Towards Symbolic Representation of Potentially Infinite Time Series

Jakub Ševcech[*]
Institute of Informatics, Information Systems and Software Engineering
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
jakub.sevcech@stuba.sk

Abstract

With ever increasing amount of data produced by various sensors and applications, we are dealing with problems with its storage and processing. Great amount of this data is produced in form of streams and due to its great volume, pace and variability or simply due to methods of its analysis, we have to process it on-line, in time of its creation. In our work, we focus on processing of time series data, which we regard as potentially infinite streams of data. We discuss various representations of time series data used for dimensionality reduction and as means to support various methods for further processing. The main aim of our work is to explore possibility of incremental processing of potentially infinite time series data as sequences of symbols. We propose a time series representation using repeating shapes in the course of time series as symbols and we propose a similarity measure operating on this representation. As we focus on incremental processing of potentially infinite time series, we pay special attention to applicability of the representation under limitations of stream data processing. We discuss applications of our representation in various data analysis tasks such as classification, indexing or forecasting.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Time series analysis; H.2.8 [**Database Applications**]: Data mining; G.1.2 [**Numerical Analysis**]: Approximation; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

Keywords

Time series, symbolic representation, evolving data streams, repeating shapes, clustering, alphabet size reduction

1. Introduction

When we think about the BigData, most often, we think mainly about the volume (as the name suggests) and we neglect the other factors making the data challenging to process. Usually, we forget that the data is hard to process not only because of the sheer amount of the data but also velocity of its creation, veracity and other Vs of the big data [7].

In our work, we focus mainly on volume and velocity as we focus on processing of very long and potentially infinite time series data incoming in form of a data streams. We thus separated the problem domain into two separate parts: methods for time series data analysis and stream data processing.

1.1 Time Series Data Processing

Time series data analysis comprises of many analysis tasks, we can stumble upon when processing other kinds of data such as classification, anomaly detection or clustering. The main difference of methods to solve these tasks on time series data from other kinds of data is the fact, that we are working in multidimensional space of sequences of real valued data. Time series have many dimensions as they are formed by rather long sequences of values, but their intrinsic dimensionality is much lower. To highlight this structure, many time series representations were proposed over time. Some of them focus on highlighting the most important frequencies present in the data (such as Fast Fourier Transform - FFT or Singular Vector Decomposition - SVD), they select similar subsequences called motifs [13] or use various approximations to reduce data dimensionality and transform them into discrete (using for example Piecewise Aggregate Approximation - PAA) or symbolic sequences (using representations such as Symbolic ApproXimation [11]).

An interesting group of time series representations are those, transforming the time series into sequences of symbols. They are interesting as they allow application of methods not directly applicable on continuous data such as Markov models or prefix trees, but also a plethora of methods from the domain of information retrieval such as Inverted index, if they are used to represent time series in a vector space model. In our work, we pay special attention to these methods as they allow application of methods not traditional for the domain of time series analysis.

The diversity of data represented as time series allowed

[*]Recommended by thesis supervisor: Prof. Mária Bieliková
Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on [To be specified later].

development of a number of time series representations and similarity measures. The most popular similarity measures are those designed to compare time series in terms of their shape such as Euclidean distance or Dynamic Time Warping - DTW. These similarity measures, however, falter in case of very long time series, where global characteristics (such as mean or trend) are no longer essential for time series comparison, but we have to focus on local features and the time series structure. To solve these problems, various model based similarity measures and measures based on compression dissimilarity were developed. Recently another approach for comparison of such time series gained a lot of interest - approaches transforming the time series into a vector in a multidimensional space, where each dimension is defined by one feature extracted from the time series. These features may be composed of most important frequencies or some symbolic representation of the time series structure. In our work, we propose a time series representation, that transforms time series into a sequence of repeating shapes and these can be used to form such features.

1.2 Challenges and Methods of Stream Data Processing

Typical data analysis process can be separated into several steps: data collection, transformation, preprocessing, analysis itself, visualization and result interpretation. More and more often, it happens that we can not separate the data acquisition and data processing stage. We often have to perform the analysis on ever increasing data collections, we have to cope with data drift and we have to address changing data properties as they come in streams. When processing big amounts of data and especially when the data is growing fast and when it is changing over time, we face the problem that in time we process the data, it is no longer accurate. In such cases, we have to process the data incrementally, in time of their creation,

This kind of processing brings several limitations we have to cope with:

- Data is incoming online, in time of their creation.
- The processing system has no control over the order of incoming items. Most often, they are timestamped, but their order may change in the processing pipeline.
- The data stream may be unbound and its speed may change over time.
- Once the data element is processed, it is discarded or archived. It cannot be retrieved easily unless it is stored in raw or aggregated form in memory which is typically small relative to the size of the stream.

