FRANKFURT
UNIVERSITY
OF APPLIED SCIENCES

HIS Project

Time Series Analysis

Assignment 1

Mehjabeen Jahangeer Khan – 1377473

1. Mathematical Definition of Time Series

A time series is a series of data points indexed sequentially over time. The most common form of time series is a sequence of observations recorded over time [1]. Time series are often divided into univariate (one-dimensional) and multivariate (multi-dimensional).

Univariate Time Series

As the name implies, a univariate time series (UTS) is a series of data that is based on a single variable that changes over time. Keeping a record of the humidity level every hour of the day would be an example of this. The time series X with t timestamps can be represented as an ordered sequence of data points in the following way:

$$X = (x1, x2, \ldots, xt)$$

where xi represents the data at timestamp $i \in T$ and$T = \{1,2,\ldots,t\}$.

In addition, a time series in mathematics and statistics, is a sequence of data points indexed by time, where each point is denoted by a timestamp. Formally, a time series can be defined as a sequence of values, X_t, where t ∈ T(with T representing discrete time intervals such as days, months, or years) [6]. Each value in the series represents an observation at a given time t. Mathematically, a time series can be expressed as:

$$X = \{X_t : t = 1,2,3,\ldots,n\}$$

Here:

- X is the time series.
- X_t is the value observed at time t.
- t denotes discrete time steps, though time can also be continuous in more advanced models.

Time series data is unique because observations are not independent, but rather ordered chronologically, often showing autocorrelation or seasonality.

Multivariate Time Series

Additionally, a multivariate time series (MTS) represents multiple variables that are dependent on time, each of which is influenced by both past values (stated as "temporal" dependency) and other variables (dimensions) based on their correlation. The correlations between different variables are referred to as spatial or intermetric dependencies in the literature, and they are used interchangeably [5]. In the same example, air pressure and temperature would also be recorded every hour besides humidity level. Consider a MTS represented as a sequence of vectors over time, each vector at time i, Xi, consisting of d dimensions:

$$X = (X1, X2, \ldots, Xt) = ((x1\ 1, x2\ 1, \ldots, xd\ 1), (x1\ 2, x2\ 2, \ldots, xd\ 2), \ldots, (x1\ t, x2\ t, \ldots, xd\ t))$$

where Xi = ($x1\ i$,$x2\ i$, . . .,$xd\ i$) represents a data vector at time i, with each $x\ j\ i$ indicating the observation at time i for the jth dimension, and j = 1,2,. . .,d, where d is the total number of dimensions.

2. **Time Series Analysis (TSA)**

Time Series Analysis involves methods for analyzing time-ordered data to identify underlying patterns, trends, and relationships over time. It aims to understand the structure and dynamics of the data, model it, and make forecasts or infer information about future values. Key components include:

- **Trend Analysis**: Identifying long-term increase or decrease patterns.
- **Seasonal Analysis**: Recognizing patterns repeating at regular intervals (e.g., daily, monthly).
- **Autocorrelation and Stationarity**: Understanding dependencies across time and ensuring statistical properties like mean and variance do not change over time.
- **Forecasting**: Predicting future values using past and present data, with popular methods such as ARIMA (Auto-Regressive Integrated Moving Average) models and machine learning-based models.

Summarize chapter 1 of the two books

(a) Modern time series forecasting with Python

Chapter 1 of *Modern Time Series Forecasting with Python* offers a comprehensive introduction to time series analysis, covering essential definitions, applications, methodologies, and terminology. Here's a detailed breakdown:

Purpose and Scope

The chapter begins by emphasizing the importance of time series analysis for data scientists and machine learning (ML) engineers who want to move beyond traditional ML methods like classification and regression to unlock business value from time-based data. The book seeks to transition readers from basic statistical methods to advanced ML techniques in time series forecasting.

Key Concepts and Definitions

1. **What is a Time Series?**

   A time series is a set of sequential observations taken over time. Examples include daily stock prices, monthly rainfall, and hourly sensor data. Time series data focuses on the temporal sequence, aiming to understand patterns and make predictions based on past behavior.

2. **Types of Time Series**
    - **Regular Time Series**: Observations occur at consistent intervals (e.g., daily, monthly).
    - **Irregular Time Series**: Observations are inconsistent over time (e.g., patient health checkups).

This book focuses on regular time series, suitable for ML-based forecasting.

