative notation for the *OverED* distance measure is based on the number of occurrences of individual time series values in the overlapping representation. To measure the contribution of individual values to the resulting representation, we can split the time series into three parts:

- Start - with increasing contribution of values to the overlapping representation.

- Centre - with constant contribution of different values to the representation.
- End - with decreasing contribution of different values.

The distance measure on such representation has to adjust to the variable contribution of values to the representation. We can define the contribution for each part of the time series to the overall distance measure separately as:

$$Start(\overline{\overline{Q}}, \overline{\overline{C}}) = \sum_{i=1}^{\lceil \frac{w}{s} \rceil} \sum_{j=1}^{min(s,w-s(i-1))} i(q_{is+j-1} - c_{is+j-1})^2 \tag{4}$$

$$End(\overline{\overline{Q}}, \overline{\overline{C}}) = \sum_{i=1}^{\lceil \frac{w}{s} \rceil} \sum_{j=1}^{min(s,w-s(i-1))} i(q_{n-is+j} - c_{n-is+j})^2 \tag{5}$$

$$Centre(\overline{\overline{Q}}, \overline{\overline{C}}) = \lceil \frac{w}{s} \rceil \sum_{i=w+1}^{n-w-1} (q_i - c_i)^2 \tag{6}$$

In Eq. (4), Eq. (5) and Eq. (6), q_i and c_i to i-th values of time series Q and C. Since every q_i and c_i from Q and C respectively is not repeated in the representation more than $\lceil \frac{w}{s} \rceil$ times, we can divide the sum of distances of three parts of the time series by $\lceil \frac{w}{s} \rceil$ and the resulting distance will be never greater than $ED(Q, C)$ thus it satisfies the lower bounding property.

$$OverED(\overline{\overline{Q}}, \overline{\overline{C}}) = \sqrt{\frac{Start(\overline{\overline{Q}}, \overline{\overline{C}}) + Centre(\overline{\overline{Q}}, \overline{\overline{C}}) + End(\overline{\overline{Q}}, \overline{\overline{C}})}{\lceil \frac{w}{s} \rceil}} \leq ED(Q, C) \tag{7}$$

The last step of the proof is to show that clustering of similar subsequences using Euclidean distance into clusters, defined by its centre and maximal distance of the subsequence from the centre, lower bounds the *OverED* distance measure. The sole difference between *SymD* and *OverED* is, that the *SymD* does not compute the distance using the raw time series subsequences, but rather centres of cluster every subsequence is attached to. To calculate the distance of time series in *ISC* representation, we have to substitute the distance of overlapping subsequences by the distance of clusters centres. However, the substitution by these clusters introduces some error as they are only approximate representations of the original overlapping subsequence. To use the cluster centres instead of the original subsequences we have to define the relation of Euclidean distance of the individual subsequences and the Euclidean distance of cluster centres. For the purpose of this proof $\tilde{a}$ and $\tilde{b}$ refer to the cluster centres time series a and b respectively are associated to. The cluster diameter or maximal distance between cluster centre and time series associated to this cluster is denoted r. We start the proof using the equality of Euclidean distance of cluster centres to itself in Eq. (8).

$$ED(\tilde{a},\tilde{b}) = ED(\tilde{a},\tilde{b}) \tag{8}$$

Using triangular inequality (Eq. (9)) of ED twice on the right side of Eq. (8), we obtain Eq. (10)

$$ED(a,b) \leq ED(a,c) + ED(c,b) \tag{9}$$

$$ED(\tilde{a},\tilde{b}) \leq ED(a,\tilde{a}) + ED(a,b) + ED(b,\tilde{b}) \tag{10}$$

As $ED(a,\tilde{a}) \leq r$ and $ED(b,\tilde{b}) \leq r$ we can transform the Eq. (10) to:

$$ED(\tilde{a},\tilde{b}) - 2r \leq ED(a,b) \tag{11}$$

The geometrical illustration of this proof is on Figure 3.

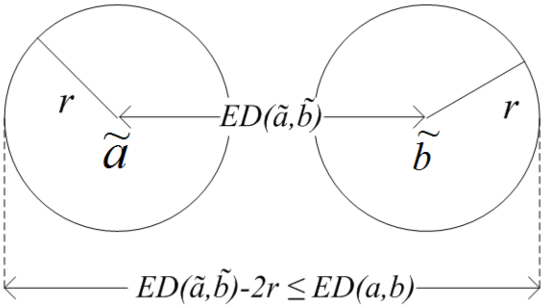

Figure 3: Geometrical illustration of the relation between distance and distance of cluster centres.

By applying the Eq. (11) on *OverED* distance measure from Eq. (3), we show that:

$$\sqrt{\frac{\sum_{i=1}^{\lceil\frac{n-w}{s}\rceil} ED(\hat{q}_i,\hat{c}_i)^2}{\lceil\frac{w}{s}\rceil}} - 2r\lceil\frac{n-w}{s}\rceil \leq \sqrt{\frac{\sum_{i=1}^{\lceil\frac{n-w}{s}\rceil} ED(\hat{q}_i,\hat{c}_i)^2}{\lceil\frac{w}{s}\rceil}} \tag{12}$$

And thus:

$$SymD(\hat{Q},\hat{C}) = \sqrt{\frac{\sum_{i=1}^{\lceil\frac{n-w}{s}\rceil} ED(\hat{q}_i,\hat{c}_i)^2}{\lceil\frac{w}{s}\rceil} - 2r\lceil\frac{n-w}{s}\rceil} \leq OverED(\overline{\overline{Q}},\overline{\overline{C}}) \tag{13}$$

where n is the total number of values in the time series, $\hat{q}_i$ and $\hat{c}_i$ refers to i-th symbol time series $\hat{Q}$ and $\hat{C}$ in *ISC* representation, r is the radius of the clusters forming the symbols, w is the length of the symbol and s is the shift between two symbols. Using the Eq. (13), we prove *SymD* lower bounds *OverED* and thus we complete the proof of Eq. (2). We show that the proposed distance measure *SymD* operating on time series transformed into *ISC* representation lower bounds the Euclidean distance on raw form of the time series.

As seen from Eq. (13), sum of distances of symbols is divided by the maximal number, a single time series value can be applied in formation of multiple symbols due to symbol overlapping ($\lceil\frac{w}{s}\rceil$). This is equivalent to averaging of overlapping values introduced from consecutive symbols. As every symbol is only approximative representation of the original data (one time series subsequence is used as representative for a whole cluster of similar time series subsequences), by averaging overlapping values, the similarity measure reduces the impact of possible outlier values on the resulting distance estimation and thus increases the measure's noise reduction capacity. Similar approach can be used to reduce noise when reconstructing the transformed time series into its raw form.

