Time Series: Defining a Search Engine

Philipp Beer

October 3, 2021

1 Thesis: Time Series Search Engine

1.1 Purpose

The purpose of this thesis is to explore the possibility of creating a time series series search engine.

1.2 Introduction

Time series is often described as "anything that is observed sequentially over time" which usually are observed at regular intervals of time **hyndman2014forecasting**. They can be described as collection of observations that are considered together in chronological order rather than as individual values or a multiset of values. Their representation can be described as ordered pairs: $S = (s_1, s_2, \ldots, s_n)$ where $s_n = (t_n, v_n)$. t_n can be a date, timestamp or any other element that defines order. v_1 represents the observation at that position in the time series.

Time series are utilized to analyze and gain insight from historic events/patterns with respect to the observational variable(s) and their interactions. A second area of application is forecasting. Here time series are utilized to predict the observations that occur in future under the assumption that the historic information can provide insight into the behavior of the observed variables.

Fu_2011 in their work **Fu_2011** categorized time series research into (1) representation, (2) indexing, (3) similarity measure, (4) segmentation, (5) visualization and (6) mining. Research in these different fields started taking off in the second half of the 20^{th} century. For example in **_str_m_1969** the authors worked on questions of representation via sampling the sampling of time series in **_str_m_1969**. All these different research areas always have to deal with the challenges that inhibit time series data. Generally datasets in this domain are large. Through this time series data incorporates the similar obstacles as high dimensional data, namely the "curse of dimensionality" **Tang_2019** and requiring large computational efforts in order to generate insights. And as will be discussed in 1.2.1 there are applications fields where vast amounts of time series are generated and a comparison between them is required.

In this thesis we will focus on creating a algorithm allowing the fast and meaningful comparison of an input time series or template against a vast array time series. Within the research many different areas and approaches have been attempted (at list here). However, there is a tendency to apply the simplest methods possible to achieve the desired results. For time series those are mainly Euclidean Distance and Dynamic Time Warping. While those methods will be explored in section 1.3 it can be said that those methods are simple, easy to understand and produce mostly reliable results in their respective application domains. Good performance is not their strong suit. Therefore, our approach

is targeted to achieving comparable capability in identifying similar series, while achieving it with a significant reduction in computational complexity.

1.2.1 Applications

Time series are encountered everywhere. Any metric that is captured over time can utilized as time series. Granularity can be used as descriptor for the sampling rate of a series or more general how often measurements for a particular metric are taken. This granularity has a tendency to increase as well. As example consumer electronics that capture health and fitness data can mentioned. Or sensors which are utilized in the automotive industry or heavy machinery where they are employed to capture information for predictive maintenance applications.

In the financial industry time series are a very fundamental component of decision making, like the development of stock prices over time or financial metrics of interest. The same is true for macro economic information or metrics concerning social structures in society, etc.

In the medical field time series are also ubiquitous. Whether they relate to patient date like blood pressure. The bio statistics field utilizes electro-graphical data like electrocardiography, electroencephalography and many others. In more aggregate medical analysis like toxicology analysis of drug treatments for vaccine approvals they are utilized and in many forms of risk management, for example, population level obesity levels.

In engineering fields the utilization is often times similar to the above but it also require that information that is captured in time series is transferred between locations in an efficient manner. For example voice calls are required to be transferred between the participants in a fast manner and with minimized levels of noise in the data. Another interesting industrial example in the biomedical technology field is Neuralink which aims to implement a brain-machine-interface (BMI) utilizing mobile hardware like smartphones as the basis for its computation. Here a large of amount of time series data is generated which requires quick processing to generate real-time information. **Musk_2019** describes a recording system with 3072 electrodes generating time series data **Musk_2019** that is used to capture the brain information and visualized in real-time **Siegle_2017**.

Time series data is paramount to a wide variety of areas, relating to many different fields. Looking at the trajectory it seems likely that going forward more time series data on a higher granularity will be generated. This in turn increases the need to be able process, analyze, compare and respond to the data with methods that are faster than today's standard options.

1.2.2 Organization of this thesis

The rest of this thesis is organized as follows:

1.2.3 TODO to be integrated

- refer to previous work on measures of similarity and outcome
- measure of similarity required
- challenges with time series (domains, granularity, length, outliers)

- area of signal processing interesting methods

1.3 Related work

Related work addressing the idea of time series search engine focuses on the system architecture and the data processing and pipelining aspect of this such an architecture **Zhang_2012**. However, in **Keogh_2000 Keogh_2000** also applied a dimensionality reduction technique (Piecewise Constant Approximation) to execute fast search similarity search in large time series databases. Other papers address domain specific questions like the introduction of a "Time-series Subimage Search Engine for archived astronomical data" **Kang_2021**.

