EuroVis Workshop on Visual Analytics (2020)
K. Vrotsou and C. Turkay (Editors)

# Progressive Parameter Space Visualization for Task-Driven SAX Configuration

S. Loeschcke[1], M. Hogräfer[1], and H.-J. Schulz[1]

[1] Aarhus University, Denmark

**Abstract**

*As time series datasets are growing in size, data reduction approaches like PAA and SAX are used to keep them storable and analyzable. Yet, finding the right trade-off between data reduction and remaining utility of the data is a challenging problem. So far, it is either done in a user-driven way and offloaded to the analyst, or it is determined in a purely data-driven, automated way. None of these approaches take the analytic task to be performed on the reduced data into account. Hence, we propose a task-driven parametrization of PAA and SAX through a parameter space visualization that shows the difference of progressively running a given analytic computation on the original and on the reduced data for a representative set of data samples. We illustrate our approach in the context of climate analysis on weather data and exoplanet detection on light curve data.*

**CCS Concepts**

• ***Human-centered computing*** → ***Visual analytics;*** • ***Applied computing*** → *Astronomy; Environmental sciences;*

## 1. Introduction

Time series data is ubiquitous as it is being collected by many sensors found in modern environments. Yet, analyzing these data is often time-consuming due to the vast number of measurements collected and the large value ranges that individual data points can cover. To nevertheless gain timely insights into such data, analysts often use data reduction methods to increase analytic throughput [SAAF18]. One approach for instance is reducing the temporal resolution of the data by aggregating data segments by their mean using a Piecewise Aggregate Approximation (PAA) [KCPM01]. Another approach is to reduce the resolution of the measured values, for instance by aggregating value ranges to symbolic values using Symbolic Aggregate approXimation (SAX) [LKLC03].

Both reduction methods are widely used for analyzing time series data [WLCB19]. However, while reducing the amount of processed data increases the performance of analysis algorithms, it at the same time decreases the accuracy of the results. Addressing this trade-off has been the focus of previous work, proposing *user-driven, manual approaches* and *data-driven, automated approaches* to find suitable parametrizations for PAA and SAX.

User-driven approaches ask the analysts to set the parameters based on their domain knowledge and expertise. The data is then reduced using these user-defined parameters. In some cases, the result is then being shown to allow the analysts to readjust the parameters in a trial&error fashion until the output fits their expectation. Visual analytic tools employing this approach are, for example, GrammarViz [SLW*14] and ARC-VIEW [NNW14].

Data-driven approaches rely on measures derived from the input dataset to determine suitable parameters for their reduction without any user input. These measures include descriptive statistics like standard deviation and frequency distributions, but also more elaborate characteristics like complexity estimates [CA15, ZY17]. The quality of these approaches is usually evaluated with a classification task on a standard benchmark dataset [BLB*17]. That other tasks – e.g., correlation or trend analyses – might benefit from other parametrizations seems not to be considered in these evaluations.

In this work, we fill this gap with a *task-driven, semi-automated approach* to the parametrization of PAA and SAX. To this end, we compute for different parameter combinations the discrepancy between the outputs of computational analyses being run on samples of the raw data and of the reduced data. Depending on the analytic task, a different computational analysis will be run – e.g., for finding periodicities, the greatest periodicity is computed, while for establishing trends a regression analysis is carried out. Visualizing the resulting discrepancies for the different parameter configurations allows the analyst to make an informed choice with respect to the gain from the data reduction and the loss from the error incurred specifically for the particular task at hand. To cope with the rather long runtimes for computing these discrepancies for many different parameter combinations, we employ a progressive computation that generates outputs of increasing quality and quantity [ASSS18]. We demonstrate this approach on two analysis tasks: finding periodic patterns in light intensity data to identify exoplanets, and analyzing trends in weather data to characterize local climates.

## 2. A Task-based Parametrization for PAA and SAX

The Symbolic Aggregate approXimation (SAX) and the Piecewise Aggregate Approximation (PAA) are usually used in concert by applying the SAX transformation on top of the PAA: $sax(paa(D,\omega),\alpha)$, with $D$ denoting the initial dataset.

