7CCSMRPJ

Directional Change Intrinsinc Time Framework as Target Transformation in Time Series Modelling

Final Project Report

Yu-Kuan, Lin

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Acknowledgements

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim ad minima veniam, quis nostrum exercitationem ullam corporis suscipit laboriosam, nisi ut aliquid ex ea commodi consequatur? Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae consequatur, vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?

1.1 Aims and Objectives

something about chapter one, section 1

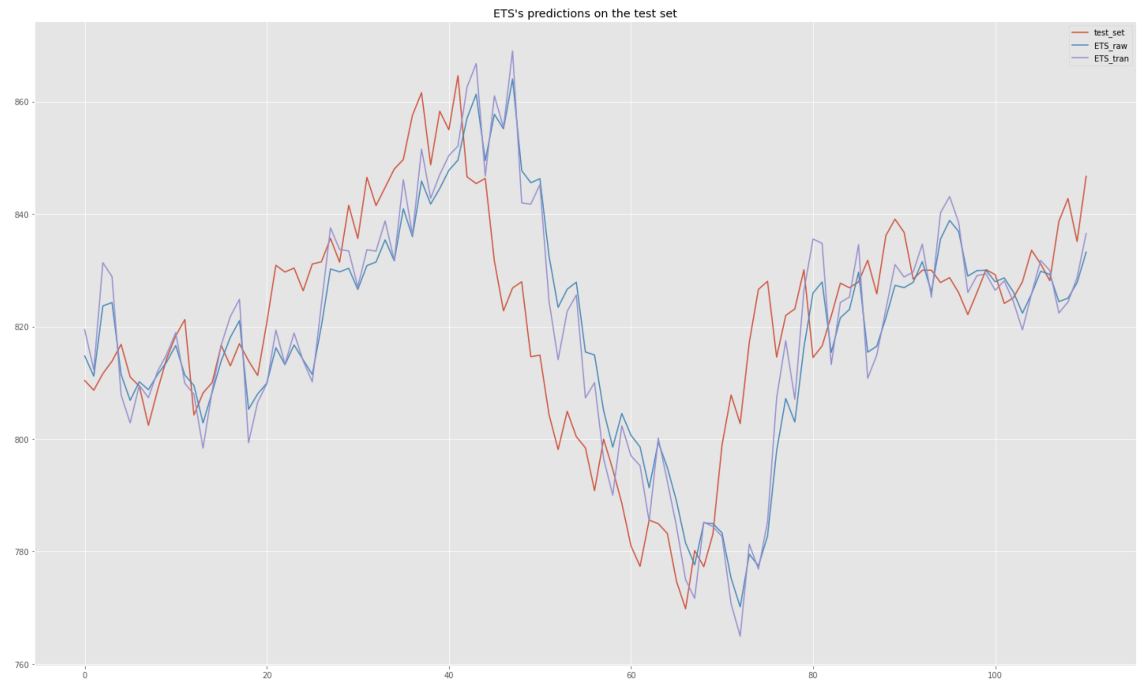

Figure 1.1: ETS's prediction on the test set

1.2 Problem Statement

something

1.3 Movination

something

1.4 Report Structure

something

Chapter 2

Background

This chapter presents a comprehensive theoretical background to the topics related to our experiment and analyses.

2.1 Univariate Time Series Forecasting

In this section, we discuss the topic of univariate time series forecasting.

2.1.1 General Notions of Time Series Forecasting

A time series is a sequential collection of random variables indexed by time. Suppose the random variables are of one dimension. The time series is considered univariate[1]. In this section, the discussions fall within the context of univariate time series forecasting. Let $Y = \{Y_{t_i}\}_{i=1,2,\cdots,n}$ be an univariate time series with $Y_{t_i} \in \mathbb{R},\ \forall i,\ n \in \mathbb{N}$ and $t = \{t_i\}_{i=1,2,\cdots,n}$ being the set of time indices of Y (also referred to as the set of timestamps). Let $\mathcal{Y}_{t_k}$ be the largest information set about Y that is accessible to the model at time point $t_k \in t$, e.g., we might have the observations of Y being known ($\{Y_{t_j} = y_{t_j}\}_{j=1,2,\cdots,k} \subset \mathcal{Y}_{t_k}$). Then in the context of modelling, for an unknown (random) target Y_{t_s} with $s > k,\ s \in \mathbb{N}$, we can articulate the notion of forecasting as the following:

> Forecasting the value Y_{t_s} at time point t_k is to find a function $f(\mathcal{Y}_{t_k}) = \widehat{Y_{t_s}}$, such that $\widehat{Y_{t_s}}$ is a good estimation of Y_{t_s}.

[1] If the random variables are of dimension higher than one, then the time series is considered multivariate. Datasets in such form are also referred to as panel data.

We can develop all sorts of functions f for such forecasting objectives. Functions devised to serve the objective are referred to as models. In a general sense, the notion of modelling refers to the methodologies of devising a model that serves the objective well. In particular, for any arbitrary pair of t_k and t_s, we want to have a model $f(\mathcal{Y}_{t_k})$ such that it gives us a reliable estimation of Y_{t_s}.