4. Evaluation

We use two different types of datasets to evaluate properties of the proposed representation. We use the well known UCR datasets collection [19] to evaluate the tightness of lower bound of the *ISC* representation as one of the most widely used metrics for evaluation of time series representations [11]. We use the UCR datasets also for evaluation of stability of symbol alphabet formed during the transformation and the size of alphabet as it determines the memory requirements of the representation and its applicability in stream data processing. We use these datasets also for evaluation of applicability of the proposed representation on time series classification.

As UCR datasets are composed of rather short time series, we use an electricity consumption dataset [20] from Belgian electricity transmission operator to evaluate the properties of the representation when processing very long time series data. We use this dataset to compare the dimensionality reduction capacity of the *ISC* representation while preserving the reconstruction accuracy.

4.1. Representation properties on short time series

Since the transformation into the *ISC* representation requires three parameters to be set, in the following figures we provide several examples of the relationship between these attributes, tightness of lower bound and symbol alphabet size. Figure 4, Figure 5 and Figure 6 display the data obtained by processing the Symbols dataset from the UCR [19] repository. Similar results were obtained for other datasets from the repository, but they are omitted due the limited length of this paper.

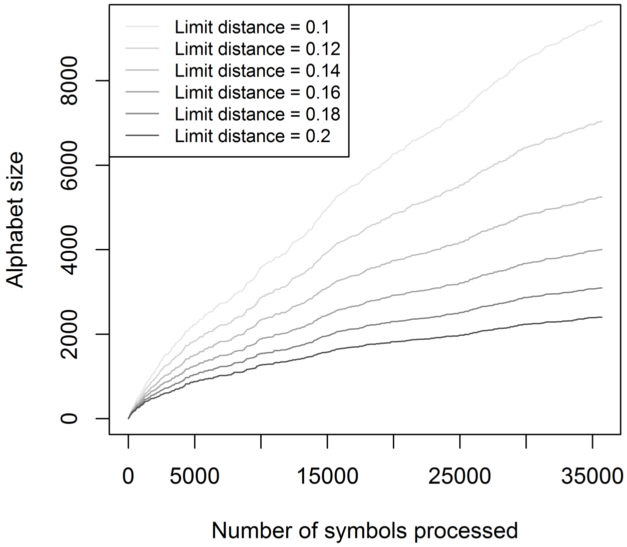

Figure 4: The relationship between alphabet size and number of data processed with different settings of maximal distances of subsequence to the centre of associated cluster. Data for UCR [19] dataset Symbols.

Figure 4 shows the relationship between the amount of data processed and the size of the symbol alphabet. The figure displays the evolution of alphabet size with increasing portion of the dataset processed and for different settings of the limit distance used in cluster formation. We can see that the speed of formation of new symbols decreases with the amount of processed data in accordance with our assumption about stability of frequent patterns introduced in the section 2. Similar results are visible also when processing very long time series such as electricity consumption data from the following section (Figure 14). The differences in the total alphabet size for distinct limit distance settings (Figure 4) indicate the increasing number of clusters formed when size of the cluster is small. The relation between the size of alphabet formed after transformation of the whole dataset and the size of cluster created during the transformation is displayed on Figure 5. One can see that the relation is not linear, but with the increasing size of the clusters the decrease in the total number of symbols slows down.

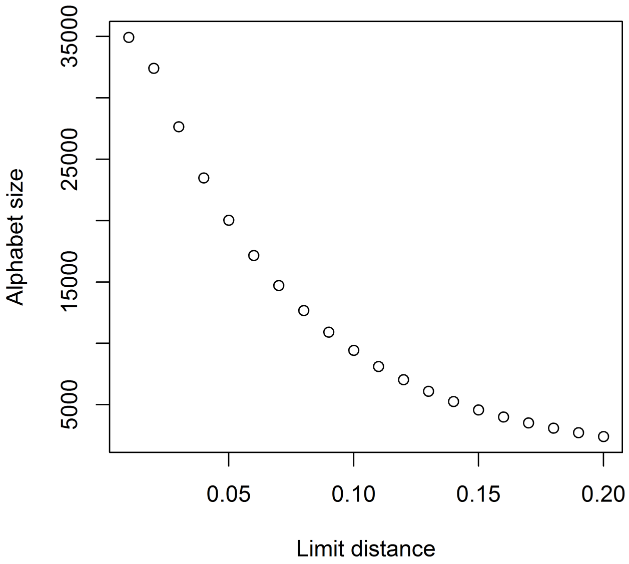

Figure 5: The relationship between the final alphabet and size of created clusters. Data for UCR [19] dataset Symbols.

With the increasing size of the clusters, more similar subsequences are associated with the same cluster centre. This should result in decreased accuracy of reconstruction of the representation to the original time series data. The accuracy of reconstruction is reflected in the tightness of lower bound metric as it indicates the ratio between the similarity of two transformed time series calculated using the *SymD* distance measure and the distance calculated using Euclidean distance on the original time series. The relation between tightness of lower bound and cluster size is presented on Figure 6.

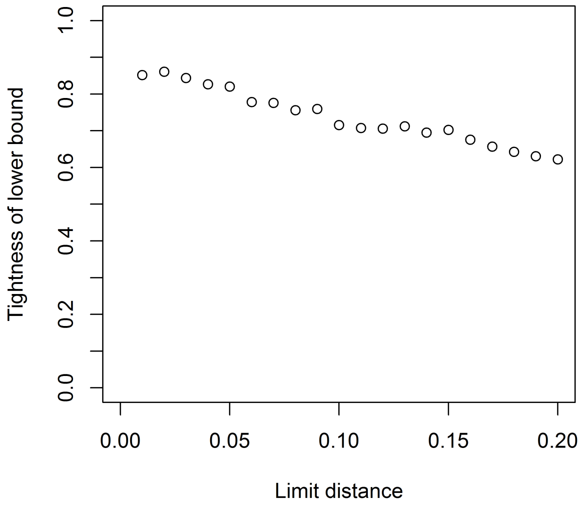

Figure 6: The relationship between the tightness of lower bound and size of created clusters. Data for UCR [19] datasets Symbols.