The function $paa(D,\omega)$ reduces the number of time points in a dataset [KCPM01]. To do so, it takes time series data of arbitrary length and splits it into equal-sized segments of length $\omega$, for which the mean value is computed and used instead of the individual values of that segment. Thus, PAA reduces the temporal resolution of the dataset. For example, by setting $\omega = 2$, the aggregated data produced by PAA is only half of the original data size.

The function $sax(D,\alpha)$ reduces the possible values the time series can take on to a set of symbols [LKLC03]. To do so, it discretizes time series by mapping intervals of continuous data values to a discrete alphabet. These intervals are regions of equal probability from a normal distribution fitted around the mean of the data. The number of intervals and therefore the number of symbols in the output alphabet is the parameter $\alpha$ passed to the algorithm. Thus, SAX reduces the resolution of the values at each time point of the dataset. For example, setting $\alpha = 3$ will map all data values onto a 3-letter alphabet: high values to the letter `a`, values around the mean to the letter `b`, and low values to the letter `c`.

While value intervals used in the original SAX transformation [LKLC03] assume a normal distribution of the data, this is hardly true in general. Multiple approaches have been proposed to better reflect arbitrary distributions by, for example, including *min* and *max* measures in the computation to more accurately capture local patterns and trends [LYK06, MGQT13]. We utilize adaptive SAX (aSAX) [PLD10] to improve the fit of the symbolized representation to the data. It does so by computing the interval boundaries using representatives produced by Lloyd's algorithm [Llo82], basically performing a 1D $k$-means clustering of the value range.

### 2.1. Computing and Using the Task-based Error Measure

To determine the utility of the dataset after differently parametrized PAA and SAX transformations, we captured the analytic task by the computational function $f$ – e.g., estimating a trend using regression analysis. Computing this function yields a numerical result $f(D)$ – e.g., the slope of the regression line. We can then compare the results generated on the original data $f(D)$ and on the reduced data $f(sax(paa(D,\omega),\alpha)$ to determine how much error was introduced by the data reduction with parameters $\omega$ and $\alpha$. Note that in order to apply $f$ to the symbolic representation, we need to transform it back into numbers – e.g., by using the mean values of the intervals.

We represent the resulting error measures in a matrix visualization, where each column represents a PAA segment size $\omega$ and each row represents a SAX alphabet size $\alpha$. Here we use $1 \leq \omega \leq 15$ and $3 \leq \alpha \leq 20$ as potentially suitable parameter combinations, but any other parameter space size would also work. Each cell in the matrix shows the error measure on a continuous color scale (see Figures 1 and 3), where the smallest difference and thus the highest quality maps to the brightest color, and the biggest difference and thus the worst quality maps to the darkest color.

### 2.2. Progressive Parameter Space Visualization

Generating the matrix visualization for all parameter combinations and for the full dataset requires extensive computational efforts that are not achievable in reasonable time. One solution to this problem is not to wait for such exhaustive result to come back, but to visualize early partial results computed on subsets of the data that are then refined over time. This approach is known as Progressive Visual Analytics (PVA) [SPG14, MSA*19], and allows users to make decisions about the analysis early on by steering the computation towards regions of interest as these emerge [WM04, BEF17]. We use PVA to address the computational bottleneck of generating the parameter space visualization described in Sec. 2.1.

For this, we begin by selecting a representative sample of the full dataset. This selection is either done manually by the analyst or automated using appropriate algorithms. Then, we divide the sample into equal-sized chunks and iteratively compute the error measure for each cell in the matrix as discussed in the previous section. This produces partial results in a breadth-first manner: Instead of processing each cell in full one after another, the computations can now be run per chunk with the ability to obtain partial results for each cell. To this end, we use Latin Hypercube Sampling [Flo92] to determine the order in which the cells are processed, which avoids oversampling individual columns and rows. After each full iteration over the cells, we continue with the first cell and the next data chunk. In doing so, an early overview of the full parameter space is available early on.

The analyst can steer the computation towards parameter subspaces of interest through interaction. For this purpose, columns, rows, or individual cells of the matrix may be selected through clicking in order to focus the computational resources on them. The user may make multiple selections throughout the analysis process to add or remove cells from the selection. Once cells have been selected, the subroutine choosing the order of cells prioritizes those cells that are part of this selection, thereby steering the computation towards the user-selected region.