Those differences from processing of static collections of data introduce several issues and challenges into the processing of the data [6]:

- Handling of continuous flow of data at variable pace. This issue is composed of two separate problems: design of systems able to continuously process the data for indefinite period of time and load shedding of variable amount of incoming data to multiple processing elements.
- The processing itself can use only limited amount of memory while processing potentially unbound stream of data.
- Result accuracy is required while preserving thorough memory limitations and single-pass through the data restriction. Sampling techniques, approximation algorithms and window functions are viable means to cope with this limitation.
- Modelling changes of mining results over time. In some cases we are not interested only in the data analysis results, but also in changes of these results with continuously arriving new data. Most of existing algorithms can not show the change of the result over time, only the result itself.
- Model shifting due to variability of incoming data stream. In many data analysis tasks such as classification or clustering, the variation of the incoming data stream can produce the need for shifting of the built model. Methods for stream data analysis have to deal with such shifting and have to model the updates.

Recently, multiple tools and frameworks were developed to process such streams of data such as Kafka[1], Storm[2] or Spark[3] and many approaches for stream data processing and for handling these limitations were proposed. The most often used are approaches based on windowed operations, on concept drift detection [10] and based on approximate algorithms.

1.3 Open Problems and Thesis Goals

Based on the analysis of current state of time series processing methods and stream data processing, we found several groups of open problems and opportunities:

- problems related to processing of high-dimensional data,
- problems concerning comparison of very long time series,
- opportunities arising from employment of methods not directly applicable on continuous data of time series such as various methods from text processing domain and
- inability of many time series representations and similarity measures to process the data under constraints faced when processing potentially infinite streams of data.

Based on these problems, we stated three goals we will focus on:

- To reduce dimensionality of very long time series data by transforming them into a sequence of symbols, where every symbol will be represented by a repeating shape.

[1] Apache Kafka - http://kafka.apache.org
[2] Apache Storm - http://storm.apache.org
[3] Apache Spark - http://spark.apache.org

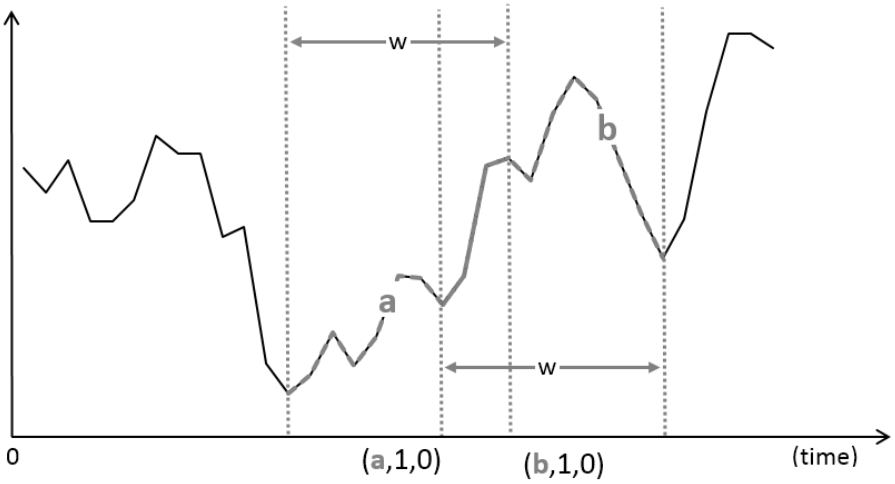

Figure 1: Sliding window (of length w) splits the time series into overlapping sequences. We cluster similar sequences to form groups of similar sequences represented by the same symbol. A sequence of symbol identifiers and normalization coefficients is used to represent the time series.

- To allow application of various methods from the domain of text processing by means of the symbolic representation.
- To propose a process of transformation of very long and potentially infinite time series into sequences of symbols while maintaining limitations imposed by the stream nature of the data.