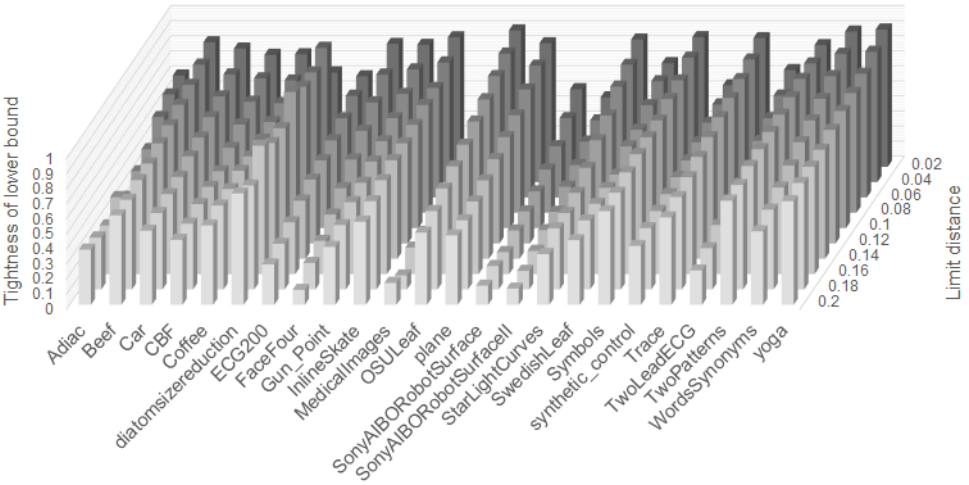

Figure 7: Tightness of lower bound for different datasets from the UCR repository [19] and different sizes of formed clusters.

To evaluate the tightness of lower bound we performed an experiment where we took a sample of 200 time series from the Symbols dataset and we calculated the average tightness of lower bound for every pair of these time series. We performed the experiment for different sizes of formed clusters. The results are presented in Figure 6. The relationship between the tightness of lower bound and cluster size is almost linear with small variability caused by the size of the used sample. These results indicate there is a trade-off between the size of the created symbol alphabet and the tightness of lower bound obtained by the *ISC* representation and associated *SymD* distance measure. When one will choose the settings for the transformation he/she has to decide on the basis of the application at hand.

The relation between the tightness of lower bound and limit distance used in cluster formation for other datasets from the UCR repository [19] is displayed on Figure 7. The graph shows the TLB increases with the decreasing size of the clusters for every used dataset. The value of the maximal obtained tightness for used settings, however, is variable between datasets. For some datasets the limit distance has to be smaller to obtain the same TLB.

To compare the proposed representation to other time series representations such as SAX, PAA or DFT, we can use the results presented in [11]. This comparison however, provides only limited informative value as these representations use different parameters and majority of them is iterative in their nature in contrast to the proposed representation. The authors of this study evaluated various time series representations with different transformation settings on EEG dataset from the UCR repository [19]. The obtained tightness of lower bound varied from 0.258 to 0.782. The results for *ISC* representation in combination with *SymD* distance measure varied from 0.268 to 0.601 with different settings of the transformation. The proposed representation thus obtained comparable results with possible improvements if smaller limit distance was used in the transformation process.

To evaluate the clustering meaningfulness we had to adapt the formula used in [7]. The clustering meaningfulness is a measure defined on two distinct datasets as a fraction of mean minimal cluster centre distances within dataset, over mean minimal cluster centre distances between datasets [7]:

$$meaningfulness(\hat{X},\hat{Y}) = \frac{within\_set\_\hat{X}\_distance}{between\_set\_\hat{X}\_and\_\hat{Y}\_distance} \tag{14}$$

The original definition of $within\_set\_\hat{X}\_distance$ presented in [7] calculates the mean minimal distance of cluster centres formed by multiple runs of K-means algorithm on the dataset. Since our clustering algorithm does not use random initialization, the minimal distance of clusters formed by multiple executions of the algorithm would be zero. We simplify the meaningfulness formula to be equal to the mean minimal distance between sets.

To evaluate the meaningfulness of subsequence clusters formed during the transformation of time series into the *ISC* representation we performed an experiment on several datasets from the UCR repository [19]. We clustered pairs of datasets and compared mean distance of formed clusters for different settings of cluster formation. We used whole time series to form the clusters and fractions of the time series as symbols in the *ISC* representation. As the lengths of the formed symbols we used 1/2, 1/4 and 1/8 of the sequence length. As for other transformation settings, the step between symbols was set for one half of the symbol size (not in the case of whole clustering, where the step was not used) and the limit distance between cluster centre and associated subsequences was set to 0.2. The results for several pairs of the datasets are displayed on Figure 8.

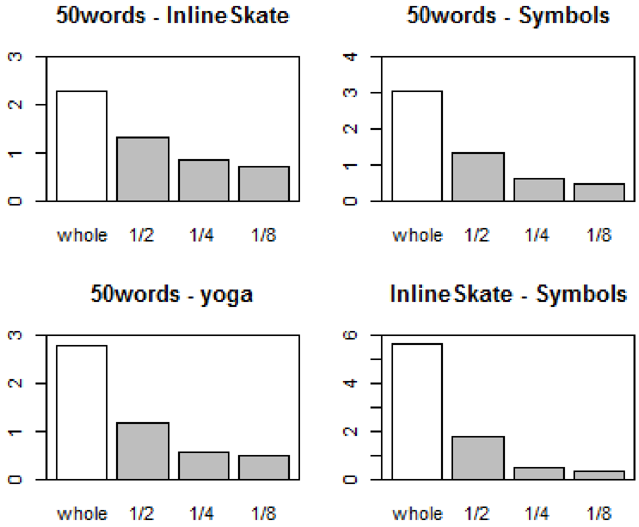

Figure 8: The meaningfulness evaluation for multiple dataset combinations and different settings of symbol lengths used for the transformation. Diagrams show mean shortest distance between clusters of two datasets when whole sequences were clustered and when *ISC* transformation was used with symbol sizes of 1/2, 1/4 and 1/8 of time series length.

One can see the mean distance between datasets decreases when the size of the symbol is decreasing for every examined combination of datasets. The change in distance approximately follows the size of the time series fraction used as symbol. This is caused by the space of similar sequences filling up when the length of clustered subsequences is decreasing and when the radius of clusters is fixed. This results in more formed clusters, closer together. When we shrink the size of symbols even more, the normalized symbols are reduced into a small alphabet of basic shapes as seen on Figure 9. The decrease in mean minimal cluster centre distance is not caused by the randomness of formed clusters, but by the shrinking subsequence space as the centres are formed from the original time series shapes.

