Forecasting Time Series, RI+LLM Edition

Yuliy Baryshnikov

1 Large Models for Time Series

Analysis of time series, and, especially, forecasting of the future segments based on the historic observation is a highly advanced area of applied statistics, with a lot of efforts poured into developing of ever more sophisticated methods. Traditionally, the deployed tools were model-based.

Lately this began to change, with the advent of AI, and, especially, Large Language Models (LLMs). The obvious analogy between texts (sequences of symbols rendering some meaning) and time series (sequences of numbers, doing the same) motivated the researchers to reuse, adopt or adapt large language models to model and predict time series.

This idea has many versions and implementations. The most straightforward (if lazy) implementations map the some short sequences of time series values onto a collection of lexemes, and then applies one of the pretrained LLM, hoping that the zero-shot properties of the model will carry over to such artificially generated texts, see, e.g., [8,9], with a survey of such tools in [10].

A more principled route was proposed recently in [2]. In that study, the time series are *tokenized*, essentially by discretizing windows of observations of fixed length, and declaring the resulting sequences tokens. Upon the tokenization, an LLM based on a transformer architecture was designed and trained on a broad variety of time series, resulting in a seemingly universal (zero-shot) time series forecaster.

This approach showed some significant improvements over the previous techniques, but has some obvious limitations which we will discuss below.

1.1 Time Series and Continuous Functions

In this proposal we use extensively the intuitive correspondences between the continuous functions on an interval and time series: in one direction, the correspondence operates by sampling (usually, at equispaced observation instances), in the other, by (say, linear) interpolation.

2 Reparameterizational Ambiguities

A standard complication in time series is missing, or duplicated data. More generally, the measurements can be sampled at irregular intervals, rendering identical functions as rather different time series. This type of data contamination is very common, and is especially detrimental for the data analysis tools relying on assumptions of homogeneity of sampling, like the model-based tools of time series analysis mentioned above.

This phenomenon of *reparameterization ambiguity* is especially common when one considers function of a continuous variable (time) (see Figure 1).

Specifically, reparameterization ambiguity happens when rather then observing a function $f : I \to \mathbb{R}, I = [0, T]$, one is presented with the function $f \circ s$, where s is a monotonically increasing reparameterization of the domain, $s : I' \to I$, where $I' = [0, T']$. (In real life this manifests in *cyclic* but *non-periodic* phenomena, like cardiac or business cycles, see Figure 2.)

This ambiguity makes application of any tools using Fourier transform (passing to frequency domain) non-robust and requiring ad hoc fixes. Nonetheless, it is often ignored for lack of first principle solutions.

2.1 Reparameterization Invariant Functionals

Ideally, one needs to develop data analysis tools that are impervious to the reparameterization ambigui-

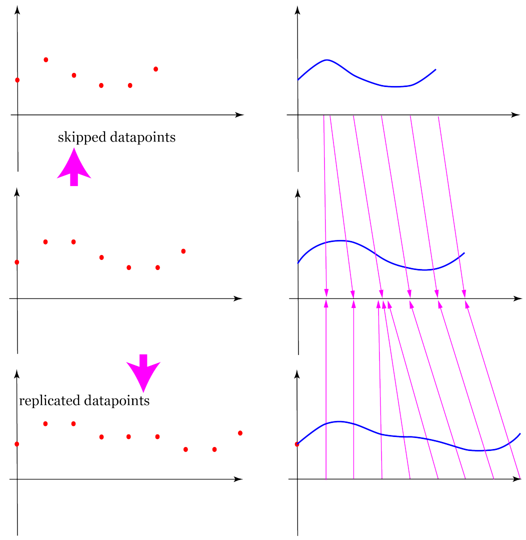

Figure 1: Omission or duplication of the datapoints in time series correspond to reparameterization of the domain. Left display shows typical corruption of time series: datapoints of the original (in the middle) can be dropped (above) or duplicated (below); right display shows correposning change of time variable.

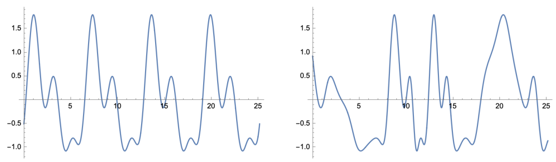

Figure 2: Left (periodic) and right functions are related by a reparameterization of the domain.

ties. This project aims at developing and deploying computational tools and practices of time series analysis that are explicitly reparameterization invariant (RI).

Formally, we say that a functional $\mathcal{I}$ of a real-valued univariate function $f : I \to \mathbb{R}, I = [0, T]$ is reparameterization invariant if for any monotonic coordinate change $s : I \to I$, $\mathcal{I}(f) = \mathcal{I}(f \circ s)$.