The visualization further includes progress bars for all columns and rows, indicating the degree two which the data has been processed across the particular set of parameter combinations (see Figure 2). This serves as a visual guide for the user to gauge both the quality of the partial results as well as the progress in overall computation [AMSS19]. An implementation in Python adapted for the following two application scenarios can be found on `https://github.com/vis-au/ExoVis`.

## 3. Application to Light Curves

We first evaluated our approach with publicly available data on light intensity emitted by stars that was measured by the Kepler space telescope during a NASA mission on discovering exoplanets between 2009 and 2013 [BKB*10, Bor16]. Using the Python libraries Lightkurve [LCH*18] and AstroPy [Thea], we make use of a cleaned version of that data provided by the Mikulski Archive for Space Telescopes (MAST) [Theb]. For an individual region covered by the telescope, the dataset contains measurements of 30-60 minutes intervals for periods of around 90 days.

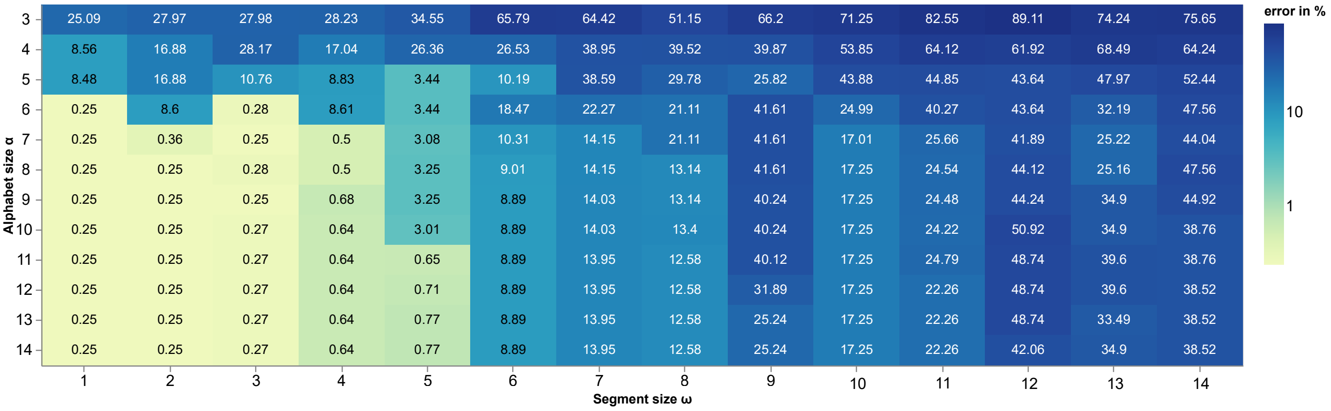

**Figure 1:** *Parameter space visualization showing the result of a complete computation of the average error between the BLS computed on a sample of light curve data for six exoplanets and the computation on the same data aggregated with SAX and PAA. The aggregated data using only one third of the original data ($\omega = 3$) while reducing its value range to an alphabet of only six symbols ($\alpha = 10$) yields only a minor computational error in comparison to using the unaggregated data.*

To find extrasolar planets in these datasets, one is looking for periodic patterns. These are produced when a planet orbits around a central host star and periodically reduces the light intensity of that star during transit from the view point of the telescope [CBBL07]. The light intensity deviates significantly between host stars, and as a result, lower light intensities measured for one star may be larger than the highest intensities measured for other stars. That is why one rather wants to look for periodicities in the light intensity curves as opposed to identifying exoplanets through absolute measurements.

To do so, we utilize the Box-fitting Least Squares (BLS) [KZM02] algorithm that searches for periodic alternations between two intensity levels in the data. Its result is the period with the highest, non-trivial periodicity coefficient. As the dataset is vast, these computations are too time-intensive to run exhaustively on the full data which is why reducing the data using PAA and SAX is a suitable approach. As the error measure to be shown in the matrix, we use the absolute difference between the periods computed on the original and on the reduced data. As data sample for the computations, we use the light curve data for six planets gathered from MAST. As parameter space, we utilized the value ranges of $1 \leq \omega \leq 15$ and $3 \leq \alpha \leq 20$.