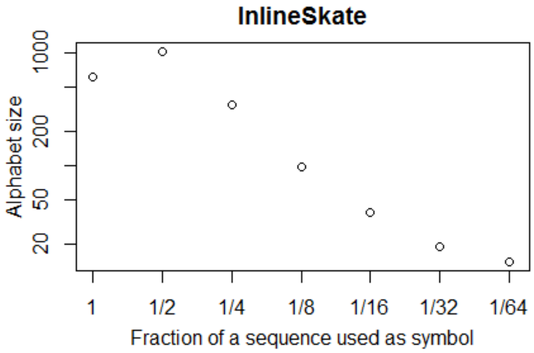

Figure 9: The alphabet size when different symbol length are used. Logarithmic scale used on both axes.

The most often used approach to evaluate various similarity measures and data representations is classification. We performed an experiment on the proposed representation using experimental setup described in [11] to evaluate applicability of the proposed ISC representation and SymD distance measure on the task of time series classification. Authors in [11] used 1-NN classifier and multiple similarity measures on UCR collection of datasets to compare their properties on various types of datasets. The results showed big difference between various datasets when comparing similarity measures on ISC transformed dataset to Euclidean distance on raw form of the time series. The proposed ISC representation in connection with the SymD distance measure showed promising results, producing smaller error ratio than Euclidean distance on most of the datasets. The Figure 10 shows results comparing the SymD distance on ISC representation and Euclidean distance on raw form of data.

The Figure 10 displays the error rates of both methods when classifying multiple datasets from the collection. The data point is shifted from the

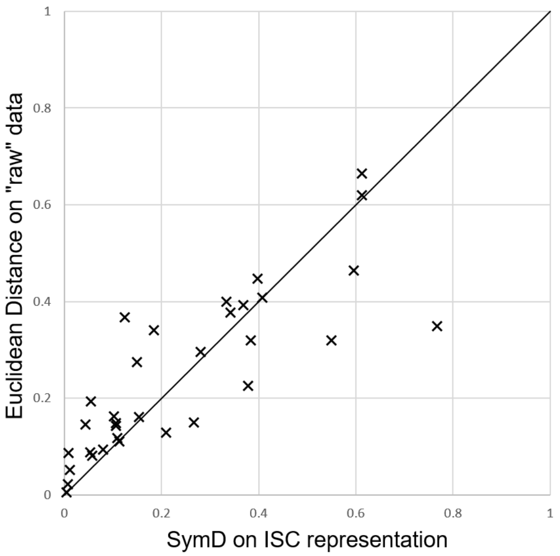

Figure 10: Figure displays the classification error on various datasets of the UCR collection. Each data point represents classification error rates of both compared methods. The diagonal line represents equivalence of compared methods. The less points is on one method's side, the better it performed.

diagonal line to the method's side, which produced higher error rate. The less points is displayed on method's side, the better it performed. As seen from the figure, the SymD distance using ISC transformed data outperformed the Euclidean distance on most datasets. In total, the combination of SymD distance measure and ISC transformation obtained smaller classification error on 24 from 33 processed datasets. The improvement ratio however greatly varied as the representation is more suitable for some datasets and produce rather high error rates on another. In general, proposed symbolic time series representation and associated similarity measure provided promising results.

4.2. Evaluation on long time series

Using the UCR dataset, we evaluated the properties of the *ISC* representation on a variety of time series data with diverse characteristics. In the next step, we will focus on very long time series where strong seasonality is present, possibly with multiple levels of seasonality (daily, weekly, monthly, ...), while multiple repeating patterns can be present in the data. Various production/consumption data are example of such datasets, where the measured value greatly depends on the time of the day and the day of the week.

This type of very long time series routinely contains various types of concept drifts and with many repeating patterns it poses great challenges for tasks such as prediction [21] or anomaly detection. We used the electricity consumption data published by Belgian electricity transmission systems operator [20]. We used the data from years 2005 to 2015, representing real time grid load sampled in 15 minutes intervals. In total, the data was composed of 374 496 data points. An example of one week portion of the data is displayed on Figure 11. Strong seasonal pattern is present in the data. The days of the workweek greatly differs from the days of the weekend and even patterns present in different days varies.

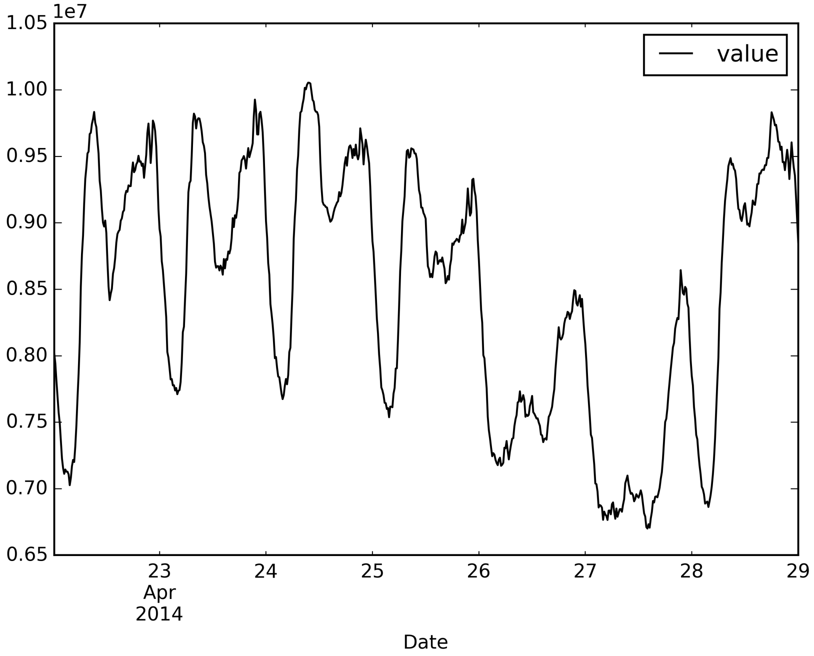

Figure 11: An example of one week portion of Belgian electricity consumption data.

In evaluation of the *ISC* representation properties on very long, seasonal time series, we focus on its dimensionality reduction ability when whole seasonal patterns are represented by symbols and on comparison of the reconstruction error and dimensionality reduction ability with the most often used time series representation - *Piecewise Aggregate Approximation* (*PAA*).