Gap

Notice that a modelling objective is parameterised by the pair of time indices (t_k, t_s). The timestamp t_k directly affects the information set $\mathcal{Y}_{t_k}$ and thus determines the information the model f can utilise. The timestamp t_s, on the other hand, controls how far in the future we are forecasting. If the gap between the two timestamps is big, the model is asked to forecast further into the future. If the gap between the two timestamps is small, then the objective might be considered easier because we are only trying to look a tiny step ahead into the future. In order to better communicate and characterise the forecasting objective, we formalise this gap between the pair of timestamps as the *gap* and denote it as τ:

$$\tau = t_s - t_k$$

With such a notion of the gap, we can then articulate the forecasting objective as τ ahead forecasting. τ takes the format of the timestamps. Depending on the format of the timestamps, the objective can be one-step ahead forecasting or one-year ahead forecasting. We will go into topics concerning timestamps in one of the upcoming paragraphs.

Forecast Horizon

Observe that the target we have in the previous forecasting objective Y_{t_s} is a single value in the future. It is possible to generalise the target and have multiple targets in the future. The number of targets we try to forecast is called the *forecast horizon*. Having a forecast horizon equal to one is to forecast one value into the future, and having a forecast horizon equal to five is to forecast all five values into the future. To put it formally, let $\langle\cdot\rangle$ be a counting operation that counts the number of elements for a finite set, and let a task have a finite collection of unknown values $Y_S = \{Y_{s_1}, Y_{s_2}, \cdots\}_{s_1,s_2,\cdots\in t,\, s_1,s_2,\cdots>t_k}$. Then the forecast horizon of such a task is denoted as

$$H = \langle Y_S \rangle$$

Timestamps

In a time series, the time index of a random variable carries information about the time point in which the random variable lives in the time domain. In some sense, the timestamps mark the 'location' of the random variables on a timeline. For example, a monthly revenue time series Y in an arbitrary year can have the set of months in a year as its timestamp set and be denoted as $Y = \{Y_{Jan}, Y_{Feb}, Y_{Mar}, \cdots, Y_{Dec}\}$. The time indices also tell us the time-relevance (geological relationship on a timeline) of the random variables among each other. In fact, the time-relevance of the random variables in a time series plays a crucial role in time series analysis. We often have to perform mathematical operations involving such relationships. An example is our coming up with the gap measurement we addressed in the previous paragraph. Another example is the calculation of the relative growth of the time series. The need for these math operations pushes modellers to devise innovative ways to define the timestamps because we cannot easily perform calculations on notations like *September* or $Friday$. To put it in math terms, what we often do is to have a mapping from physical timestamps to the real number line (or a subset of the real number set, say, the natural number set) and use the target set of this mapping as the timestamp set for math operations. In the next paragraph, we discuss some examples of such mapping.

Take the previous monthly revenue time series as an example; one simplest way is to index the time series chronologically with natural numbers $\{1, 2, 3, \cdots, 12\}$. The new index system allows for mathematical operations on the timestamps, such as addition. The objective of two-month ahead forecasting can be considered as two-step ahead forecasting with $\tau = 2$. Three-month moving average of the time series can now be of a generalised form of a three-step moving average. Another good example is the financial studies of stochastic processes, in which we often use the non-negative real line and adopt an annual scale, i.e., the starting point of the time series is indexed as 0, one month after that is indexed 0.0833, one-year time point is indexed 1.0, and so on. This is particularly useful when we expand the time series studies using Stochastic Differential Equations (SDE). Let the SDE of a stochastic process dY_t be given as

$$\frac{dY_t}{Y_t} = \mu_t dt + \sigma_t dW_t, \quad dW_t \sim N(0, dt).$$

dt in the drift term is now properly defined as a real number on which we can do all sorts of math operations (observe how dW_t is defined as a Brownian Motion that follows a Gaussian distribution with variance dt).

Time Heterogeneity

There are cases where the timestamps of the time series are not identically spaced between consecutive random variables. The previous example of monthly revenue in a year is one of them due to the months having different durations. Time series as such is technically referred to as being *time heterogeneous*[2][3]. Time heterogeneity can be an interesting source of information carried by the time series but can also be a major issue in time series studies. The following paragraph presents a common problem caused by time heterogeneity.