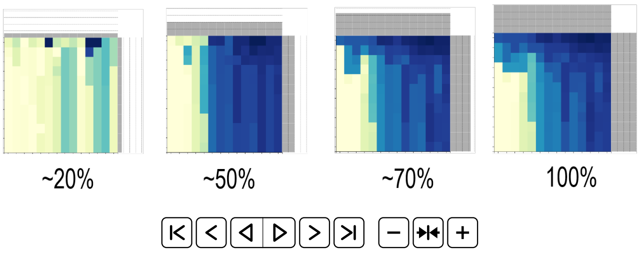

**Figure 2:** *Partial results of the progressive computation for the parameter space depicted in Figure 1, including progress bars for columns and rows. Trends in the data found in the full computation gradually become apparent in partial results. A user may thus decide on a parametrization for SAX and PAA before the full matrix is even generated, based on the task at hand. To view the animated figure, a stand-alone PDF viewer is needed.*

Figure 1 shows the parameter space visualization for six lightcurve measurements from the Kepler mission (with Kepler ID's 2 to 7). A light curve for a planet orbiting a star contains a relatively constant signal that at periodic intervals exhibits short sections of significantly lower light intensity, whenever the planet transits between its host star and the telescope sensor. Because of this property, the light-curve analysis is not very sensitive towards the particular numeric values at each time point, but to capture these brief transit periods we need to maintain a high temporal resolution. The parametrization of the data reduction reflects these properties in that the error measure varies more strongly between neighboring columns than between neighboring rows. To determine a suitable parametrization, the analyst may first identify a suitable PAA segment size $\omega$ along the horizontal axis, before determining a sufficient alphabet size $\alpha$ for SAX. One plausible pair of parameters is for instance $\alpha = 10$ and $\omega = 3$, resulting in a reduction of the overall data by approximately 96% compared to the size original dataset. In an exemplar computation of BLS on the first 10 quarters of exoplanet kepler-8b, this parametrization lead to a computational speed-up from 61.7 down to 20.6 minutes, thus decreasing the computation time by approximately 66%.

## 4. Application to Weather Data

The second context in which we evaluated our approach is the publicly available dataset on global weather data, provided by Berkeley Earth [MM]. The data contains multiple parameters about the weather for different countries, measured in monthly time intervals from as early as 1754 until 2013. For each country, the data thus contains around 3,000 measurements. Therefore, the dataset con-