2. Symbolic Time Series Representation

To solve open problems defined in previous section, we proposed a representation of time series data transforming repeating shapes in the course of time series into symbols. We refer to this representation as to Incremental Subsequence Clustering - ISC. The representation is based on a symbolic representation presented in [5], but it removes two of its major drawbacks.

The original representation used K-means algorithm to create clusters of subsequences, which were then used to replace the original subsequence with the cluster identifier. In theory, the cluster centre could be used to approximate the original data. However, it was later shown, that these clusters don't accurately represent the original data and tend to converge into a shifted sinusoidal curve regardless the data they were formed from [8].

The other limitation is the fact, that K-menas algorithm is iterative in its nature and thus not directly applicable in single pass through the data restriction of stream data processing.

To alleviate these two limitations, we use Leader clustering algorithm (not limiting the number of formed symbols, but rather their size) instead of the K-means algorithm in the transformation process.

A visualization of the transformation process of time series into the ISC representation is displayed on Figure 1. The data is split into overlapping subsequences of defined length and shift. All formed subsequences are normalized using Z-normalization and clustered incrementally. Every subsequence is then replaced by a cluster identifier and normalization coefficients. As cluster centres represent the data they were formed from, they can be used in connection with normalization coefficients to approximately reconstruct the original data.

To evaluate whether clusters formed using this representation truly represent the original data, we performed an evaluation of Meaningfulness [8] of formed clusters. The clustering of subsequences using the Leader algorithm produced meaningful clusters when symbol size was comparable with length of shapes typical for the dataset. However, when we shrank the symbol size into only a few data points, the symbol alphabet was reduced into a set of basic shapes, which are common to every dataset. Thus, meaningful clusters are formed in the process of transformation into ISC representation, but when we use too small symbols, they fall into a set of few basic shapes, which can be expected.

2.1 Similarity Measure on ISC Representation

To have an applicable time series representation, along with the representation, we proposed a similarity measure running on top of the ISC representation. The similarity measure (referenced as Symbolic distance - SymD) is an adaptation of Euclidean distance not comparing individual values, but whole symbols. We showed, the similarity measure lower bounds the Euclidean distance, which makes it applicable in indexing schemes such as GEMINI.

However, we don't limit the potential user of the proposed symbolic representation representation to use only the SymD distance. As this measure does not employ the fact that the time series is represented as a sequence of symbols, others, symbolic similarity measures can be used in its place. We performed several experiments, where we used Levenshtein distance to exploit the symbolic representation. Despite the fact that in this case we don't guarantee the lower bounding property, it is beneficial in many applications. Especially, when we are not interested in global features of the transformed time series, but their local features.

2.2 Alphabet Size Management

As we replaced the K-means algorithm, used in the time series representation proposed by Das et al. [5], by Leader clustering algorithm not limiting the number of clusters, we caused that the transformation into ISC representation produces ever growing alphabet of symbols. The ever growing alphabet of symbols is not permitted when processing streams of data as we are restricted for constant processing time per element of the data stream. This is especially troubling when transforming evolving data streams, where new shapes in the data can occur continuously and some shapes seize to occur over time.

To overcome this limitation, we proposed several approaches for alphabet size limitation based on forgetting non-frequent symbols. We used an assumption, that we are more interested in frequent and the most recent symbols than in old or unused symbols and thus we have to represent the recent part of the data stream with greater accuracy than the older parts of the data stream. When this assumption holds and the application at hand allows it, we could simply forget non-frequent or no longer used symbols.

To introduce the notion of frequent items (symbols) into the ISC representation, we extended the definition of a symbol by a occurrence count estimate. We used a counter based frequent item mining algorithm to maintain a set of counters for the most frequent item candidates and we merged set of these candidates and the alphabet of symbols. Every time a frequent item mining algorithm assigned a counter to new potentially frequent item, we associated it with a new symbol and every time a counter was removed from this set, we forgot a symbol. This approach, however, produces sequences of symbols with holes - sections that are impossible to reconstruct due to forgotten symbols representing them. We adapted the original forgetting approach to reduce and completely remove holes formed by forgotten symbols.

All proposed approaches for alphabet size reduction are not limited for one frequent item mining algorithm. All counter based and most of sketch based algorithms are applicable [4] as long as they maintain a set frequent item candidates. In our experiments, we used the Frequent algorithm, but various other algorithms can be used instead. For example time decaying [3] or sliding window based [2] algorithms for frequent item mining could be used to introduce decay and age of the symbols into alphabet management process.