Classically, the RI functionals (up to *detours*, i.e., segments of trajectories that are exactly backtraced) with values in a space of dimension $d \geqslant 2$ are characterized by the iterated integrals [5]. This realization lead to several research thrusts, one known as the *signatures* [7], the other as *cyclicity analysis*, [1,4].

These approaches manifestly fail when the time series is *one-dimensional*, where any trajectory is just a collections of detours.

There is, nonetheless, a complete descriptor of the continuous functions $f : I \to \mathbb{R}$ which is reparameterization invariant. This descriptor is, again, classical, and is given by the excursion-to-trees correspondences.

Recall that an *excursion* is a continuous function $f : I \to \mathbb{R}$ attaining its global minimum exactly at the boundary points of the interval I. To such a function one can assign a *continuous tree*, a rooted planar tree[1] $\mathcal{T}(f)$ with a length function assigned to each edge. The vertices of the tree $\mathcal{T}(f)$ correspond to the local extrema of f, and the distance to the root is equal to the values of f at those extrema.

These planar rooted metric trees (PRMTs) are the key personae dramatis of this project.

The inverse transformation, from a planar rooted tree to an excursion, is known as *height contours*, or *Dyck walk*. The correspondence $\{\text{excursions}\} \leftrightarrow \{\text{planar rooted trees}\}$ was widely used to both study the random trees, and random walks (see, e.g., [11]). We notice that in general, the trees corresponding to a continuous function are rather wild (e.g., can have vertices with countably many offsprings), but generically are binary (still, with countably many vertices).

One can easily see that two excursions $f : I \to \mathbb{R}$ and $f' : I' \to \mathbb{R}$ have equal trees, if and only if one is a reparameterization equivalent to the other: $\mathcal{T}(f) = \mathcal{T}(f') \Leftrightarrow$ there exists a continuous increasing $s : I' \to I : f' = f \circ s$.

In other words, these trees form a complete invariants of the one-dimensional time series, up to reparameterizations. (While we focussed here on the excursions, any function can by turned into one, by adding increasing/decreasing segments at the either end of the function/time series.)

The use of these trees as descriptors of random trajectories was investigate (in the context of topological data analysis, persistent homology and *merge trees* [6]) in several papers, in particular by the proposer [3].

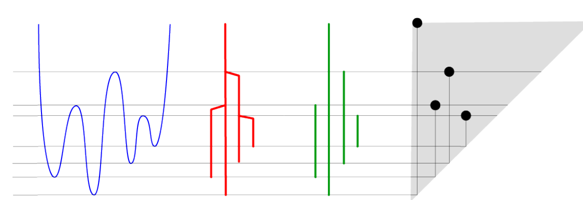

Figure 3: Correspondence: function $\to$ merge tree $\to$ bars $\to$ persistence diagram.

Our project aims to adopt the trees $\mathcal{T}(f)$ as descriptors of the continuous function f, and apply this idea to the analysis and prediction of the time series, com-

[1] A tree together with an embedding into a plane, or, equivalently, with an order of offspring for any non-leaf vertex.

bining them with the computational pipe of Large Language Models.

3 Tokenization

The key innovation proposed here is the RI approach to tokenization of the time series.

As we mentioned, the tree $\mathcal{T}(f)$ is a complete RI descriptor of the time series. However, to effectively design a language model, one needs to set up a discretization procedure, encoding a long time series into a sequence of symbols of a finite alphabet. Turning a set of continuous objects (like time series) into subset is a necessary step. In our approach we plan to deploy the tools of *persistent homology*.

In our context, persistent homology splits the critical points of a function (maxima and minima, in the case of time series) into pairs: the pairs with the small difference between the critical values are considered noise, those with high difference, a signal. The scale (what is considered low and what high) is chosen adaptively, depending on the context.

One byproduct of this view is the efficient algorithmic procedure of *trimming a PRMT* removing all the branches of small length (below a sensitivity threshold σ_*). The result of this procedure is a tree having the same larger-scale features, but cleaned of the noisy small branches.

One can immediately see that if one upper-bounds their total length (the size threshold δ_*), the number of combinatorially different PRMTs becomes finite. This finite collection can then be used as a potential alphabet for the tokenization procedure.

The size of the alphabet (token space) depends on the upper bound δ_* of the tree length, and the trimming scale σ_*. It is easy to see that it grows exponentially in δ_*/σ_*, so a careful balance should be maintained here.

3.1 Token Generation

Using these heuristics as guidance, one can propose a class of procedures of partitioning a continuous function/time series into a sequence of PRMTs. Namely, one each step has as an input the current marker position t_k, and considers the restriction of the time series to the interval $[t_k, T]$. An online (one-pass) algorithm can be deployed to generate the corresponding growing family of PRMTs, and strip it off the branches below the predetermined scale σ_* (see Figure fig:tokens).