When transforming time series data into the ISC representation, the transformed data size is composed of two parts: the sequence of symbol identifiers and the symbol alphabet. We hypothesize, that when the symbol length is set equal to the length of the seasonal pattern, similar patterns can be replaced by symbols from the alphabet of shapes and the size of the alphabet necessary to represent the whole dataset will be much smaller than if other symbol sizes were used in data transformation process. To evaluate this hypothesis, we performed an experiment, where we transformed the electricity consumption dataset into ISC representation using various symbol lengths. No overlap was used in this experiment and limit distance was set to 3.0. Results of this experiment are displayed on Figure 12.

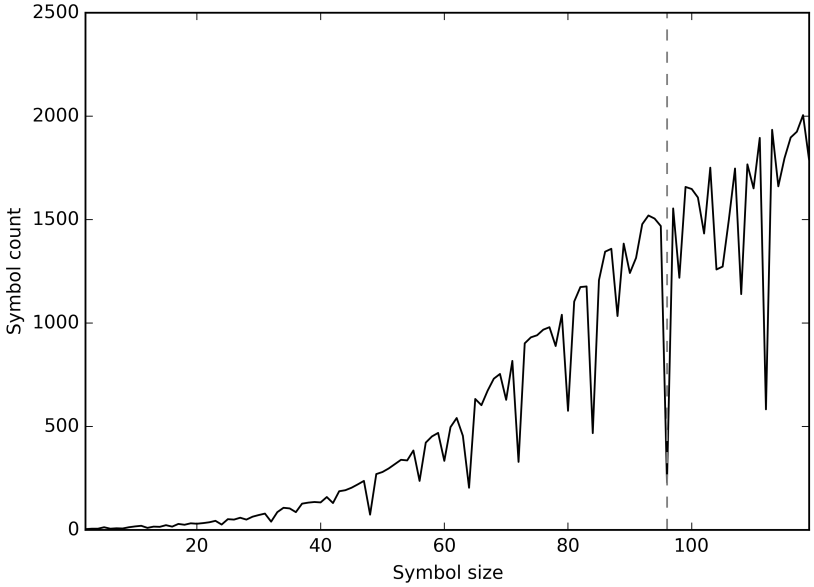

Figure 12: Size of symbol alphabet necessary to represent the long seasonal time series when various symbol lengths are used. The grey vertical dashed line indicates the actual size of seasonal pattern.

The Figure 12 shows very small number of symbols necessary to represent the whole time series when small symbols are used. This is consistent with the findings from the previous section as short symbols represent few basic shapes of the time series and the space of possible shapes is rather small. As the size of symbols grows, the number of symbols also grows. For some symbol lengths, we can see a sudden (very narrow) drop in the size of the alphabet necessary to represent the data. These symbol lengths indicate some kind of repeating pattern present in the data. The biggest drop showed on the figure is present on the symbol size equal to the number of data points necessary to represent one day worth of data. Other drops in the number of symbols in the alphabet are present on places, where symbol size represent fractions or multiples of the most important pattern lengths present in the data. This figure illustrates the dimensionality reduction ability enabled by replacing repeating shapes by symbols and at the same time it shows the necessity to correctly choose the size of symbols used when transforming the dataset as by missing by a single point can cause big difference especially for data with very strong seasonal patterns.

The next step in evaluation of the *ISC* representation on long time series data is comparison of its reconstruction error and dimensionality reduction ability with (*PAA*). We chose the PAA as the most frequently used time series representation (other than the raw form of the data). Since the two compared representations require different parameters to be set, we set these parameters empirically, in a way to obtain approximately the same reconstruction error for both representations. As the reconstruction error metric, we use *Root Mean Square Error* (*RMSE*) calculated between the original time series and the transformed time series reconstructed back to its original form. In the experiment we set the *PAA* coefficient to be equal to 7 (seven consecutive values will be averaged). The symbol size of *ISC* representation was set equal to the number of points in one day of data - 96. The step between two symbols is also equal to 96, meaning the symbols are not overlapping and no data point is skipped. We selected the limit distance parameter of the transformation into the ISC representation in a way to achieve approximately the same reconstruction error when reconstructing the ISC transformed data into its original form and when transforming data from the PAA representation. Experimentally, we set the limit distance of a cluster to 2.4. Normalization coefficients for every symbol created using ISC were calculated from previous one week worth of data to eliminate seasonal effects on coarser granularity (monthly and yearly seasonality). The *RMSE* metric for both compared representations computed repeatedly for increasing portions of the transformed time series (one week increments were used) is presented on Figure 13.

By fixing the reconstruction error for both compared time series representations, we were ale to evaluate the dimensionality reduction ability of both representations in equivalent conditions. On the Figure 14, we present the evolution of the transformed data size for ever increasing portions of the dataset transformed.

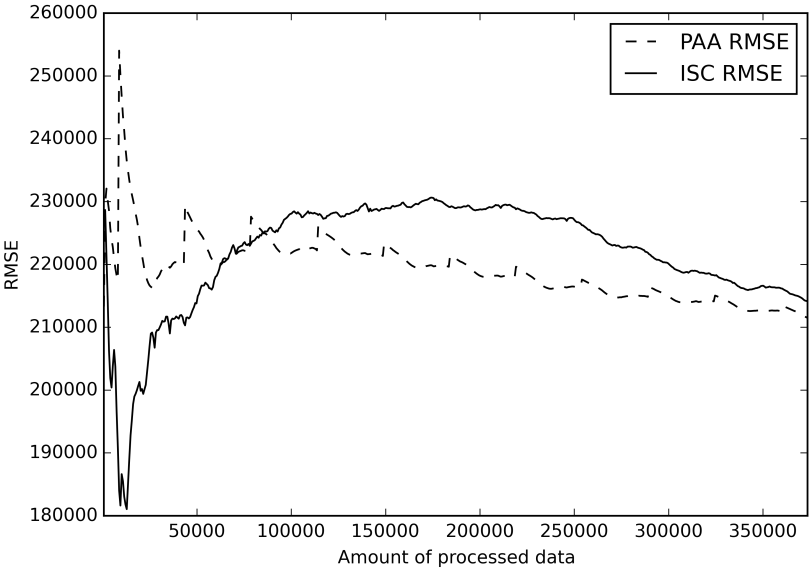

Figure 13: The evolution of reconstruction error for different portions of the data transformed.