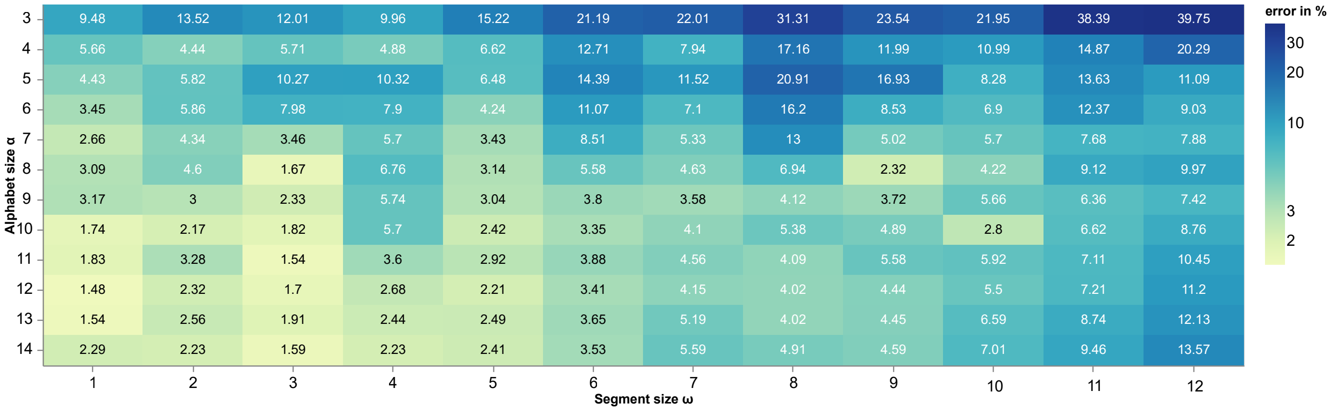

**Figure 3:** *Parameter space visualization showing the result of a complete computation of the average error between a linear regression model computed on a sample of weather data from ten countries and computed on the same data aggregated with SAX and PAA. The overall computational error using aggregated data compared to the original is low. Analysts may thus decide to use aggregated data for their computation, for instance using only a fifth of the overall data ($\omega = 5$) and reducing the value range to 12 symbols ($\alpha = 12$).*

taining all countries is reasonably large to provide a feasible target for its reduction.

One analytic task of current interest to be performed on this data is to assess climatic trends in certain regions of the Earth – i.e., to quantify the climate change. For this purpose, data capturing different aspects of the weather such as air and surface temperatures, humidity, precipitation, and wind speeds are considered. In order to establish trend in such data, Different variants of regression analysis are commonly used to describe climate development [Mud19].

For our purposes, we compute a linear regression model for the data and use the difference in slope of the resulting regression line as error measure for assessing parameter combinations for SAX and PAA. We base our assessment on a data sample of temperature data for ten countries (including Afghanistan, Denmark, and Portugal). A suitable parametrization found for this sample then serves as basis configuration for assessing the climate in other regions. As parameter space, we maintained the configurations of $1 \leq \omega \leq 15$ and $3 \leq \alpha \leq 20$.

The resulting parameter space matrix is depicted in Figure 3. As temperature changes gradually over the course of a year, the temporal resolution is not as important in this case as it was in the previous scenario. Yet, the resolution of the value range is much more important here, as small deviations in the values can have a large influence on the resulting slope. The matrix visualization reflects this property in that the error measure varies stronger between neighboring rows than between neighboring columns. To determine a suitable parameter combination, analysts may therefore first identify an alphabet size $\alpha$ that produces sufficient accuracy of the slope of regression line before identifying a segment size $\omega$ that includes enough data points. One plausible pair of parameters is for instance $\alpha = 10$ and $\omega = 3$, resulting in a reduction of the overall data by approximately 93% compared to the size of the original dataset. In an exemplar computation of a linear regression models on the weather data consisting of around $2,509$ entries, this parametrization again lead to a computational speed-up of approximately two thirds.

## 5. Summary

In this paper, we presented a task-driven approach to parametrization of PAA/SAX-based data reduction. With this approach, we are able to capture that some tasks are more sensitive to reductions of the temporal granularity, while others depend more on the granularity of the value range. This result-oriented approach to parameter estimation is made possible through PVA methods for computing partial error measures for multiple parameter combinations and thereby being able to show them to the analyst long before an exhaustive computation would be complete. The user can interactively steer the progression of these computations towards regions of interest in the parameter space, and finally make an informed decision for a suitable parametrization.

While we were able to address particular user requirements, our approach makes the following assumptions about the data and algorithms used. For instance, computing the error measure assumes a representative sample from the dataset on which the ground truth value can be computed. For time series data, progressive, representative sampling is a non-trivial task, in particular when aiming to preserve local patterns. This is a standing problem in the field of PVA and needs to be addressed in future work. Currently, our implementation of the iterative matrix computation relies on a single-threaded implementation. Yet, since the computation for each cell can be done independently of the remainder of the matrix, it is straight-forward to extend our code with a multi-threaded implementation in the future for further increased performance. Another assumption is that the algorithm that is used to compute the error measure can produce meaningful, progressive results. Implementations for such algorithms often require a manual, “progressive” re-implementation, or cannot be made progressive at all.

## Acknowledgements

The authors thank Simon Albrecht for his invaluable input on this project, as well as Kevin Wenkai Han and the anonymous reviewers for their feedback on the paper.