3. Applications of the Proposed Representation

We evaluated properties and applicability of the proposed representation in multiple different applications on various short and long time series.

3.1 Classification of Various Short Time Series

We used the ISC representation in combination with two time series similarity measures (SymD and Levenshtein distance) to evaluate applicability on various types of time series data. For this experiemnt, we used the well known UCR collection of datasets [9].

We showed that the performance of various methods greatly varies on different dataset types. At the same time, we showed two methods running on ISC representation outperform other popular time series similarity measures running on raw data on multiple types of time series datasets.

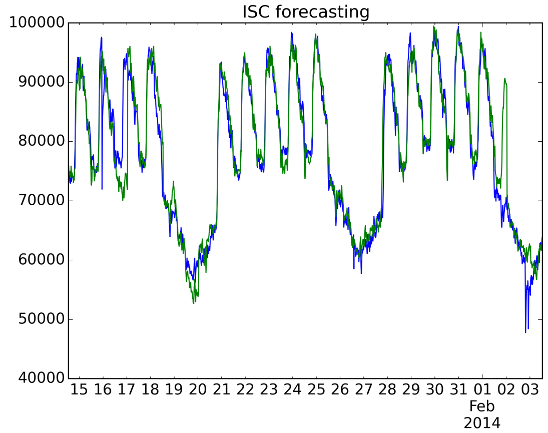

Figure 2: An example of forecast using the method based on ISC. Blue line indicates the true data and green line indicates 1/2 day forecast. Three weeks worth of data is displayed.

In general, it showed promising results and can be used to improve classification error on various datasets.

3.2 Forecasting

We used the fact, that we maintain an alphabet of overlapping shapes to forecast future values of a very long time series. We used a simple idea of searching for the closest symbol in the alphabet of symbols by comparison of the most recent part of the dataset with start of every symbol in the alphabet. As we found the closest symbol, we used the rest of the symbol as prediction of next values of the time series. On Figure 2, one can see a result of prediction created using 1/2 day of the most recent data to find the closest, one day long, symbol. The second half of the closest symbol was used as a prediction.

Our simple model could be trivially improved by taking into account frequent sequences of symbols and we could use it to prolong the predicted sequence and to increase forecast accuracy.

The simple model we used, was able to outperform Hot-Winters seasonal model and Exponential smoothing as they were not able to learn multiple pattern and they were slow to train new shapes in the data. We thus demonstrated, the ISC representation is also applicable on forecasting.

3.3 Classification of Next Symbol in Very Long Time Series

In another experiment, we used methods requiring categorical data (Markov model) on time series data, to increase precision of classification of data in incoming stream. We used the ISC representation to transform multivariate time series data representing an entire book of handwritten characters into a sequence of symbols. We used 1NN classifier to classify character trajectories incoming in stream and we used Markov model trained on a symbolic representation to increase classification accuracy with the knowledge about frequent symbol sequences. We were able to significantly increase the accuracy of handwritten symbol identification from multivariate time series using the frequent symbol occurrence model. We thus demonstrated applicability of the representation on multivariate time series and the merit of application of text processing methods on symbolic representation of time series.

3.4 Classification Of Very Long Time Series Using Bag-of-Words Representation

Text processing methods can be divided into those working on the level of symbols and strings (eg. edit distance measures) and those, working on the level of entire documents (eg. TF-IDF or Inverted index). The ISC representation of time series, we proposed, can be applicable not only to reduce dimensionality and to allow employment of edit based similarity measures, but also to use time series as bags-of-words. By transforming time series into a bag-of-words representation, we could use a number of diverse methods commonly used in text processing domain to process time series.

Multiple similar approaches were proposed recently [1, 12, 14] and in this experiment, we determined, whether ISC can achieve comparable results in analysis of very long time series data transformed into bag-of-words representation as one of the most popular bag-of-words representations of time series [12]based on SAX. To evaluate applicability of both representations, we use classification of various types of very long time series created by composition of shorter time series from UCR collection of datasets [9]. Similarly to results of classification on short time series, we achieved variable results for compared methods. On multiple datasets, we were able to outperform the SAX based method and thus the ISC based method can be used to represent time series as bags-of-words.