Applications of Time Series Analysis

The chapter outlines three primary applications:

- **Forecasting**: Predicting future values based on historical data, such as temperature forecasting or financial trend analysis.
- **Classification**: Categorizing time series data based on past patterns, such as identifying abnormalities in health data (e.g., EEG readings).
- **Interpretation and Causality**: Analyzing relationships and causations within time series or between multiple series, though the book notes that causal inference is outside its scope.

Data-Generating Process (DGP)

The DGP is the mechanism behind time series generation. Each time series is influenced by various factors (e.g., seasonal demand, machine performance, or external events). Due to incomplete knowledge, DGP is approximated using models:

- **Model vs. Reality**: Models aim to capture key aspects of the DGP but are inherently simplifications, similar to how a map represents, but does not replicate, a physical landscape.

Predictability of Time Series

Not all time series are equally predictable. The chapter illustrates predictability with examples:

- **Highly Predictable**: Cyclical events, like ocean tides.
- **Random**: Lottery numbers.
- **Complex, but Predictable**: Stock prices, though with limitations due to their random-like behavior and the efficient-market hypothesis, which suggests that all known information is already factored into stock prices.

The book proposes a mental model to gauge predictability based on:

1. **Understanding the DGP**: The more known about the generation process, the easier forecasting becomes.
2. **Data Quantity**: More data points typically lead to better predictions.
3. **Repetition of Patterns**: The clearer and more frequent the patterns, the easier the predictability.

Terminology and Forecasting Techniques

Understanding time series vocabulary is essential for advanced analysis. Key terms introduced include:

1. **Forecasting**: Predicting future values using historical data and related factors.
2. **Multivariate Forecasting**: Involving multiple dependent variables, like multiple economic indicators influencing each other.
3. **Explanatory Forecasting**: Using additional, related data (e.g., promotions impacting sales) to improve forecasting accuracy.
4. **Backtesting**: Assessing model performance by comparing predictions on historical data split into training (in-sample) and testing (out-sample) sets.
5. **Exogenous and Endogenous Variables**: Exogenous variables (e.g., weather) impact the model without being affected by it, while endogenous variables are influenced by factors within the system.
6. **Forecast Combination**: Similar to ML ensembling, where multiple forecasts are combined to improve accuracy.

(b) Deep Learning for Time Series Cookbook: Use PyTorch and Python recipes for forecasting, classification, and anomaly detection.

This chapter introduces the fundamentals of time series analysis, particularly within the context of deep learning applications, providing a foundation for loading, analyzing, and preparing time series data. It emphasizes key preprocessing techniques and tools used for working with time series data in Python, especially using libraries like pandas, statsmodels, and seaborn.

Introduction to Time Series

The chapter begins by defining time series data and highlighting its significance in data science. Time series analysis is essential for fields like forecasting, anomaly detection, and trend analysis. The chapter also introduces key time series characteristics, such as trend and seasonality, along with the concept of stationarity.

Key Components of Time Series

A time series consists of several core components:

- **Trend**: Represents the general direction of the data over a long period.
- **Seasonality**: Periodic fluctuations that occur at regular intervals, such as daily, monthly, or yearly patterns.
- **Stationarity**: A key property in time series that implies consistency in the data's statistical properties over time. Non-stationary data, often affected by trends and seasonality, needs preprocessing before modeling.
- **Noise (or Residual)**: The random variation that remains after removing trends and seasonal effects.

Understanding these components is critical for accurately modeling time series data and extracting valuable insights.

Technical Requirements

The chapter recommends the following Python libraries to handle time series data:

- **pandas (2.1.4)** for data manipulation
- **numpy (1.26.3)** for mathematical operations
- **statsmodels (0.14.1)** for statistical tests
- **pmdarima (2.0.4)** for ARIMA modeling
- **seaborn (0.13.2)** for data visualization

The authors suggest using pip to install these packages and provide links to datasets on GitHub for hands-on learning.

Key Techniques Covered

The chapter covers multiple practical recipes that offer step-by-step guidance on working with time series data. Below is a breakdown of these recipes:

1. **Loading Time Series Data with pandas**
    - *Overview*: The first recipe focuses on loading time series data, specifically using pandas, one of the most popular Python libraries for data analysis.
    - *Example Dataset*: The chapter uses solar radiation data collected by the U.S. Department of Agriculture, spanning hourly readings over six years.
    - *Method*: This involves reading a .csv file with pd.read_csv() and using Datetime as the index column, enabling easy time-based manipulations.
2. **Visualizing Time Series Data**

- *Purpose*: Visualization helps identify patterns such as seasonality or trends.
- *Tools*: Two libraries are used: pandas for quick plots and seaborn for more detailed visualizations.
- *Steps*: Users are guided through plotting time series data, adjusting figure size, titles, and labels, and using seaborn to create visually appealing plots.

