The Journal of Open Source Software

MPA: a novel cross-language API for time series analysis

Andrew H. Van Benschoten[1], **Austin Ouyang**[1], **Francisco Bischoff**[1, 2, 3], **and Tyler W. Marrs**[1]

1 Matrix Profile Foundation **2** MEDCIDS-FMUP **3** AI4HEALTH-CINTESIS

DOI: 10.21105/joss.02179

Software

- Review
- Repository
- Archive

Editor: Arfon Smith

Reviewers:

- @coljac
- @krystophny

Submitted: 10 March 2020
Published: 06 May 2020

Summary

Two fundamental tasks in time series analysis are identifying anomalous events ("discords") and repeated patterns ("motifs"). Successfully accomplishing these tasks is of the utmost importance across many disciplines, and can lead to powerful technological advancements, prevention of catastrophic failures and the generation of significant economic gain. Dozens of algorithms have been developed to solve these problems, including AR(I)MA regression (Däubener, Schmitt, Wang, Bäck, & Krause, 2019), Hierarchical Temporal Memory (Ahmad & Purdy, 2016), Extreme Studentized Deviate (Däubener et al., 2019) and Artificial Neural Networks (Bishop, 2006). Unfortunately, these approaches are hampered by a combination of steep methodological learning curves, numerous parameters that require tuning and the inability to scale across large datasets (Yeh et al., 2016). The explosive growth of the data science community provides an additional hurdle for traditional time series analysis methods, as many practitioners lack experience in advanced mathematical and statistical principles. In this paper we present MPA (the *Matrix Profile API*) as a solution to all of these challenges. MPA is a cross-language platform in Python (*matrixprofile*), R (*tsmp*) and Golang (*go-matrixprofile*) that leverages a novel data transformation known as the Matrix Profile (Yeh et al., 2016) to rapidly identify motifs and discords. Perhaps most importantly, MPA is an easy-to-use API that's relevant for time series novices and experts alike.

The intuition behind Matrix Profile is straightforward. It begins with a snippet of data and then slides it along the rest of the time series, calculating the overlap at each new position. More specifically, it evaluates the Euclidean distance between a subsequence and every possible time series segment of the same length, building up the snippet's "Distance Profile." If the subsequence repeats itself in the data, there will be at least one perfect match and the minimum Euclidean distance will be zero, or close to zero in the presence of noise. In contrast, if the subsequence is highly unique due to the presence of outliers, matches will be poor and all overlap scores will be high. Every possible snippet is slid across the time series, building up a collection of Distance Profiles. The minimum value for each time step across all distance profiles is collected, creating the time series' final Matrix Profile. Both ends of the Matrix Profile value spectrum are useful. High values indicate uncommon patterns or anomalous events; in contrast, low values highlight repeatable motifs.

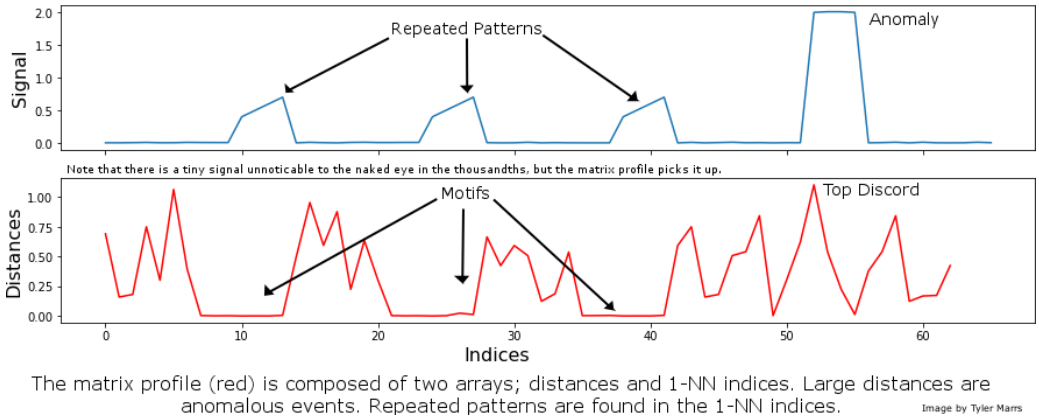

Figure 1: Overview of the Matrix Profile.

The Matrix Profile scales extremely well when applied to large datasets, as demonstrated in several recent publications (Gharghabi et al., 2017; Zhu et al., 2016). Its usage requires the selection of only a single parameter *k*, which is the length of the subsequence for which Euclidean distances are calculated. The recent formulation of the pan-Matrix Profile (Madrid et al., 2019) simplifies this result even further, as it creates a global calculation of all possible subsequence lengths condensed into a single visual summary (*Figure 2*). The X-axis is the index of the Matrix Profile, and the Y-axis is the corresponding subsequence length. The darker the shade, the lower the Euclidean distance at that point. Thus, the pan-Matrix Profile enables truly naive exploration of any time series, which can then be examined in more detail for greater understanding.

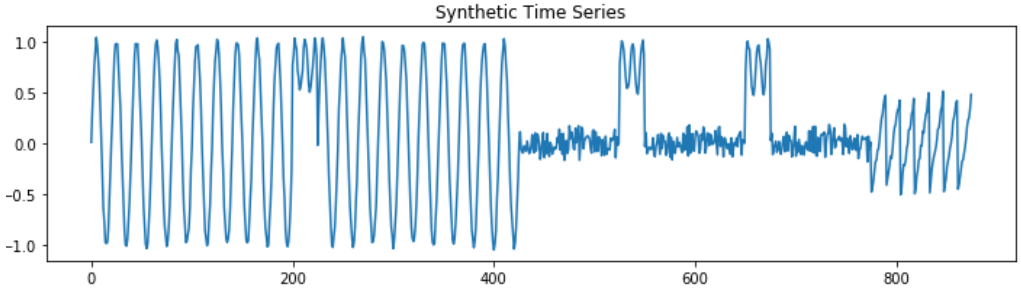

Figure 2: A synthetic time series.