Calculating measurements related to the unit of time can be a problem with time heterogeneous time series. One common example of such measurement is the return used in finance. Return measures the relative change of the price (or, say, value) over a period of time with respect to its initial level. Several definitions can be drawn to the notion of return, but we will look at the simplest one as they all exhibit the same relationship with time heterogeneity. Let $Y = \{Y_{t_i}\}_{i=1,2,\cdots n},\ n \in \mathbb{N}$ now be a time series of the price movement of an asset over time, and the time index set being $t = \{t_i\}_{i=1,2,\cdots,n}$. Define the corresponding net return measure as

$$R_{t_i} = \frac{Y_{t_i} - Y_{t_{i-1}}}{Y_{t_{i-1}}}, \quad i \in \{2, 3, \cdots, n\}.$$

The net return R_{t_i} for some i is thus the relative price change of Y_{t_i} over the time period $t_i - t_{i-1}$ with respect to $Y_{t_{i-1}}$. Let $R = \{R_{t_i}\}_{i=2,3,\cdots,n}$ be the time series of one-step net returns derived from Y. If t is equally spaced by, say, a day, the time series Y is time-homogeneous. The time series R we calculated is a time series of daily net return of Y. Nevertheless, in the case where t is not equally spaced, the time series Y is time heterogeneous. Then we no longer know the period of the net returns R we calculated from the time series Y. This is an example of how time heterogeneity can complicate time series analyses. Time heterogeneity in finance has been studied a lot, especially with the popularisation of electronic systems. See Dacarogna et al. (2001) for more information.

[2]Time heterogeneity is common in financial time series due to the nature of how financial markets work. For example, most markets are open only during working hours on working days. Another example is the high-frequency financial time series (see Dacarogna et al. 2001)

[3]*Time homogeneity* is the counterpart of time heterogeneity, specifying time series which have equally spaced timestamps

2.1.2 Machine Learning Regression Modelling

In this section, we further discuss univariate time series modelling and address how we approach the articulated forecasting objective with Machine Learning (ML) regression modelling. In addition to addressing the approach, we also provide relevant statistical notions that can be seen as the underlying theoretical foundation of the standard machine learning procedure.

Recall the forecasting objective is to come up with a model f capable of generating 'good' estimations of some target Y_{t_s} at time t_k using the provided information $\mathcal{Y}_{t_k}$. We will see in the later paragraphs that this is very similar to the process of solving for a Quasi-Maximum Likelihood Estimation (QMLE) in a statistical sense (White (1982)). In the context of modelling, the procedure is normally to gather the available information and formulate it into an optimisation problem: we define a fitness measure (like the 'L' in QMLE), which should be a function of the estimates $f(\mathcal{Y}_{t_k})$ and the target, and we then optimise the fitness as an objective function with respect to the model f in some algorithmic way (the 'M' in QMLE). The final output of such a procedure will be a model (function f) that optimally serves our objective (the 'E' in QMLE). In the remainder of this section, we discuss the general framework of how the procedure works.

The Design Matrix and Target

In regression problems, the design matrix and the target are made from accessible information and formulated for a modelling environment. In this paragraph, we discuss how to develop the design matrix and target in univariate time series modelling. Recall that the forecasting objective is to estimate Y_{t_s} at time point t_k, with the gap $\tau = t_s - t_k > 0$ and the model is only able to utilise the accessible information set $\mathcal{Y}_{t_k}$. This dataset is called the *training set*. For our modelling problem, we can only use what is provided in our training set $\mathcal{Y}_{t_k}$. The idea is to create a sandbox in which we simulate the modelling process. The design matrix and the target create this sandbox (environment). Without loss of generality, for the target Y_{t_s}, the information set accessible is $\mathcal{Y}_{t_s-\tau}$. In the event we have a one dimensional time series, $\mathcal{Y}_{t_s-\tau}$ is simply the collection of realised values of Y until time point $t_s - \tau = t_k$, namely

$$\mathcal{Y}_{t_s-\tau} = \{Y_{t_s-\tau} = y_{t_s-\tau}, Y_{t_{s-1}-\tau} = y_{t_{s-1}-\tau}, Y_{t_{s-2}-\tau} = y_{t_{s-2}-\tau}, \cdots, Y_{t_1} = y_{t_1}\}.$$

With this long list of past observations, there is a parameter we have to decide, which controls the number of past observations the model will use for a single prediction. Such parameter is called the *number of lags*[4]. Let the number of lag be denoted as $\lambda,\ \lambda \in \mathbb{N},\ \lambda \ll k$. Then the values we allow the model to use for some target Y_{t_i} is

$$\mathcal{Y}_{t_i-\tau} = \{y_{t_i-\tau}, y_{t_{i-1}-\tau}, y_{t_{i-2}-\tau}, \cdots, y_{t_{i-\lambda}-\tau}\}.$$

Then the design matrix $\mathbf{X}$ and the corresponding $\mathbf{y}$ can be formulated as

$$\mathbf{y} = \begin{bmatrix} y_{t_\lambda} \\ y_{t_{\lambda+1}} \\ \cdot \\ \cdot \\ y_{t_{k-1}} \\ y_{t_k} \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} y_{t_{\lambda-1}} & y_{t_{\lambda-2}} & \cdots & y_{t_1} \\ y_{t_\lambda} & y_{t_{\lambda-1}} & \cdots & y_{t_2} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ y_{t_{k-2}} & y_{t_{k-3}} & \cdots & y_{t_{k-\lambda}} \\ y_{t_{k-1}} & y_{t_{k-2}} & \cdots & y_{t_{k-\lambda+1}} \end{bmatrix}.$$

In this sandbox, the model has $k - \lambda + 1$ prediction to make, each being using a row vector in $\mathbf{X}$, denoted as $\mathbf{X}_i,\ i \in \{1, 2, \cdots, k - \lambda + 1\}$ and estimate the corresponding a row element in $\mathbf{y}$, denoted as $\mathbf{y}_i$. The idea is to find a model that best maps the rows in $\mathbf{X}$ to elements in $\mathbf{y}$ in general, and we say this is our best model f that serves the objective of a τ ahead forecasting task given the time series we have.

