Time Series

Applied Mathematics

Shay Malkin
Tal Ladijensky
Nir Titelbom

Introduction

Time series is a sequence of data points in chronological sequence, most often gathered in regular intervals.

Time series adds an explicit order dependence between observations: a time dimension.

In addition, time series forecasting is an important area of machine learning.

Forecasting involves making models fit on historical data and using them to predict future observations.

An important distinction in forecasting is that the future is completely unavailable and must only be estimated from what has already happened.

Examples: average daily temperatures, stock value at the end of each business day, hourly electricity demand.

Components of Time Series

Time series analysis provides a body of techniques to better understand a dataset. Perhaps the most useful of these is the decomposition of a time series into 4 constituent parts: trend, seasonality, noise or randomness, and the level. not all time series data will include every one of these time series components. For instance, audio files that are taken in sequence are examples of time series data, however they won't contain a seasonal component. On the other hand, most business data will likely contain seasonality.

- Level:

  When we read about the "level" or the "level index" of time series data, it's referring to the mean of the series.

- Noise:

  All time series data will have noise or randomness in the data points. The optional variability in the observations that cannot be explained by the model.

- Seasonality:

  If there are regular and predictable fluctuations in the series that are correlated with the calendar, then the series includes a seasonality component. It’s important to note that seasonality is domain specific, for example real estate sales are usually higher in the summer months.

- Trend:

  The optional and often linear increasing or decreasing behavior of the series over time. An example of a trend would be a long term increase in a companies sales data.

Time series forecasting

Predicting the future values of the time series using a current information set.

Current information set: current and past values of the series and other "exogenous" series.

Information set definition:

x_t - exogenous series values at time t

y_t - series value at time t

$$\hat{y}_{t+K|t} = f\left(y_t, y_{t-1}, \ldots, y_1, x_t, x_{t-1}, \ldots, x_1\right)$$

Forecasting models

Naive, rule based models:

- ❖ Constant value: $\hat{y}_{t+1} = a$
- ❖ Last value: $\hat{y}_{t+1} = y_t$
- ❖ Rolling average: $\hat{y}_{t+1} = \frac{1}{M} \sum_{s=t-M}^{t} y_s$

Statistical models:

- ❖ Auto regressive model: $\hat{y}_{t+K} = \left(y_t, y_{t-1}, \ldots, y_{t-M}\right)^T \beta$

M is memory length, β predictor weights, $\hat{y}_{t+K}$ is linear function of inputs window $y_{t-M:t}$

Time series forecasting with Neural networks:

In many cases, a time series is quite complex and cannot be accurately fitted with a linear model. We'll need a more complex, non-linear model.

Feed forward networks are a good place to start.

The problem is that if we simply feed the network with the past values of the series as the input vector, the model doesn't account for time order.

One way to deal with that is to use a convolutional layer (or more than one) before the fully connected layers. The convolution takes a temporally coherent input and is able to extracts meaningful features. Those features are then passed on to the fully connected layers for classification.

DAE:

DAE- Denoising Autoencoders is a way to handle slightly corrupt data.

Time series is <u>often inconstient</u> and in forecasting models, we do not want to use training data which is problematic, it makes our goal to minimize the harm that the noise is creating to our data.

If we use a data which is corrupt or a small percentage is missing, DAE will provide us with a 'fixed' data to input into our NN preinitialization, for example: Salt-and-pepper noise or failed data recovery.

The idea is to train our DAE with parts of the data that we know are trustworthy. This way, we can later use it to fix the corrupted data before feeding it to our model.

(For more information on DAE go to the last presentation - Artificial Neural Network).

Training the model:

Given a time series, we will use it as our training and test data.

The idea is to split the data at some point in time. All the data up to that point will be the training data, and everything beyond that point will be the test data.

Having part of the series as training data, we can now take different time windows as data points to train our network (knowing their correct labels).

We will use the training data to optimize the cost function defined as:

$$error = \frac{1}{k}\sum_{i=1}^{k}\left(\hat{y}_{t+i} - y_{t+i}\right)^2$$

To be continued.....