University of Zurich[UZH]

Department of Informatics

University of Zürich
Department of Informatics
Binzmühlestr. 14
CH-8050 Zürich
Phone. +41 44 635 43 11
Fax +41 44 635 68 09
www.ifi.uzh.ch/dbtg

UZH, Dept. of Informatics, Binzmühlestr. 14, CH-8050 Zürich

Prof. Dr. Michael Böhlen
Professor
Phone +41 44 635 43 33
Fax +41 44 635 68 09
boehlen@ifi.uzh.ch

Zürich, August 16, 2024

Master Project (15 ECTS)
Database Technology

Topic: Scalable and Flexible Downsampling of Time-series Data

A time-series database system (TSDBS) is designed to handle huge amounts of data that are time-stamped with time points. Such data must be tracked, stored, downsampled, and aggregated over time so that the data can be analyzed efficiently. A TSDBS leverages the specific characteristics of time series data to store and process data more efficiently than general-purpose database systems.

Timescale `https://github.com/timescale/timescaledb` is an open-source database system designed to make database technology scalable for time-series data and is implemented as a PostgreSQL extension. The Largest Triangle Three Buckets (LTTB) algorithm [3] `https://github.com/timescale/timescaledb-toolkit/blob/main/extension/src/lttb.rs`, is a downsampling algorithm that strives to retain visual similarity between the downsampled data and the original dataset. The idea of LLTB is to select data points that form the largest triangular area with a previously selected data point and the average value of the next bin. Although LTTB is effective, there are limitations that shall be investigated and improved in this Master project.

Task 1: Scalable LLTB

Visualization plays an important role in analyzing and exploring time series data. To facilitate efficient visualization of large datasets, downsampling has emerged as a well-established approach. LTTB is a widely adopted downsampling algorithm for time series data point selection, however it becomes inefficient when applied to high-frequency data over extended periods. The goals of this task are

- To enhance the performance of LTTB by implementing the MinMaxLTTB algorithm [1] and apply it to different signals.
- Evaluate the quality of MinMaxLTTB. Possible approaches are to check the visual fidelity, calculate root mean square error between original and resampled signals or compare it to other resampling algorithms [2]. Part of this task is to evaluate options to measure the quality of the algorithm. The goal here is to know for which kind of signals the algorithm performs well and for which not.
- Identify signals for which the algorithm performs well or poor. Construct a signal where the algorithm would fail if there is any.

Implementing MinMaxLTTB in the productive system is out of scope for this task.

Task 2: Aggregated LLTB

In its current form, LTTB allows visualization of individual time-series signals separately, but for a lot of industry application the sum of multiple signals is very important. An example is the total power output of a hydro power plant. The following plot shows the power output of 2 turbines, and we would like to know the total power output of the whole power plant.

So here we need to sum up the two signals to visually return the total power output. The goals of this task are:

- Familiarize with the lttb algorithm, i.e., understand how the algorithm works etc. See [?]
- Extend the algorithm that allows operations like lttb(sum(sig1, sig2)) lets call it LTTB_SUM
- A version of LTTB_SUM is implemented, tested and evaluated with the method(s) from Task 1.
- Extend the found algorithm to a more generic form i.e. lttb(func(x1, x2, x3,...)), where func could be any aggregation function like sum, minimum, maximum, or average and it can be apply to n signals.

Considerations for Task 2

Sensor data is usually not regularly sampled but in order to sum them up mathematically they need to be regularly sampled first. Since this is very time consuming and also storage intensive it is not feasible for more then 5000 Signals to pre-resample all signals. (Axpo has more than 200'000 signals).

The algorithm is used for visualization purposes only. No further calculations like integrating or others will be performed on the output of this algorithm.

LTTB in it's current form returns raw signal data once zoomed in enough, e.g., if n_out is 2000 and the underlying signal data points are smaller then 2000. A smart solution for this should be found for LTTB_SUM.

Task 3: Multi-Dimensional LLTB (Optional)

Presently, LTTB is limited to processing one single signal at a time (e.g., lttb(sig1)). The goal is to expand LTTB to handle multiple signals simultaneously (e.g. lttb(sig1, sig2, ...)). If n signals are needed to be visualized at the same time, LTTB needs to be called n times. If LTTB can be extended to multi dimension one can find the optimal n for multi signal visualizations.

Task 4: Report

Describe your solution in a report.

References

[1] J. Donckt, J. Donckt, M. Rademaker, and S. Hoecke. Minmaxlttb: Leveraging minmax-preselection to scale lttb. In *2023 IEEE Visualization and Visual Analytics (VIS)*, pages 21–25, Los Alamitos, CA, USA, oct 2023. IEEE Computer Society.

[2] David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10:112–122, 1973.

[3] Sveinn Steinarsson. Downsampling time series for visual representation. Master's thesis, School of Engineering and Natural Sciences, University of Iceland, 2013.

Supervisor: Michael Böhlen (`boehlen@ifi.uzh.ch`), Emanuel Joos (`Emanuel.Joos@axpo.com`)

Start date: August 18, 2024

End date: August 18, 2025

University of Zurich
Department of Informatics

Prof. Dr. Michael Böhlen
Professor