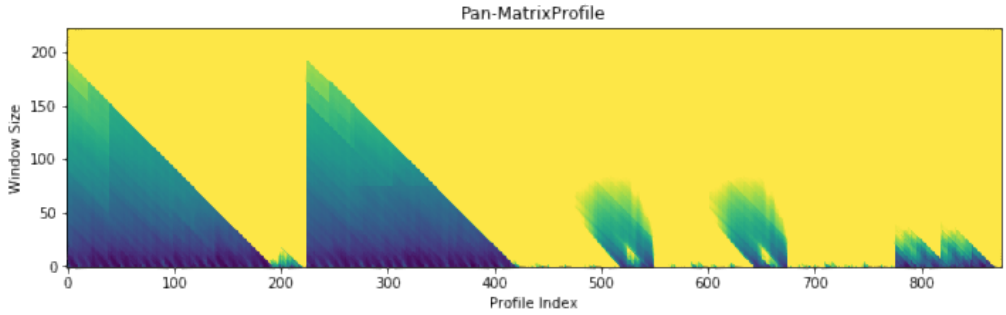

Figure 3: The pan-Matrix Profile of the time series in Figure 2.

Although the Matrix Profile can be a game-changer for time series analysis, leveraging it

to produce insights is a non-trivial, multi-step computational process.The original MATLAB code released by UCR(Yeh et al., 2016), as well as the first implementations in each major programming language (Python: matrixprofile-ts, R: tsmp, Golang: go-matrixprofile) each require some degree of technical understanding. This has also been the approach of more recent Matrix Profile implementations (Python: STUMPY, Law (2019); C++: SCAMP), where the target audience is primarily advanced practitioners. MPA removes all barriers to entry through three unique facets: an "out-of-the-box" working implementation, gentle introductions to core concepts that naturally lead into deeper exploration, and multi-language accessibility.

To standardize the natural flow of using the Matrix Profile, MPA consists of three core API components: 1. *Compute*, which computes the Matrix Profile, 2. *Discover*, which provides methods for evaluating the MP for motifs & discords and 3. *Visualize*, which displays results through basic plots. These three capabilities are wrapped up into a high-level capability called *Analyze*, a user-friendly interface that enables individuals lacking prior knowledge about the inner workings of Matrix Profile to quickly leverage it on their own data. With a single line of code, *Analyze* combines the pan-Matrix Profile with an under the hood algorithm to choose the three most sensible motifs and discords from across all possible window sizes. As users gain more experience and intuition with MPA and its outputs, they can easily dive deeper into any of the three core components to make further functional gains.

As data footprints continue to expand, the need for more robust time series methodologies will grow in lockstep. MPA provides an effective solution to this challenge that can simultanously unlock the potential of seasoned statistical veterans and brand-new data scientists across a myriad of applications.

Acknowledgements

All authors contributed equally to this work.

The authors would like to thank their fellow Matrix Profile Foundation board members Jack Green and Frankie Cancino, as well as Eamonn Keogh, Abdullah Mueen and their numerous graduate students for creating the Matrix Profile and continuing to drive its development.

References

Ahmad, S., & Purdy, S. (2016). Real-time anomaly detection for streaming analytics. Retrieved from http://arxiv.org/abs/1607.02480

Bishop, C. M. (2006). *Real-time anomaly detection for streaming analytics*. Information Science; Statistics, Springer, New York.

Däubener, S., Schmitt, S., Wang, H., Bäck, T., & Krause, P. (2019). Anomaly Detection in Univariate Time Series: An Empirical Comparison of Machine Learning Algorithms. In *Industrial Conference on Data Mining (ICDM)*. IEEE.

Gharghabi, S., Ding, Y., Yeh, C.-C. M., Kamgar, K., Ulanova, L., & Keogh, E. (2017). Matrix Profile VIII: Domain Agnostic Online Semantic Segmentation at Superhuman Performance Levels. In *International Conference on Data Mining (ICDM)* (pp. 117–126). IEEE. doi:10.1109/ICDM.2017.21

Law, S. (2019). STUMPY: A powerful and scalable python library for time series data mining. *Journal of Open Source Software*, *4*(39), 1504. doi:10.21105/joss.01504

Madrid, F., Imani, S., Mercer, R., Zimmerman, Z., Shakibay, N., & Keogh, E. (2019). Matrix Profile XX: Finding and Visualizing Time Series Motifs of All Lengths using the Matrix

Profile. In *International Conference on Big Knowledge (ICBK)* (pp. 175–182). IEEE. doi:10.1109/ICBK.2019.00031

Yeh, C.-C. M., Zhu, Y., Ulanova, L., Begum, N., Ding, Y., Dau, H. A., Silva, D. F., et al. (2016). Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. In *International Conference on Data Mining (ICDM)* (pp. 1317–1322). IEEE. doi:10.1109/ICDM.2016.0179

Zhu, Y., Zimmerman, Z., Senobari, N. S., Yeh, C.-C. M., Funning, G., Mueen, A., Brisk, P., et al. (2016). Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins. In *International Conference on Data Mining (ICDM)* (pp. 739–748). IEEE. doi:10.1109/ICDM.2016.0085