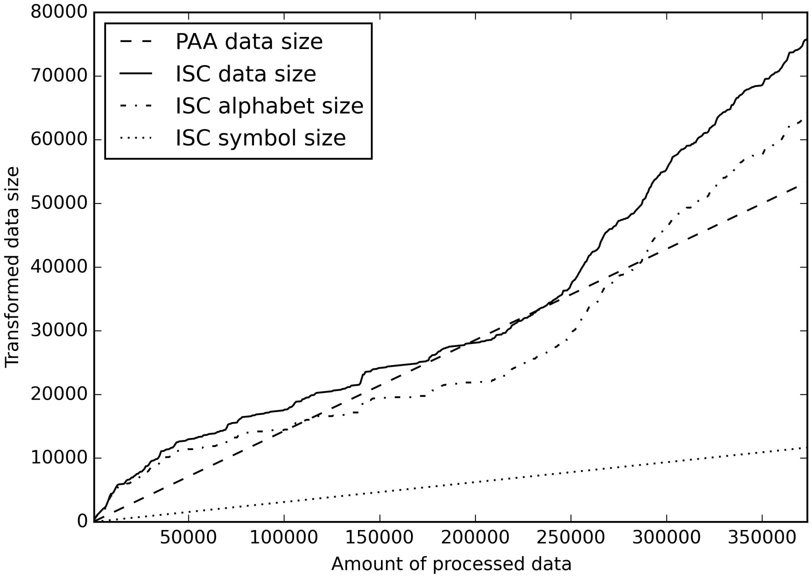

Figure 14: The comparison of the size of transformed data into PAA and ISC representations.

As expected, the size of the data transformed into *PAA* changes linearly with the amount of processed data (dashed line). The size of the *ISC* representation, however, is composed of two parts: the alphabet size (dash-dot line) and the size of the sequence of symbol representatives (dotted line). The sum of these components is displayed as full line. The size of the alphabet grows much faster than the size of *PAA* transformed data at the beginning, but it slows down as the transformation continues. We already saw this in the previous section, when we transformed the data from the UCR datasets. As a result, when half the dataset was processed, the overall size of ISC transformed data and PAA transformed data aligned and ISC produced even slightly smaller data representation. If the shape of the ISC transformed data would continue in the same manner as until this point, the ISC would produce smaller representation and we could say the ISC representation produces smaller time series representation on very long time series compared to PAA. However, from this point on, the symbol alphabet produced by ISC resumed in its rapid growth. This could be in conflict with our initial assumption about frequent pattern stability in comparison to the stability of the whole time series. We see two possible causes for this effect:

1. The older symbols are slowly worsening in accurate representation of the original data as the data drifts.
2. The patterns in the course of the time series change suddenly and thus new symbols have to be formed.

We believe both of these cases are present, but the second one is much more powerful in this dataset. To show the effect of the changing patterns in the number of formed symbols, we analyse the number of frequently occurring symbols (Figure 15) and the number of first occurrences of symbols in the course of the time series (Figure 16).

The Figure 15 displays the distribution of symbols by their frequency. On the left side are the most frequent symbols and on the right side the rarest ones. We divided the symbols into three groups:

- Frequent symbols covering 50% of all transformed data.
- Rare symbols occurring only once.
- Common symbols representing the rest of the dataset.

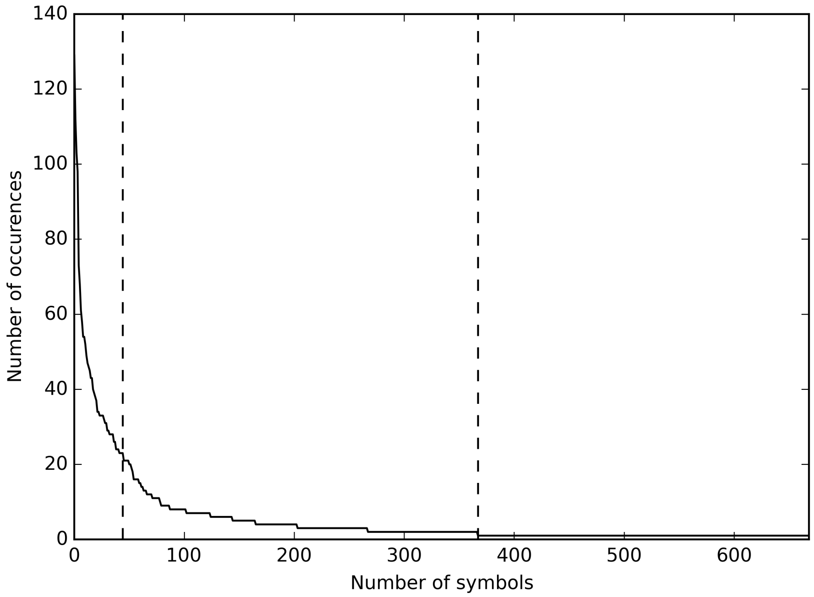

Figure 15: The distribution of symbols by their frequency.

One can see the symbol distribution follows the power law: very few symbols covers most of the dataset and almost half of the symbols occur only once.

In the Figure 15 these groups are separated by vertical dashed lines. To evaluate the number of symbols created in the course of the electrical energy consumption data, we split the time series into sequences of fixed size (one half of a year) and for every sequence we counted the number symbols occurring for the first time in the course of the entire dataset. The Figure 16 displays counts of first occurrences of symbols from different groups and sum of all first occurrences.

As we can see, the biggest number of new symbols, from all groups, is formed in the opening part of the time series and continues much slower throughout the course of the time series. After the first half of the dataset is processed, a sudden increase in the number of formed symbols appears. This supports the previous observations and suggests some sudden change in the data in the second half of the dataset.