In order to be able to describe the closeness of time series or multiple time series to each a measure for similarity is required. In the literature various general measures and corresponding computation methods can be found. **Wang_2012** reviewed time series measures and categorized the similarity measures into 4 categories: (1) lock-step measures, (2) elastic measures, (3) threshold-based measures, and (4) pattern-based measures. **Zhang_2020** classify similarity measures in the categories: (1) time-rigid methods (Euclidean Distance), (2) time-flexible measures (dynamic time-warping), (3) feature-based measures (Fourier coefficients), and (4) model-based methods (auto-regression and moving average model) **Zhang_2020**. Lock-step measures include the L_p-norms (Manhattan and Euclidean Distance) as well as Dissimilarity Measure (DISSIM). Elastic measures include metrics like Dynamic Time Warping (DTW) and edit distance based measures like Longest Common Subsequence (LCSS), Edit Sequence on Real Sequence (EDR), Swale and Edit Distance with Real Penalty. An example for threshold-based measures are threshold query based similarity search (TQuEST). And Spatial Assembling Distance (SpADe) is an example for pattern-based measures. In another paper, **Gharghabi_2020** classify the space of similarity measures by the the most common measures into: (1) Euclidean Distance, (2) Dynamic Time Warping (DTW), (3) Least Common Subsequence (LCSS), and (4) K-Shape.

1.4 Similarity metrics

1.4.1 Euclidean Distance

Euclidean Distance is the most widely used distance metric in the research of time series. It is either used as a metric on its on or a as metric used used inside other methods to compute distances, for example, computation of distance of subsections of the data (**Faloutsos_1994**) or to compute the distance between various points of two time series (see section 1.4.2). Having two time series $S = \{s_1, s_2, \ldots, s_n\}$ and $Q = \{q_1, q_2, \ldots, q_n\}$ both of length n the Euclidean distance can be computed as:

$$D(S,Q) = \sqrt{\sum_{i=1}^{n} (S_i, Q_i)^2} \tag{1}$$

It is a measure that is easy to compute and comprehend and gives intuitive input for the distance of two time series. From the standpoint of time complexity the algorithm is applicable also to larger datasets with $\mathcal{O}(n)$. Its simplicity also creates some limitations. For example, to compute the euclidean distance between two series their length needs to be the same. Furthermore, it can be easily impacted in its results by the presence of outliers or increased levels of noise. It is not elastic with respect to the warping of information between two series in which effects that could indicate similarity happen even at slightly disparate steps.

Despite its shortcomings it is a prominent metric and widely used for distance calculations for short comings. Some of its limitations are addressed by more sophisticated metrics that utilize its computation as component.

1.4.2 Dynamic Time Warping

Berndt94usingdynamic introduced Dynamic Time Warping in **Berndt94usingdynamic** finding the minimal alignment between two time series computed through a cost matrix and identifying the minimized path through the matrix starting from the final elements of each time series. This warps the points in time between the different series as shown in figure 1.