3. **Resampling Time Series**
   - *Overview*: Resampling alters the frequency of time series data, a common preprocessing step.
   - *Scenarios Covered*: The recipe shows how to change hourly data to daily data by summing values within each day, and also demonstrates how to manage irregular time series data (data with inconsistent intervals).
   - *Example Code*: series.resample('D').sum() for daily summation of hourly data.
4. **Handling Missing Values**
   - *Challenge*: Missing values are common in time series due to factors like sensor failures.
   - *Methods*: Multiple imputation methods are discussed, including:
     - **Mean Imputation**: Replacing missing values with the average of the series.
     - **Forward Fill (ffill)**: Filling missing values based on the last available observation.
     - **Backward Fill (bfill)**: Using the next available observation.
   - *Insights*: ffill maintains the temporal integrity of data, while mean imputation may obscure time-dependent trends.
5. **Time Series Decomposition**
   - *Concept*: Decomposition breaks down a series into components—trend, seasonality, and residual (irregular) elements.
   - *Methods Covered*:
     - **Classical Decomposition**: Using statsmodels to apply an additive or multiplicative model.

        - **STL (Seasonal-Trend Decomposition using LOESS)**: A more flexible decomposition method suitable for non-linear trends.
        - **MSTL (Multiple STL)**: Extends STL for handling multiple seasonal patterns, such as weekly and yearly cycles in daily data.

6. **Autocorrelation Analysis**
    - *Purpose*: Autocorrelation measures how a time series correlates with itself over different lags, helping to identify seasonal patterns.
    - *Implementation*: The chapter shows how to use acf (autocorrelation function) and pacf (partial autocorrelation function) from statsmodels to analyze patterns up to specified lag periods.
7. **Stationarity Testing**
    - *Importance*: Many time series models assume stationarity—constant mean and variance over time.
    - *Testing Methods*:
        - **Augmented Dickey-Fuller Test**: Checks if a series is stationary.
        - **KPSS Test**: Assesses if a series requires differencing to become stationary.
    - *Practical Steps*: The chapter covers how to use the ndiffs() function to estimate the number of differencing steps required for stationarity and shows how to apply differencing.
8. **Dealing with Heteroskedasticity**
    - *Definition*: Heteroskedasticity refers to changes in variance over time, which can affect model performance.
    - *Testing for Variance Consistency*: The White and Breusch-Pagan tests in statsmodels help identify heteroskedasticity.
    - *Variance Stabilization*: Logarithmic transformations are recommended to stabilize variance, and the chapter introduces Box-Cox transformations for further flexibility.

Advanced Multivariate Time Series Analysis

The chapter extends time series analysis to multivariate data, which involves multiple variables or features evolving over time, as commonly seen in complex systems (e.g., weather models).

1. **Loading and Visualizing Multivariate Time Series**

- *Approach*: Multivariate time series are represented in `pandas` DataFrames, unlike univariate series stored in Series objects.
- *Transformation and Visualization*: The chapter covers visualizing multivariate series with the `plot()` function, while highlighting the need to preprocess individual variables separately.

2. **Resampling Multivariate Series**
    - *Challenges*: Different variables may require distinct resampling techniques. For instance, solar radiation might be summed daily, while wind speed may use daily averages.
    - *Method*: A dictionary can specify distinct aggregation rules for each variable, allowing flexible resampling.
3. **Analyzing Correlation Among Variables**
    - *Purpose*: Examining correlations between variables helps understand interactions within multivariate time series.
    - *Method*: The chapter demonstrates how to create correlation matrices and visualize them as heatmaps using `seaborn`.

References:

1. James Douglas Hamilton. 2020. Time series analysis. Princeton university press.
2. *Hamilton, J.D. (1989). "A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle."* Econometrica.
3. *Chakrabarti, K., & Mehrotra, S. (2000). "Local Dimensionality Reduction: A New Approach to Indexing High Dimensional Spaces."*
4. *Hyndman, R.J., & Athanasopoulos, G. (2018). "Forecasting: Principles and Practice."* This open-access textbook provides a
5. Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. 2021. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In KDD. 3220–3230.
6. *Zhao, C., & Yuan, X. (2020). "Time Series Anomaly Detection: A Survey."* arXiv preprint arXiv:2012.01803.
7. *OpenAI. (2024). ChatGPT (October 2024 version) [Large language model].*