To explain the sudden increase in the number of formed symbols, we performed another experiment. We hypothesize that if this sudden increase in the number of formed symbols was caused by the deterioration of the

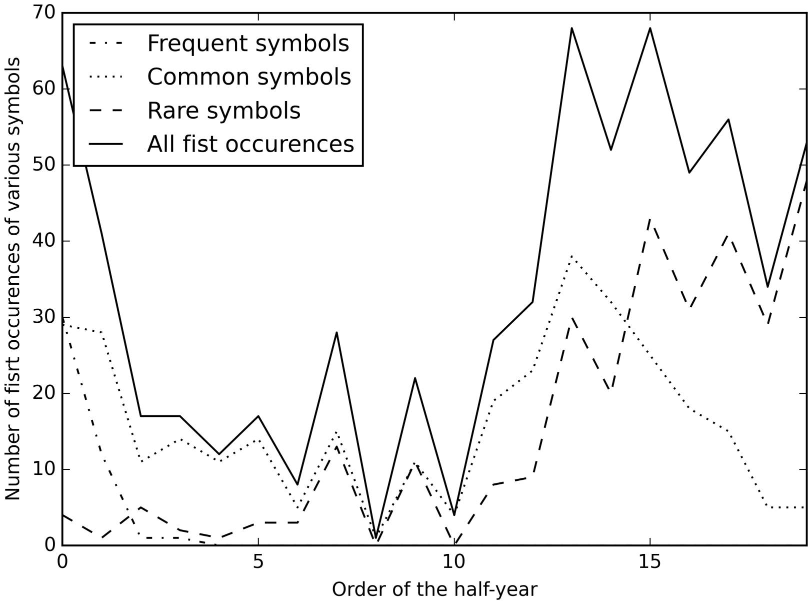

Figure 16: The number of first occurrences of symbols per fixed period of time.

alphabet, we would need much smaller number of symbols to represent the time series if we would transform only the second part of the time series.

We calculated the number of unique symbols present in fixed size windows of the time series when the whole time series was transformed into *ISC* representation and when only the second half of the dataset was transformed. The results are displayed on Figure 17 and Figure 18 respectively. In accordance with previous results, the Figure 17 shows increase in the number of different symbols used in the fixed time span near the end of the processed dataset. This suggests, that the data become more variable. The Figure 18 however, show almost the same number of used unique symbols as the second half of the Figure 17. The number of used symbols is much more stable as in the case of the whole dataset transformed, but the number of symbols used in half-year periods is the same as in the case of the whole dataset transformed. This means, that the symbols from early parts of the dataset are no longer used in the transformation and thus in later parts of the dataset are present completely different patterns as in the early parts. This suggests, that the cause of the sudden increase of the number of formed symbols is the change in the data itself and not the degradation of the ability of older symbols to represent the data.

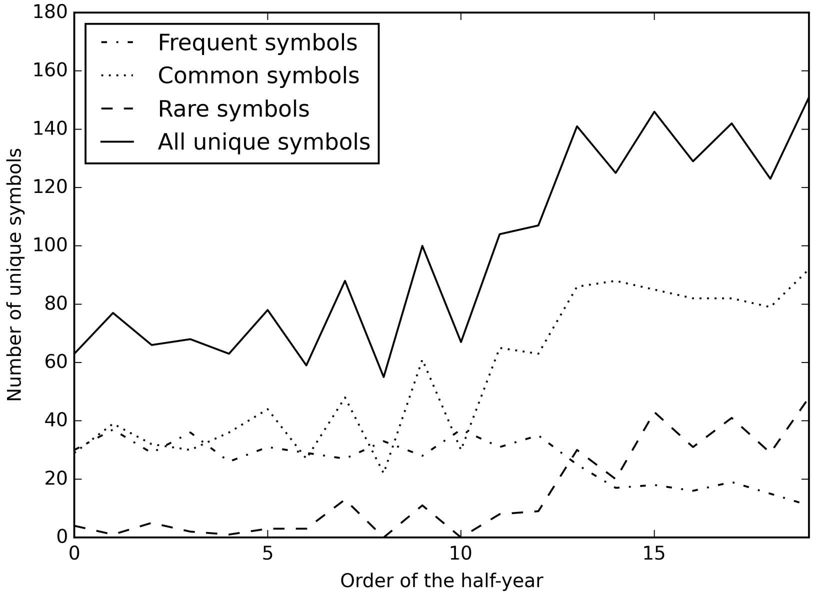

Figure 17: The average number of symbols used in a fixed period of time.

This poses a limitation of the proposed transformation as the alphabet is growing even though old symbols are no longer used. We see two opportunities in decreasing the size of created alphabet:

- To remove rare symbols occurring only once or very few times in the course of the whole dataset. This would result in dramatic reduction of the alphabet size as they represent almost half of all symbols in the alphabet.

- If we assume, we are not equally interested in all the data as authors in multiple previous works did [22, 17], we could remove old, no longer used symbols, which would provide us with another opportunity for alphabet size reduction and even with a possibility to preserve constant alphabet size.

These alphabet management approaches could help in alphabet size reduction for the prize of increased reconstruction error for some parts of the dataset. Depending on the application, this sacrifice may be acceptable. However, we leave the symbol alphabet management for the future work as it exceeds the scope of this paper.

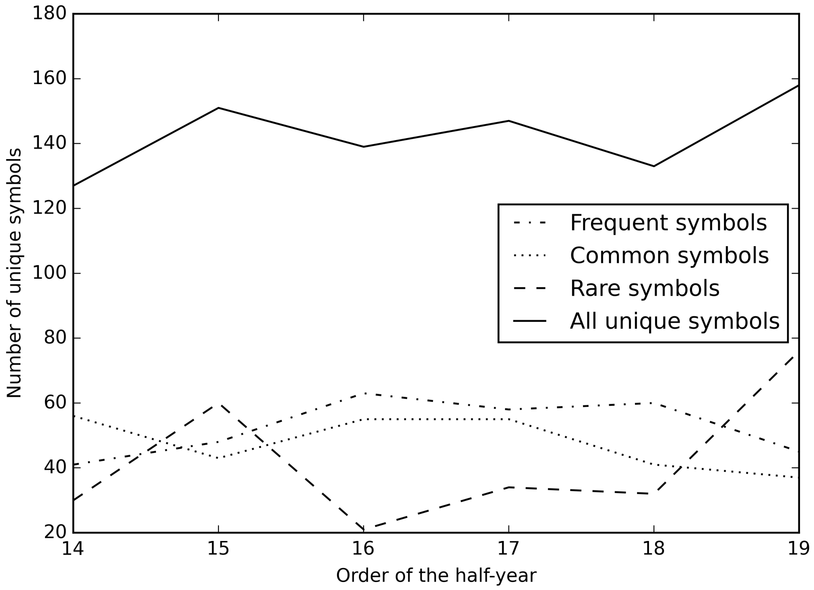

Figure 18: The average number of symbols for the second half of the dataset.