The Model

We start by describing what a model is in more detail. Given the design matrix $\mathbf{X}$ and target $\mathbf{y}$ we made in the previous section, the forecasting objective is now transformed into coming up with a model f that best maps the rows in $\mathbf{X}$ to the corresponding element in $\mathbf{y}$.

Consider f as an arbitrary machine learning regression model with a known structure. Knowing the structure of f implies we know the structure of its parameter set and how f maps $\mathbf{X}_i$ to $\mathbf{y}_i$ for some i. Let θ be the mapping that generates the parameters that go into f. The output of θ is parameterised by its input set; let it be ϕ. Parameter set like ϕ is called the *hyperparameters* - it characterises the parameters

[4] Its naming originates in autoregressive estimation in time series studies. Such estimation aims at finding the optimal order of the autoregressive feature of a time series. Typically, such order is referred to as the *lag*.

of f. The model f in our forecasting objective can thus be noted as

$$f(\theta(\phi); \mathbf{X}) = \widehat{\mathbf{y}} \sim \mathbf{y}. \tag{2.1}$$

To better measure the performance of such an estimation, we define a fitness function $\mathcal{E}$. The fitness function takes the estimations generated by f and returns a real number signifying how well they fit the target. We can finally formulate our modelling objective as

$$\arg_{\theta(\phi)} \max \mathcal{E}(f(\theta(\phi); \mathbf{X}), \mathbf{y}). \tag{2.2}$$

This process is what we call *training the model.*

To give an example, if f is a simple linear regression model, then we know θ gives a tuple of real numbers (known as weights) which the model uses to generate a linear combination of its inputs; in this case, a row matrix $\mathbf{X}_i$. Specifications of θ are then controlled by its input ϕ, e.g., whether there is an intercept term or the number of elements of the tuple. Note that the number of elements in the tuple depends on the number of lags λ we adopted in making the design matrix $\mathbf{X}$, i.e., $\lambda \in \phi$. Not knowing the exact values of θ (and certainly ϕ as well) describes the state of the model f being untrained. Then the modelling objective is to find ϕ and θ such that $\mathcal{E}(f(\theta(\phi); \mathbf{X}), \mathbf{y})$ is maximised.

The Statistical Resemblance

Analogously, training a machine learning model can be put into statistical terms. In particular, it resembles the Quasi-Maximum Likelihood Estimation (QMLE) process with some minor tweaks. Consider our objective with Y, but with the elements following an arbitrary distribution $\mathcal{D}$ characterised by θ, i.e., $Y_{t_i} \sim \mathcal{D}(\theta),\ \forall\ t_i \in t$[5]. Let $g_{Y_{t_s}|\mathcal{Y}_{t_k}}(\cdot; \theta)$ be the conditional joint probability density function (pdf) of the random variable Y_{t_s} given $\mathcal{Y}_{t_k}$ and θ. Then the expression

$$g_{Y_{t_s}|\mathcal{Y}_{t_k}}(y_{t_s}|y_{t_k}; \theta)$$

describes the probability of observing $Y_{t_s} = y_{t_s}$ given the past observations and parameter θ. Then the objective of QMLE is to find the θ conditional on observing $\mathcal{Y}_{t_k}$, under which y_{t_s} is most likely to be observed. We can formalise such objectives

[5]The 'Quasi-' simply means we do not know whether the distribution $\mathcal{D}$ is Gaussian or not (see White 1982).

as

$$\arg_\theta \max g_{Y_{t_s}|\mathcal{Y}_{t_k}}(\theta; y_{t_s}|y_{t_k}).$$

The objective function that is to be optimised is called the *likelihood function*, denoted as $\mathcal{L}(\theta)$. $\mathcal{L}(\theta)$ is essentially still a pdf. It returns the probability of observing y_{t_s} conditional on y_{t_k} with the parameter θ. Maximising such likelihood function with respect to θ is thus equivalent to finding a θ dictating the generating mechanism of the random variable Y_{t_s} such that it is most likely to be observed as y_{t_s}.

Training a machine learning model bears some resemblance to QMLE. Training a model aims to find a mechanism to reproduce the target with the observations. At the same time, QMLE tries to find the set of parameters characterising the distribution of the target variable conditional on its past realisations. The outcome of training an ML model is to have a deterministic function that serves the purpose of forecasting. On the other hand, QMLE yields a set of parameters dictating the underlying distribution of the target variables, i.e., you do not end up having a recipe for making deterministic values but a probabilistic distribution. Both methodologies tackle the problem with an optimisation framework involving an objective function: QMLE utilises the probability density function (pdf) that comes with a random variable with a known distribution. ML modelling devises a fitness function to evaluate the goodness of the estimation. We hope this section contributes to a better theoretical understanding of ML modelling in terms of statistical analysis.