Once the PRMT total length exceeds the threshold δ_*, it is output as the next token, and the rightmost local minimum in the covered timespan is assigned as t_{k+1}.

This iterative procedure would effectively render a time series as a sequence of tokens form a given alphabet.

There are many technical details of the tokenization that we are unable to spell out here, for lack of space, involving, inter alia, dual trees and single-pass property of the algorithms.

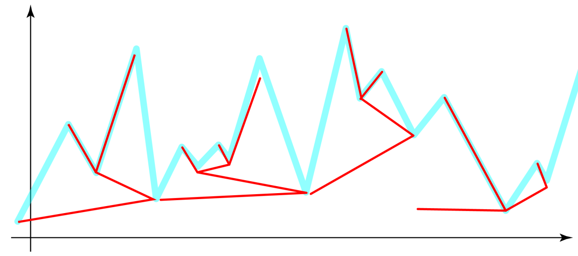

Figure 4: Several steps of token generations shown.

4 Plan of Works

The overall project aims at creation of a novel Reparameterization Invariant autoregressive model for time series.

4.1 Tokenization Module

As we mentioned, the key innovation is in the RI tokenization procedure for the time series, sketched above. Therefore, most of the effort in the first year of the project will be on implementing it, addressing (inevitable) challenges along the way, from questions about discretization procedures for PRMTs, to the selection of the parameters δ and σ for the denoising and finitization.

As a result we expect a suite of algorithms, implemented as python packages, which would select tokenize time series effectively, with appropriately chosen token classes.

4.2 LLM Prototype

Once the RI tokenization is under control, we intend to design a prototype LLM. As mentioned, the key innovation of this project is in the RI tokenization, so the (by now) routing architectures will be used, such as transformers, relying on the standard software tools, such as `TensorFlow/Keras`. We postpone a detailed discussion of the implementation of this step till later.

4.3 Training Data Acquisition

It is not clear whether one can hope for a zero-shot LLM at the rather modest scale of this computational prototype design. As the initial datasets, we plan to employ some of the potentially medically important traces, like the intracranial EEG datasets (see `https://www.ieeg.org/`), with the focus on the detecting preictal events.

As the work progresses, we plan to expand the data collections used for the model training, first to other neurophysiological datasets, then to general collections, involving financial, climatic, geological data.

References

[1] I. Abraham, S. Shahsavarani, B. Zimmerman, F. Husain, and Y. Baryshnikov, *Slow Cortical Waves through Cyclicity Analysis*, Network Neuroscience, (2024).

[2] A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor, J. Zschiegner, D. C. Maddix, M. W. Mahoney, K. Torkkola, A. G. Wilson, M. Bohlke-Schneider, and Y. Wang, *Chronos: Learning the Language of Time Series*, Mar. 2024. arXiv:2403.07815 [cs].

[3] Y. Baryshnikov, *Time series, persistent homology and chirality*, (2019). arXiv:1909.09846.

[4] Y. Baryshnikov and E. Schlafly, *Cyclicity in multivariate time series and applications to functional MRI data*, in 2016 IEEE 55th Conference on Decision and Control (CDC), IEEE, Dec. 2016.

[5] K.-T. Chen, *Integration of Paths–A Faithful Representation of Paths by Noncommutative Formal Power Series*, Transactions of the American Mathematical Society, 89 (1958), p. 395.

[6] H. Edelsbrunner and J. L. Harer, *Computational topology. An introduction*, Providence, RI: American Mathematical Society (AMS), 2010.

[7] G. Flint, B. Hambly, and T. Lyons, *Discretely sampled signals and the rough Hoff process*, Stochastic Processes and their Applications, 126 (2016), pp. 2593–2614.

[8] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson, *Large Language Models Are Zero-Shot Time Series Forecasters*.

[9] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, *Time-LLM: Time Series Forecasting by Reprogramming Large Language Models*, Jan. 2024. arXiv:2310.01728 [cs].

[10] M. Jin, Q. Wen, Y. Liang, C. Zhang, S. Xue, X. Wang, J. Zhang, Y. Wang, H. Chen, X. Li, S. Pan, V. S. Tseng, Y. Zheng, L. Chen, and H. Xiong, *Large Models for Time Series and Spatio-Temporal Data: A Survey and Outlook*, Oct. 2023. arXiv:2310.10196 [cs].

[11] J. Pitman, *Combinatorial Stochastic Processes: Ecole d'Eté de Probabilités de Saint-Flour XXXII - 2002*, Springer, July 2006.