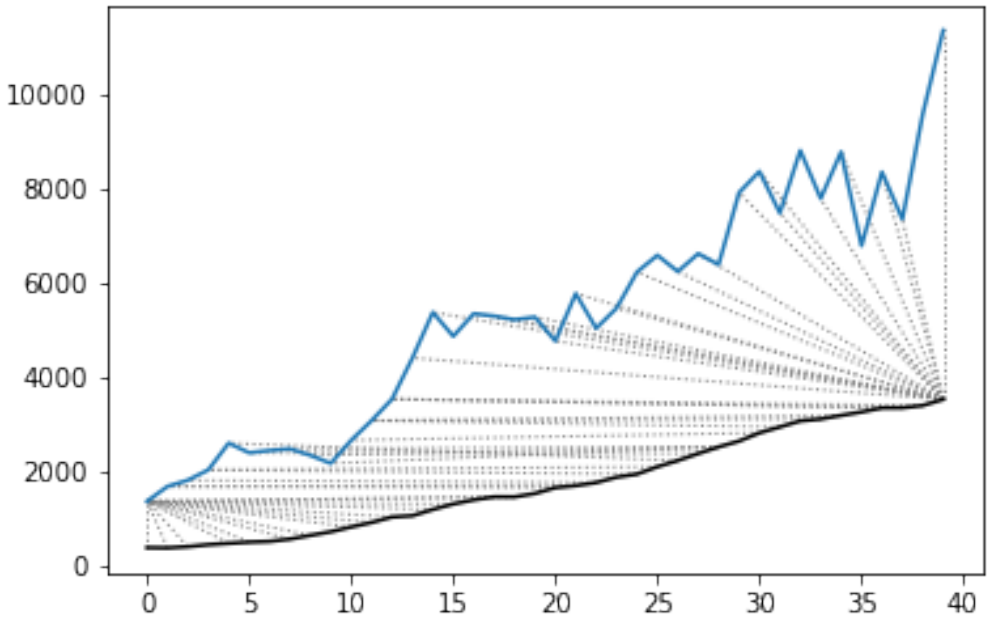

Figure 1: Dynamic Time Warping - M4 Example: Y5683 and Y5376

Two series $S = \{s_1, s_2, \ldots, s_n\}$ of length n and $Q = \{q_1, q_2, \ldots, q_m\}$ of length m are considered. For the series a n-by-m cost matrix M is constructed. Each element in the matrix represents the respective i$^{\text{th}}$ and j$^{\text{th}}$ element of each of the two series which contains the distance of those to points:

$$m_{ij} = D(s_i, q_j) \tag{2}$$

where often time euclidean distance is used as distance function $D(s_i, q_j) = (s_i - q_j)^2$. From the matrix a warping path P is chosen, $P = p_1, p_2, \ldots, p_k, \ldots, p_K$ where:

$$max(m, n) \leq k < m + n - 1 \tag{3}$$

The warping path is constrained with bound with the following condition $p_1 = (1, 1)$ and $p_K = (m, n)$. That means that both first elements of each series, as well as, the last element of each series are bound to each other in the computation. The warping path also is continuous. This means that

from each chosen element p_k only the neighboring elements to the left, right and diagonally can be chosen for the continuation of the path: $p_k = (a, b)$ and $p_{k-1} = (a', b')$ with $a - a' <= 1$ and $b - b' <= 1$. The path elements p_k are also monotonous, meaning that $a - a' \geq 0$ and $b - b' \geq 0$. From the resulting matrix considering the mentioned constraints a cumulative distance $\gamma(i, j)$ is computed recursively:

$$\gamma(i, j) = D(s_i, q_j) + min\{\gamma(i-1, j-1, \gamma(i-1, j), \gamma(i, j-1)\} \tag{4}$$

Therefore, the path can obtained by the following definition:

$$DTW(S, Q) = min\{\sum_{k=1}^{P} \delta(\omega_k) \tag{5}$$

Figure 2 provides an example for a warping path result.