2.2 Target Transformation in Univariate Time Series Forecasting

2.3 Directional Change Intrinsic Time Framework

Chapter 3

Literature Review

Target transformation techniques have been researched as part of the modelling process over the past sixty years. The same applies to the analyses of the underlying mechanisms in financial dynamics. This paper builds on both research directions and analyses the Directional Change (DC) intrinsic time framework as a target transformation technique in time series forecasting. To the best of our knowledge, as this paper was written, we are unaware of any other attempt to bring these two methodologies together. In this chapter, we cover the most relevant works in both realms. We first review some target transformation developments in Section 3.1 and then look at some of the advancements in the Directional Change intrinsic time framework in Section 3.2.

3.1 Target Transformation

In most cases, time series analyses yielded from modelling are justified conditional on the assumptions made by the models about the data. If the assumptions are not met by the original data, applying some transformation (often non-linear) to the data can help generate these conditions. This has led to various transformation techniques for these types of purposes. In this section, we go through some important assumptions made by the models and the corresponding transformations found in the literature that we find most worthy of mentioning.

Homoscedasticity condition[1] is one of the most common assumptions for models

[1]The homoscedasticity (homogeneity of variances) condition requires the variances of different subsets of the sample to be the same. In the case of time series modelling, it is equivalent to

involving statistical inferences. As a reference to the analysis of variance, M. S. Bartlett (1947) provided one of the earliest summaries of transformations on raw data addressing this. He covered parametric transformations used in stabilising the variance of modelling error, especially for Poisson and Binomial distributed variables where the variance is a known function of the mean. He discussed, both theoretical and empirical, some of the optimal scales and families of transformations to choose from given different circumstances. His work showed that modelling tasks do benefit from suitable transformations.

Another common assumption made by the models is normality. Almost all statistical inferences assume the variable of interest is normally distributed, e.g., t-test, Analysis of Variance (ANOVA), linear regression etc. Therefore, it is of great interest if we have the opportunity to create such conditions for the models to operate under. Box & Cox (1964) made a major contribution in this regard by proposing the well-known Box-Cox transformation. The Box-Cox transformation includes both power and logarithmic transformations. It aims at achieving normality of the observations and has been popular in developing modelling methodologies ever since. A good example is a method proposed in C. Bergmeier et al. (2016). Combined with the widespread exponential smoothing (ETS) forecasting method and the bootstrap aggregation (bagging) technique in machine learning, they proposed a bagging exponential smoothing method using STL decomposition and Box-Cox transformation. The proposed method was tested on the M3 competition dataset and achieved better results than all the original M3 participants (see Makridakis et al. 2000 for the M3 competition).

The method published by Box & Cox, however, is only valid for positive real values. Modifications of the Box-Cox transformation have thus been proposed to address the problem. A major one is made by Bickel & Doksum (1981). They embedded a sign function to the power transformation such that the transformation function covers the whole real line. Nevertheless, this modification has its shortcomings: it is shown by Yeo & Johnson (2000) that Bickel & Doksum's modified version of the transformation handles skewed distribution poorly. Yeo & Johnson pointed out the reason being the signed power transformation was designed primarily for handling kurtosis, thus losing its edge concerning skewness. Following up, Yeo & Johnson proposed a new version of the power transformation in the same publication (2000). Their transformation is a generalised version of the Box-Cox transformation

requiring a constant variance throughout time.

and approximates normality while being well-defined on the real line and inducing appropriate symmetricity.

Until now, we looked at how a mathematician or statistician would apply response transformation techniques to foster mathematical and statistical conditions assumed by the models. Meeting these assumptions improves the robustness of the conclusions drawn by the modelling results. In a general sense, one can say that the transformations help the models to get better at 'learning' the problem such that they generate more robust outputs. In the upcoming paragraphs, we take on this perspective of treating the transformation techniques as helpers in terms of the learning process of the models and look at some transformation techniques very different from what was covered previously.

Decomposition methods constitute a significant category of techniques that can be considered target transformation in helping the models learn when applied to the target variable. These methods decompose the mixture of information contained within the observation into patterns, trends, cycles, or other dynamics that are easier to model, i.e., easier for models to learn from. Also referred to as spectrum analysis, a particularly good example would be the giant branch of studies on Fourier-styled transformations in time series modelling (see Kay & Marple 1981 and Bloomfield 2004). Typical Fourier methodologies build on transforming the observations sampled from the time domain into the frequency domain and decomposing them into more informative signals. Out of the great history and advancements of spectrum analysis, we selectively provide a very brief overview of some relevant methodologies in the context of target transformation.