## References

[AMSS19] Angelini M., May T., Santucci G., Schulz H.-J.: On Quality Indicators for Progressive Visual Analytics. In *Proc. of EuroVA* (2019), Eurographics Association, pp. 25–29. doi:10.2312/eurova.20191120. 2

[ASSS18] Angelini M., Santucci G., Schumann H., Schulz H.-J.: A Review and Characterization of Progressive Visual Analytics. *Informatics 5*, 3 (2018), 31:1–31:27. doi:10.3390/informatics5030031. 1

[BEF17] Badam S. K., Elmqvist N., Fekete J.-D.: Steering the Craft: UI Elements and Visualizations for Supporting Progressive Visual Analytics. *Computer Graphics Forum 36*, 3 (2017), 491–502. doi:10.1111/cgf.13205. 2

[BKB*10] Borucki W. J., Koch D., Basri G., Batalha N., Brown T., Caldwell D., Caldwell J., Christensen-Dalsgaard J., Cochran W. D., DeVore E., Dunham E. W., Dupree A. K., Gautier T. N., Geary J. C., Gilliland R., Gould A., Howell S. B., Jenkins J. M., Kondo Y., Latham D. W., Marcy G. W., Meibom S., Kjeldsen H., Lissauer J. J., Monet D. G., Morrison D., Sasselov D., Tarter J., Boss A., Brownlee D., Owen T., Buzasi D., Charbonneau D., Doyle L., Fortney J., Ford E. B., Holman M. J., Seager S., Steffen J. H., Welsh W. F., Rowe J., Anderson H., Buchhave L., Ciardi D., Walkowicz L., Sherry W., Horch E., Isaacson H., Everett M. E., Fischer D., Torres G., Johnson J. A., Endl M., MacQueen P., Bryson S. T., Dotson J., Haas M., Kolodziejczak J., Van Cleve J., Chandrasekaran H., Twicken J. D., Quintana E. V., Clarke B. D., Allen C., Li J., Wu H., Tenenbaum P., Verner E., Bruhweiler F., Barnes J., Prsa A.: Kepler Planet-Detection Mission: Introduction and First Results. *Science 327*, 5968 (2010), 977–980. doi:10.1126/science.1185402. 2

[BLB*17] Bagnall A., Lines J., Bostrom A., Large J., Keogh E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery 31*, 3 (2017), 606–660. doi:10.1007/s10618-016-0483-9. 1

[Bor16] Borucki W. J.: KEPLER Mission: development and overview. *Reports on Progress in Physics 79*, 3 (2016), 036901. doi:10.1088/0034-4885/79/3/036901. 2

[CA15] Castro N. C., Azevedo P. J.: Automatically estimating iSAX parameters. *Intelligent Data Analysis 19*, 3 (2015), 581–595. doi:10.3233/IDA-150733. 1

[CBBL07] Charbonneau D., Brown T. M., Burrows A., Laughlin G.: When Extrasolar Planets Transit Their Parent Stars. In *Protostars and Planets V* (2007), University of Arizona Press, pp. 701–716. arXiv:astro-ph/0603376. 3

[Flo92] Florian A.: An efficient sampling scheme: Updated Latin Hypercube Sampling. *Probabilistic Engineering Mechanics 7*, 2 (1992), 123–130. doi:10.1016/0266-8920(92)90015-A. 2

[KCPM01] Keogh E., Chakrabarti K., Pazzani M., Mehrotra S.: Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowledge and Information Systems 3*, 3 (2001), 263–286. doi:10.1007/PL00011669. 1, 2

[KZM02] Kovács G., Zucker S., Mazeh T.: A box-fitting algorithm in the search for periodic transits. *Astronomy & Astrophysics 391*, 1 (2002), 369–377. doi:10.1051/0004-6361:20020802. 3

[LCH*18] Lightkurve Collaboration, Cardoso J. V. d. M., Hedges C., Gully-Santiago M., Saunders N., Cody A. M., Barclay T., Hall O., Sagear S., Turtelboom E., Zhang J., Tzanidakis A., Mighell K., Coughlin J., Bell K., Berta-Thompson Z., Williams P., Dotson J., Barentsen G.: Lightkurve: Kepler and TESS time series analysis in Python, Dec. 2018. URL: http://adsabs.harvard.edu/abs/2018ascl.soft12013L. 2