4. Contributions and Conclusions

By analysing state-of-the-art of time series representation, we identified several problems with processing of very long and potentially infinite time series data. At the same time, we identified an opportunity to introduce the plethora of methods from text processing domain into time series processing by transforming time series into symbols.

We addressed these open problems by proposing a symbolic representation of time series referenced as Incremental Subsequence Clustering (ISC), which allows incremental transformation of potentially infinite time series data into symbols. We proposed a similarity measure operating on data transformed into the ISC representation and we showed it lower bounds the Euclidean distance and can be thus used in indexing schemes such as GEMINI. We addressed the problem of ever growing alphabet of symbols by forgetting infrequent symbols and we demonstrated the applicability of the representation in various tasks of data analysis such as classification of short and long time series data or forecasting.

Acknowledgements. This work was partially supported by 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, the Scientific Grant Agency of The Slovak Republic, grant No. VG 1/0752/14 and VG 1/0646/15, the Slovak Research and Development Agency under the contract No. APVV-15-0508 and the Cultural and Educational Grant Agency of the Slovak Republic, grant No. KEGA 009STU-4/2014.

References

[1] A. Bailly, S. Malinowski, R. Tavenard, T. Guyet, and L. Chapel. Bag-of-temporal-sift-words for time series classification. In *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, 2015.

[2] H. Chen. Mining top-k frequent patterns over data streams sliding window. *Journal of Intelligent Information Systems*, 42(1):111–131, 2014.

[3] L. Chen and Q. Mei. Mining frequent items in data stream using time fading model. *Information Sciences*, 257:54–69, 2014.

[4] G. Cormode and M. Hadjieleftheriou. Finding frequent items in data streams. In *Proceedings of the VLDB Endowment*, pages 1530–1541, 2008.

[5] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, and P. Smyth. Rule discovery from time series. In *KDD*, volume 98, pages 16–22, 1998.

[6] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy. Mining data streams: a review. *ACM Sigmod Record*, 34(2):18–26, 2005.

[7] Hopkins Brian. Blogging From the IBM Big Data Symposium - Big Is More Than Just Big, 2011. Accessed: 2014-05-28.

[8] E. Keogh and J. Lin. Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowledge and Information Systems*, 8(2):154–177, 2004.

[9] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana. The ucr time series classification/clustering homepage, 2011.

[10] J. Z. Kolter and M. A. Maloof. Using additive expert ensembles to cope with concept drift. In *Proceedings of the 22nd international conference on Machine learning*, pages 449–456. ACM, 2005.

[11] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

[12] J. Lin, R. Khade, and Y. Li. Rotation-invariant similarity in time series using bag-of-patterns representation. *Journal of Intelligent Information Systems*, 39(2):287–315, 2012.

[13] A. Mueen and E. Keogh. Online discovery and maintenance of time series motifs. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1089–1098. ACM, 2010.

[14] J. Wang, P. Liu, M. F. She, S. Nahavandi, and A. Kouzani. Bag-of-words representation for biomedical time series classification. *Biomedical Signal Processing and Control*, 8(6):634–644, 2013.

Selected Papers by the Author

J. Ševcech, M. Bieliková. Repeating Patterns as Symbols for Long Time Series Representation. Journal of Systems and Software, Elsevier, CC, 2016. IF 2014 = 1.352

J. Ševcech, M. Bieliková. Symbolic Time Series Representation for Stream Data Processing. In Proceedings of 13th IEEE international conference on big data science and engineering, Helsinki, Finland. Los Alamitos, 217–222, 2015. IEEE

J. Ševcech, M. Bieliková. UserâĂŹs interest detection through eye tractking for related documents retrieval. In Proceedings of SMAP 2014, 9–13, 2014. IEEE.

M. Holub, R. Móro, J. Ševcech, M. Lipták, M. Bieliková. Annota: towards enriching scientific publications with semantics and user annotations. In D-lib Magazine. 20(11/12), 6–11, 2014.

J. Ševcech, R. Móro, M. Holub, M. Bieliková. User annotations as a context for related document search on the web and digital libraries. In Informatica. 38(1), 21–30, 2014.

M. Bieliková, M. Šimko, M. Barla, J. Tvarožek, M. Labaj, R. Móro, I. Srba, J. Ševcech. ALEF: from application to platform for adaptive collaborative learning. In Recommender systems for technology enhanced learning : research trends and applications. Springer, 195–225, 2014.