A novel type of method builds on what Fourier methods do and operates in the time-frequency domain, i.e., these methods come up with time-frequency representations of the observations. The empirical mode decomposition (EMD) proposed by Huang et al. (1998) is one of the key methodologies as such. EMD decomposes the original time series into 'intrinsic mode functions' (IMF). The IMF's carry information of the underlying structures contained in the observations and can then be used for modelling tasks. The family of wavelet transform methods constitutes another class of methods that operates in the time-frequency domain (see Daubechies 1992 and Percival & Walden 2000). Shensa et al. 1992 first proposed and provided a framework for the discrete wavelet transform (DWT), which belongs to the wavelet transform family. DWT filters the original time series in several folds and yields a denoised version of the observation. The information carried by the transformed time series is then more clear and preferably easier to learn. The rationale of using such techniques as target transformation in time series modelling is to provide the models with a more informative, e.g., less noisy, dataset to learn from. We hope this extra procedure helps produce a better trained (learned) model.

Another branch of decomposition methods is more related to statistical approaches. These decomposition methods generally consider the time series as a mixture of three components: seasonal, trend and remainder. They are often used to filter different information contained within the time series (see Wang, Smith, & Hyndman 2006). For a real-world example, monthly unemployment data are usually presented after removing the seasonality. The resulting time series is hence more indicative of the variation of the general economy instead of seasonal disturbance (see Chapter 3.2 in Hyndman & Athanasopoulos 2021). The STL decomposition proposed by Cleveland et al. 1990 has been a robust method. The abbreviation stands for Seasonal and Trend decomposition using Loess. STL considers a time series as a sum (additive) or product (multiplicative) of the seasonal, trend and remainder components. STL is flexible and applicable to many use cases as it can handle any type of seasonality. Its flexibility also resides in its allowance for the user to have control over the time-varying seasonal component and smoothness of the trend cycle. The X-11 method and the Seasonal Extraction in ARIMA Time Series (SEATS) procedure are time series models that rely heavily on seasonal and trend decompositions. They have had many variants and are favoured by official statistical agencies around the world (see Dagum & Bianconcini 2016)[2]. One of the state-of-the-art variants of this family is the X-13ARIMA-SEATS method produced, distributed and maintained by the US Census Bureau (see US Census Bureau 2012 and Monsell & Blakely 2013). It inherits powerful features from X-11, SEATS, and ARIMA methodologies while specialising in seasonal adjustment in extensive time series modelling. The model is conveniently accessible online[3].

[2] X-11 was initially developed by the US Census Bureau, and SEATS was created by the Bank of Spain.

[3] A webpage demonstration of the model is accessible on `http://www.seasonal.website/`; the open-source implementation of the model can be found in the `seasonal` package in R, and a distributed version can be found in the US Census Bureau website `https://www.census.gov/data/software/x13as.X-13ARIMA-SEATS.html`.

3.2 Directional Change intrinsic time framework

The technical core of the Directional Change (DC) intrinsic time framework is quite simple; it is an algorithm (the DC dissection) that samples a time series and yields a new time series, which is the subset of the original observations. By analysing the properties of the resulting time series, it has been found that despite the simplicity of this algorithm, it provides powerful perspectives for looking at market dynamics in the time domain. In this section, we first look at critical works contributing to the advancements of the DC intrinsic time framework. Then we cover some applications that further develop the framework's value by trying to harness its potential.

Like the developments of many frameworks, the DC intrinsic time framework started out being simple and has developed over time. Guillaume et al. (1997) first published the Directional Change dissection algorithm. The algorithm was presented and used to generate a set of measurements (statistics, variables), from which the authors presented a set of stylised facts found empirically in the spot intra-daily foreign exchange (FX) markets. These stylised facts shed new light on our understanding of market dynamics, especially concerning micro-structure topics, including time-heterogeneity, price formation, market efficiency, liquidity, and both the modelling and the learning process of the market. A little more than a decade later, Tsang formalised the definition of a Directional Change in Tsang (2010), and Glattfelder et al. (2011) discovered a set of twelve scaling laws derived from the DC sampling algorithm. The discovery of the laws added a theoretical foundation to the DC dissection algorithm because the output time series carries not only qualitative information (stylised facts) but also interesting quantitative properties. As the DC dissection algorithm's ability to extract information has been studied, it has given rise to the methodology becoming a framework (see Tsang's introduction of a set of profiles (indicators) derived under the DC framework in Tsang (2015) and (2017)).

One thing we know about analysing financial time series is that the source of many the challenges can be traced back to the use of physical time (see Dacarogna et al. (2001)). Aloud et al. (2012) discussed the potential of studying financial time series using the DC framework (referred to as the DC approach in their paper) resides in its underlying 'intrinsic time' paradigm. They pointed out that mapping financial time series from the physical time to event-based intrinsic time is the key to how the approach filters out irrelevant information and disturbance observed in the dataset and generates valuable market insights of our interests. Inspired by the studies of complex systems, Petrov et al. (2018) took a different route of demonstrating this point with the use of agent-based modelling. They created a market with trading agents that operate in event-based intrinsic time and found that the price movements generated under such conditions experience statistical properties we observed in real-world physical time FX markets. Such reproduction of real-world stylised facts is another indication of the intrinsic time mechanism being one of the contributing factors to the market dynamics. Recently, Glattfelder & Golub (2022) derived an analytical relationship between physical and intrinsic time based on the scaling laws. In particular, the expression they derived decomposes the movements of the physical-time time series into volatility and liquidity components expressed in intrinsic time. That allows us to explicitly characterise the dynamics observed in physical time using its intrinsic-time representation.