[LKLC03] Lin J., Keogh E., Lonardi S., Chiu B.: A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In *Proc. of 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (2003), DMKD '03, ACM, pp. 2–11. doi:10.1145/882082.882086. 1, 2

[Llo82] Lloyd S.: Least squares quantization in PCM. *IEEE Transactions on Information Theory 28*, 2 (1982), 129–137. doi:10.1109/TIT.1982.1056489. 2

[LYK06] Lkhagva B., Yu Suzuki, Kawagoe K.: New Time Series Data Representation ESAX for Financial Applications. In *Proc. of ICDEW* (2006), IEEE, pp. 17–22. doi:10.1109/ICDEW.2006.99. 2

[MGQT13] Malinowski S., Guyet T., Quiniou R., Tavenard R.: 1d-SAX: A Novel Symbolic Representation for Time Series. In *Advances in Intelligent Data Analysis XII* (2013), Springer, pp. 273–284. doi:10.1007/978-3-642-41398-8_24. 2

[MM] Muller E., Muller R.: Berkeley Earth Weather Data. URL: http://berkeleyearth.org/ [cited 26-02-2020]. 3

[MSA*19] Micallef L., Schulz H.-J., Angelini M., Aupetit M., Chang R., Kohlhammer J., Perer A., Santucci G.: The Human User in Progressive Visual Analytics. In *Short Paper Proc. of EuroVis* (2019), Eurographics Association, pp. 19–23. doi:10.2312/evs.20191164. 2

[Mud19] Mudelsee M.: Trend analysis of climate time series: A review of methods. *Earth-Science Reviews 190* (2019), 310–322. doi:10.1016/j.earscirev.2018.12.005. 4

[NNW14] Nguyen H.-L., Ng W.-K., Woon Y.-K.: Closed motifs for streaming time series classification. *Knowledge and Information Systems 41*, 1 (2014), 101–125. doi:10.1007/s10115-013-0662-6. 1

[PLD10] Pham N. D., Le Q. L., Dang T. K.: Two Novel Adaptive Symbolic Representations for Similarity Search in Time Series Databases. In *Proc. of the International Asia-Pacific Web Conference* (2010), IEEE, pp. 181–187. doi:10.1109/APWeb.2010.23. 2

[SAAF18] Shurkhovetskyy G., Andrienko N., Andrienko G., Fuchs G.: Data Abstraction for Visualizing Large Time Series. *Computer Graphics Forum 37*, 1 (2018), 125–144. doi:10.1111/cgf.13237. 1

[SLW*14] Senin P., Lin J., Wang X., Oates T., Gandhi S., Boedihardjo A. P., Chen C., Frankenstein S., Lerner M.: GrammarViz 2.0: A Tool for Grammar-Based Pattern Discovery in Time Series. In *Machine Learning and Knowledge Discovery in Databases* (2014), Springer, pp. 468–472. doi:10.1007/978-3-662-44845-8_37. 1

[SPG14] Stolper C. D., Perer A., Gotz D.: Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 1653–1662. doi:10.1109/TVCG.2014.2346574. 2

[Thea] The AstroPy Project: Homepage of the AstroPy project. URL: https://mast.stsci.edu/portal/Mashup/Clients/Mast/Portal.html [cited 26-02-2020]. 2

[Theb] The Barbara Mikulski Archive for Satellite Telescopes: MAST Portal. URL: https://mast.stsci.edu/portal/Mashup/Clients/Mast/Portal.html [cited 26-02-2020]. 2

[WLCB19] Wang L., Lu F., Cui M., Bao Y.: Survey of Methods for Time Series Symbolic Aggregate Approximation. In *Data Science* (2019), Springer, pp. 645–657. doi:10.1007/978-981-15-0118-0_50. 1

[WM04] Williams M., Munzner T.: Steerable, Progressive Multidimensional Scaling. In *Proc. of VIS* (2004), IEEE, pp. 57–64. doi:10.1109/INFVIS.2004.60. 2

[ZY17] Zan C. T., Yamana H.: Dynamic SAX parameter estimation for time series. *International Journal of Web Information Systems 13*, 4 (2017), 387–404. doi:10.1108/IJWIS-04-2017-0035. 1