5. Conclusions and Future Work

We proposed a symbolic representation of time series (*ISC*) using clusters of similar subsequences as symbols. The clusters are formed using incremental, greedy algorithm which differs the representation from the representation used in [4] and makes it applicable on stream data processing. The major difference of the proposed representation to the *SAX* representation is the meaning of individual symbols as they represent repeating shapes in the course of the time series.

The similarity metric on the proposed representation (*SymD*) is introduced along with the proof that it lower bounds the Euclidean distance. Experiments on datasets from the UCR collection[19] show that the clustering algorithm we used in symbol formation decreases the mean minimal cluster centre distance but it is caused by the shrinking space and not the randomness or meaninglessness of formed sequences as they are formed from the basic shapes of the original time series. The evaluation of tightness of lower bound of the proposed representation and similarity metric combination showed that it is comparable with other time series representations. The potential user has to make a trade-off between the accuracy of the representation and the size of the alphabet of symbols created during the transformation by choosing the settings for size of formed clusters.

The representation allows dimensionality reduction while preserving the reconstruction error comparable to *PAA*. As the growth of symbol alphabet size slows down with the amount of processed data, the improvement to representations such as *PAA* widens when very long time series are processed. One of the limitations of the *ISC* representation though, is the ever growing database of symbols when processing very long time series. This would require management of old, unused symbols. Forgetting of unused symbols, merging of infrequent and splitting of frequent symbols could lead to manageable size of symbol alphabet when processing infinite streams of data, smaller size of transformed data and smaller reconstruction error. We leave this however as a possible extension of the representation for the future work.

Another obstacle in application of the *ISC* representation are three parameters required to be set before the transformation process starts: symbol length, between symbol step and cluster radius. However, two of those attributes can be learned from the data or application at hand as symbol length depends solely on periodicity of processed data and the between symbol step depends on the intended application. This leaves only the cluster radius to be determined experimentally depending on the required reconstruction accuracy and required level of dimensionality reduction.

The representation is applicable in domains where symbols of stable length are repeating over time and where we process large amounts of data. These are for example various domains where counting metrics on production or consumption data streams are evaluated. We use the representation for short term prediction of electricity consumption, anomaly detection and application monitoring. We see applications of the proposed representation in monitoring applications for example in the domain of network attack detection, where great number of various metrics is running continuously on diverse attributes of the network. In the future work, we will focus on management of ever growing alphabet of symbols during data stream processing, on processing of multiple parallel time series and on comparison of properties of the proposed representation with frequently used methods in tasks such as classification or forecasting.

Acknowledgment

This work was partially supported by grants VG 1/0646/15, APVV-15-0508 and was created with the support of the Research and Development Operational Programme for the project International centre of excellence for research of intelligent and secure information communication technologies and systems, ITMS 26240120039, co-funded by the ERDF.

References

[1] J. Sevcech, M. Bielikova, Symbolic time series representation for stream data processing, 1st IEEE International Workshop on Real Time Data Stream Analytics (2015).

[2] P. Esling, C. Agon, Time-series data mining, ACM Computing Surveys (CSUR) 45 (2012) 12.

[3] J. Lin, E. Keogh, L. Wei, S. Lonardi, Experiencing SAX: a novel symbolic representation of time series, Data Mining and Knowledge Discovery 15 (2007) 107–144.

[4] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, P. Smyth, Rule discovery from time series., KDD 98 (1998) 16–22.

[5] M. G. Baydogan, G. Runger, Learning a symbolic representation for multivariate time series classification, Data Mining and Knowledge Discovery 29 (2014) 400–422.

[6] A. Bagnall, E. Keogh, S. Lonardi, G. Janacek, et al., A bit level representation for time series data mining with shape based similarity, Data Mining and Knowledge Discovery 13 (2006) 11–40.

[7] E. Keogh, J. Lin, Clustering of time-series subsequences is meaningless: implications for previous and future research, Knowledge and Information Systems 8 (2004) 154–177.

[8] J. R. Chen, Useful clustering outcomes from meaningful time series clustering (2007) 101–109.

[9] T.-c. Fu, F.-l. Chung, R. Luk, C.-m. Ng, Preventing meaningless stock time series pattern discovery by changing perceptually important point detection, Fuzzy Systems and Knowledge Discovery (2005) 1171–1174.

[10] E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrotra, Dimensionality reduction for fast similarity search in large time series databases, Knowledge and information Systems 3 (2001) 263–286.

[11] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, E. Keogh, Experimental comparison of representation methods and distance measures for time series data, Data Mining and Knowledge Discovery 26 (2013) 275–309.

[12] C. Giannella, J. Han, J. Pei, X. Yan, P. S. Yu, Mining frequent patterns in data streams at multiple time granularities, Next generation data mining 212 (2003) 191–212.

[13] S. Miao, U. Vespier, R. Cachucho, M. Meeng, A. Knobbe, Predefined pattern detection in large time series, Information Sciences (2015).

[14] P. D. Grünwald, The minimum description length principle, MIT press, 2007.

[15] B. Chiu, E. Keogh, S. Lonardi, Probabilistic discovery of time series motifs, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (2003) 493–498.

[16] Y. Chen, G. Dong, J. Han, B. W. Wah, J. Wang, Multi-dimensional regression analysis of time-series data streams, in: Proceedings of the 28th international conference on Very Large Data Bases, VLDB Endowment, pp. 323–334.

[17] T. Palpanas, M. Vlachos, E. Keogh, D. Gunopulos, W. Truppel, Online amnesic approximation of streaming time series, Data Engineering, 2004. Proceedings. 20th International Conference on (2004) 339–349.

[18] C. Niederee, N. Kanhabua, F. Gallo, R. H. Logie, Forgetful digital memory: Towards brain-inspired long-term data and information management, ACM SIGMOD Record 44 (2015) 41–46.

[19] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, The ucr time series classification archive, http://www.cs.ucr.edu/~eamonn/time_series_data, 2015.

[20] Elia - grid data download, `http://www.elia.be/en/grid-data/data-download`, 2015. Accessed: 2015-09-07.

[21] G. Koskova, V. Rozinajova, A. Bou Ezzeddine, M. Lucka, P. Lacko, M. Loderer, P. Vrablecova, P. Laurinec, Application of biologically inspired methods to improve adaptive ensemble learning, 7th World Congress on Nature and Biologically Inspired Computing (2015). (Accepted).

[22] J. H. Chang, W. S. Lee, Finding recent frequent itemsets adaptively over online data streams, Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (2003) 487–492.