As DC intrinsic time framework becomes theoretically sound, applications building on the framework have been devised. Golub et al. (2016) introduced the Intrinsic Network - an event-based framework based on directional changes. Combining the Intrinsic Network and information theory, they devised a liquidity measure that was shown to be able to predict market stress in terms of liquidity shocks. In Golub et al. (2018), the liquidity measure was integrated with other implementations derived from the DC framework and an algorithmic trading strategy called The Alpha Engine was introduced. The Alpha Engine has several interesting features. First, the bare-bones version of the model (without tweaking) has been shown to be robust, profitable, and can be implemented in real-time. Second, Alpha Engine provides liquidity in the market, i.e., it opens long positions when other market players intend to short and vice versa. The Alpha Engine thus contributes to the healthiness of the market as a participant. Third, Alpha Engine 'beats' random walk processes - it is shown to be profitable even on price dynamics generated by a random walk. Within the context of volatility and risk management in finance, Petrov et al. (2019a) proposed an instantaneous volatility measure under the DC intrinsic time framework. They found seasonality patterns and long memory of volatility through empirical studies. Their work contributes not only to the development of practical tools but also to the understanding of the underlying stochastic drive of financial dynamics. Two further generalisations of the DC framework have been proposed. First, Petrov et al. (2019b) brought the framework into multidimensional space by extending the analytical expressions yielded from one-dimensional analyses to multidimensional space. Their methodology implies that previous works in one-dimensional space (analytical insights, empirical findings, and all the tools and implementations) can be extended to higher dimensions. Another generalisation was developed with respect to the types of stochastic processes (Mayerhofer (2019)). These generalisations, as well as the advancements of the DC intrinsic time framework discussed previously, are indicative of the framework's promising potential worthy of further exploration.

In this chapter, we reviewed relevant works in two different realms: target transformations being used as an additional layer in modelling and the DC intrinsic time framework being a rising methodology. The literature surveyed demonstrates excellent potential in both research directions that justifies our curiosity in bringing them together. In the next chapter, we go into detail about the methodologies of combing the framework and target transformation in time series modelling.

Chapter 4

Methodology

Chapter 5

Results and Evaluation

Chapter 6

Conclusion

Bibliography

A. Adegboye and M. Kampouridis. Machine learning classification and regression models for predicting directional changes trend reversal in fx markets. *Expert Systems with Applications*, 173:114645, 2021.

S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah. Time-series clustering - a decade review. *Information Systems*, 53:16–38, 2015. ISSN 0306-4379. doi: https://doi.org/10.1016/j.is.2015.04.007. URL `https://www.sciencedirect.com/science/article/pii/S0306437915000733`.

N. Ahmed, A. Atiya, N. Gayar, and H. El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29:594–621, 08 2010. URL `https://www.tandfonline.com/doi/abs/10.1080/07474938.2010.481556`.

M. Aloud, E. Tsang, R. Olsen, and A. Dupuis. A directional-change event approach for studying financial time series. *Economics*, 6(1), 2012.

M. S. Bartlett. The use of transformations. *Biometrics*, 3(1):39–52, 1947. ISSN 0006341X, 15410420. URL `http://www.jstor.org/stable/3001536`.

C. Bergmeir, R. J. Hyndman, and J. M. Benítez. Bagging exponential smoothing methods using stl decomposition and box–cox transformation. *International journal of forecasting*, 32(2):303–312, 2016.

P. J. Bickel and K. A. Doksum. An analysis of transformations revisited. *Journal of the american statistical association*, 76(374):296–311, 1981.

P. Bloomfield. *Fourier analysis of time series: an introduction*. John Wiley & Sons, 2004.

G. E. Box and D. R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243, 1964.

U. C. Bureau. *X-13ARIMA-SEATS Reference Manual, Version 1.0.* U.S. Census Bureau, U.S. Department of Commerce, 2012.

R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning. Stl: A seasonal-trend decomposition. *J. Off. Stat*, 6(1):3–73, 1990.

E. B. Dagum and S. Bianconcini. *Seasonal adjustment methods and real time trend-cycle estimation.* Springer, 2016.

I. Daubechies. *Ten lectures on wavelets.* SIAM, 1992.

J. G. De Gooijer and R. J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443–473, 2006. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2006.01.001. URL `https://www.sciencedirect.com/science/article/pii/S0169207006000021`. Twenty five years of forecasting.

J. D. Farmer and D. Foley. The economy needs agent-based modelling. *Nature*, 460 (7256):685–686, 2009.

W. A. Fuller. *Introduction to statistical time series.* John Wiley & Sons, 2009.

R. Gençay, M. Dacorogna, U. A. Muller, O. Pictet, and R. Olsen. *An introduction to high-frequency finance.* Elsevier, 2001.

J. B. Glattfelder and A. Golub. Bridging the gap: Decoding the intrinsic nature of time in market data. *arXiv preprint arXiv:2204.02682*, 2022.