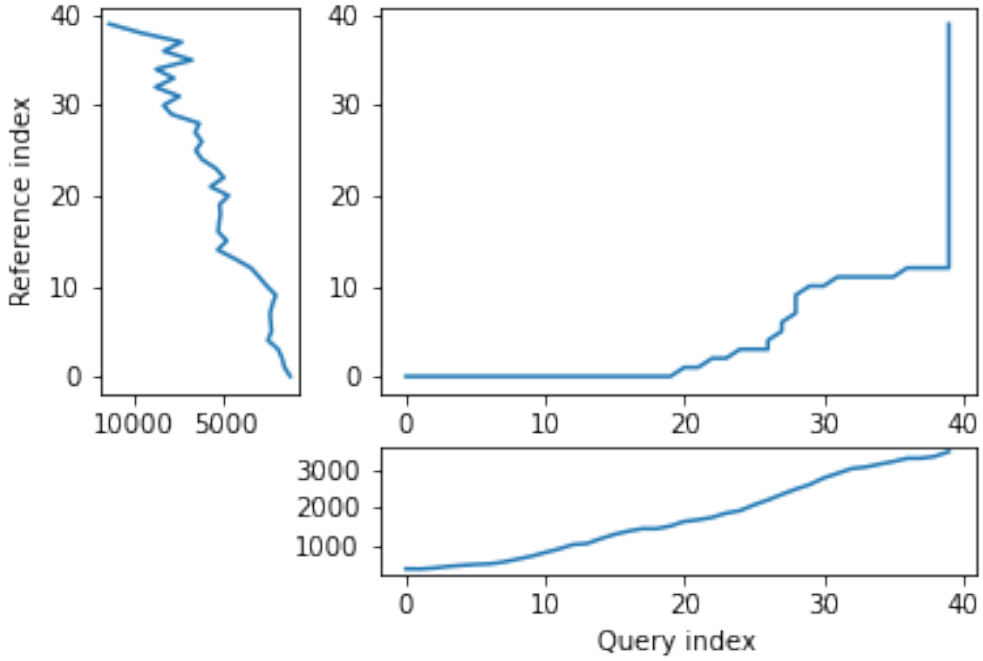

Figure 2: Warping path example - M4 data: Y5683 and Y5376

The challenge with the application of DTW is the time complexity of the algorithm $\mathcal{O}(m * n)$ due to the fact that the distance compuation needs to be executed for each element of each series. Various methods for speed improvements have been introduced. The favorite principle was described by **Ratanamahatana_2004**. They introduced an adjustment window condition that where it is assumed that the optimal path does not drift very far from the diagonal of the cost matrix **Ratanamahatana_2004**. However, this does not change the fundamental nature of the algorithm and computing DTW for multiple time series against a database of time series will require days of computation time even on modern computer architectures.

In favor of DTW needs to be stated, that it is flexible with regards to the series used. The compared time series do not require to have the same length and can still be compared. This is a property

that is not avaiable with Euclidean Distance. However, the user also needs to be aware of outliers in either data set which can lead to a clustering of the warping path or pathological matches around those extreme points in the series.

Therefore in practice, Dynamic Time Warping is not a method suitable for comparing a single time series against a large array of series when speed is an important criterion as well as the handling of outliers in the dataset.

- Similarity through decomposition
  - introduce time series decomposition (reference in **hyndman2014forecasting**)
  - trend and seasonality (mention assumptions about period)

1.5 Time series representation

1.5.1 Challenges when building a time series

- length of series
- trend
- seasonality
- time complexity -> issue because of data size
- granularity or sampling rates
- noise
- data quality
- similarity is task dependent (level)
- usual need for preprocessing the time series data (denoising, detrending, amplitude scaling) -> any pre-processing does modify the series

1.5.2 Data Analysis

what does M4 data look like

1.5.3 Challenges

- How many frequencies to compare?
- priorities of frequencies (power spectrum)
- different length of time series (leading to different frequencies) - ranges solved with logs

1.6 Methodology

1.6.1 Used Data

The research in time series has been numerous and focused on various properties of them as well as finding methods to accurately predict them. Aside of forecasting all researched areas are measures of similarity and retrieval of time series.

- Forecasting In the arena of forecasting the M-competition organized by Prof. Makridakis played a big role in the development of forecasting methods shortly after their inception in 1979.

  One of the aspects that has been correct up until the 5th installment of the M-competition is that statistical methods in forecasting have outperformed more complex machine learning methods. So learning algorithms did not benefit sufficiently from learning from multiple series to generate more accurate point predictions and prediction intervals compared to the statistics-based alternatives.

  One interesting question in this area is whether clustering of time series that have similar properties and training algorithms per cluster of "similar" series can help simplify the learning process for machine learning methods and in consequence improve their performance in future competitions.

  However, expressing similarity for time series is a challenging questions with respect to which metrics to utilize, time complexity as well as limiting assumptions that need to be made for time series.

1.6.2 Main contribution of the thesis

- transformation into Fourier-space
- transfer frequencies into frequency range band with increasing range width (using log scale)
- computation of frequency energy levels (sort and keep top 5) -> ask Prof. how to name this parameter
- conversion of ordered frequencies into frequency range band
- for each series to compare -> compare whether the frequency matches on the ordered positions -> provide exponential value per position -> match on more powerful frequencies is valued higher

1.6.3 additional computations

- utilization of FFT utilizes only frequency space (future work should consider comparison of energy levels per frequency)
- additional simple statistics computed (mean, std, quantiles)
- ts decomposition for trend estimation (requires parameter for period) -> then best line fit for slope of the time series
- computation of deltas for each series to search with statistics and slope of all other time series (review computational complexlity)
- ranking of matching series based highest frequency range match and ONE statistic

1.6.4 Preprocessing

- M4 data wide format vs. long format

1.6.5 Parallelization

- computation times
- scalability
- Samples for results only (stratification vs. non-stratification)
- Threads vs. Processes

1.6.6 Technology (check with Prof. if required)

R vs. Python vs. Mathematica, Matlab

1.6.7

- load
- transform to FFT vector space
- compare most important frequencies
- compare candidates
- select winner (which criteria)

1.7 Exploratory Data Study

- what do results look like

1.8 Formal Evaluation

- (maybe) improvement in forecasting approach
- find dataset with ground truth and compare DTW to this approach
- Distance metrics
- time complexity

1.9 Conclusion & future work

1.9.1 Successes

1.9.2 Failures

1.9.3 Flaws

- final computation

1.9.4 What is missing

- denoising of time series
- adjustment of number of frequencies used
-

1.10 Results & Discussion

1.11 References