J. B. Glattfelder, A. Dupuis, and R. B. Olsen. Patterns in high-frequency fx data: discovery of 12 empirical scaling laws. *Quantitative Finance*, 11(4):599–614, 2011.

A. Golub, G. Chliamovitch, A. Dupuis, and B. Chopard. Multi-scale representation of high frequency market liquidity. *Algorithmic Finance*, 5(1-2):3–19, 2016.

A. Golub, J. B. Glattfelder, and R. B. Olsen. The alpha engine: Designing an automated trading algorithm. In *High-Performance Computing in Finance*, pages 49–76. Chapman and Hall/CRC, 2018.

P. Goodwin et al. The holt-winters approach to exponential smoothing: 50 years old and going strong. *Foresight*, 19(19):30–33, 2010.

V. M. Guerrero. Time-series analysis supported by power transformations. *Journal of forecasting*, 12(1):37–48, 1993.

D. M. Guillaume, M. M. Dacorogna, R. R. Davé, U. A. Müller, R. B. Olsen, and O. V. Pictet. From the bird's eye to the microscope: A survey of new stylized facts of the intra-daily foreign exchange markets. *Finance and stochastics*, 1(2): 95–129, 1997.

N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences*, 454(1971):903–995, 1998.

J. C. Hull. *Options futures and other derivatives.* Pearson Education India, 2003.

R. Hyndman and G. Athanasopoulos. *Forecasting: Principle and Practice, 3rd edition.* OTexts: Melbourne, Australia, 2021. URL `OTexts.com/fpp3`. Accessed in June 2022.

R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

J. Hämäläinen and T. Kärkkäinen. Problem transformation methods with distance-based learning for multi-target regression. 10 2020.

S. M. Kay and S. L. Marple. Spectrum analysis—a modern perspective. *Proceedings of the IEEE*, 69(11):1380–1419, 1981.

A. J. Koning, P. H. Franses, M. Hibon, and H. Stekler. The m3 competition: Statistical tests of the results. *International Journal of Forecasting*, 21(3): 397–409, 2005. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2004.10.003. URL `https://www.sciencedirect.com/science/article/pii/S0169207004000810`.

S. Makridakis and M. Hibon. The m3-competition: results, conclusions and implications. *International journal of forecasting*, 16(4):451–476, 2000.

S. Makridakis, E. Spiliotis, and V. Assimakopoulos. The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2019.04.014. URL `https://www.sciencedirect.com/science/article/pii/S0169207019301128`. M4 Competition.

E. Mayerhofer. Three essays on stopping. *Risks*, 7(4):105, 2019.

A.-H. Mihov, N. Firoozye, and P. Treleaven. Towards augmented financial intelligence. *Available at SSRN*, 2022.

B. Monsell and C. Blakely. X-13arima-seats and imetrica. *US Census Bureau, Washington, DC*, 2013.

D. B. Percival and A. T. Walden. *Wavelet methods for time series analysis*, volume 4. Cambridge university press, 2000.

V. Petrov, A. Golub, and R. B. Olsen. Agent-based model in directional-change intrinsic time. *Available at SSRN 3240456*, 2018.

V. Petrov, A. Golub, and R. Olsen. Instantaneous volatility seasonality of high-frequency markets in directional-change intrinsic time. *Journal of Risk and Financial Management*, 12(2):54, 2019a.

V. Petrov, A. Golub, and R. B. Olsen. Intrinsic time directional-change methodology in higher dimensions. *Available at SSRN 3440628*, 2019b.

M. J. Shensa et al. The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, 40(10):2464–2482, 1992.

E. Tsang. Directional changes, definitions. *Working Paper WP050-10 Centre for Computational Finance and Economic Agents (CCFEA), University of Essex Revised 1, Tech. Rep.*, 2010.

E. P. Tsang, R. Tao, and S. Ma. Profiling financial market dynamics under directional changes. *Quantitative finance, http://www. tandfonline.com/doi/abs/10.1080/14697688.2016*, 1164887, 2015.

E. P. Tsang, R. Tao, A. Serguieva, and S. Ma. Profiling high-frequency equity price movements in directional changes. *Quantitative finance*, 17(2):217–225, 2017.

X. Wang, K. Smith, and R. Hyndman. Characteristic-based clustering for time series data. *Data mining and knowledge Discovery*, 13(3):335–364, 2006.

H. White. Maximum likelihood estimation of misspecified models. *Econometrica: Journal of the econometric society*, pages 1–25, 1982.

I.-K. Yeo and R. A. Johnson. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959, 12 2000. ISSN 0006-3444. doi: 10.1093/biomet/87.4.954. URL `https://doi.org/10.1093/biomet/87.4.954`.

Appendix A

chapter one

Something about chapter one

Another something ...

And another something ...

A.1 ch1 sec 1

Something about chapter one, section 1

Appendix B

chapter two

Something about chapter two

B.1 ch2 sec 1

Something about chapter two, section 1

B.2 ch2 sec 2

Something about chapter two, section 2

Appendix C

chapter three

Something about chapter three

C.1 ch3 sec 1