Time Series Fundamentals

Pooja Parameswaran and Lucy Chikwetu
Duke University

What is time series data?

When data is gathered, it is often representing some activity over a period. This requires analysis with time itself. Therefore, we introduce a topic called time series data that evaluates activities while factoring in time. It is a key method to track change. Time series data is used in multiple analyses, including calculating brain activity, rainfall measurements, activity changes, retail sales, stocks, and more. Time Series data reflects data being observed over consistent intervals and is classified on a linear basis, describing how certain actions occur as a product of time and is collected from the same source.

What are the components of a time series dataset?

Time series data contains features, classes, and a timestamp. The time component is key to calculate change in a long pattern of activities/situations. The data gathered against time will be used to reflect the activity being observed, and these features will be evaluated with a classification network to differentiate between activities. Most of the data being gathered will be serially dependent, so one data point is dependent on another datapoint. The time values are correlated with data reflecting activities quantitatively, and the data is also classified in a certain way. To understand the data and classification, we will ignore the time aspect momentarily. Each data point is presented with a class, describing the data values. The data values are presented as ‘features’ which are specific details that separate them into different groups or classes. The more features provided, the easier it is for a machine learning model to distinguish between separate classes. Concatenating all the features while ignoring the time variable is useful in looking at all the data. However, this causes the training data size to be large, and unparalleled to what the prediction sizes should be.

How to read a time series model?

We can read a time series model effectively with a sliding window. Often, time series data represents a long period of time, so the data size can be colossal. This can cause classification and proper feature understanding to be difficult. However, if a sliding window is utilized, a smaller portion of the data can be analyzed at a time, and the window can ‘slide’ across the entire dataset. Consequently, focusing on specific details while also understanding all the data. This allows better classification between different groups.

How does a sliding window function?

A sliding window sections off a specific amount of data, usually as specified as a time segment by the user and performs analysis on it. In order to ensure continuity, there is usually overlap between consecutive sliding windows. It is significant to carve out a single time frame and further analyze that. This allows more detailed analysis of the entire time series as change can be found between time windows rather than analyzing every data point against the rest. This ensures any data analyzed from the sliding window is part of a continuous system. From the segmented window, the given features can be further analyzed, and a new feature can be engineered- one that describes all the data in that specific window. This allows for the data size to be reduced so the model can handle the data more efficiently, creating a more robust and optimal classification model.

What are features? How are they used?

A dataset is measured with features. A feature is a measurable property of the object/activity being analyzed and are used to detail classes. In a classification model, the features are used as variables to discern between classes. The more features, the more data the model can use to differentiate between classes. Classes are categories describing the activities occurring in the time series dataset presented. In order to have an optimal network that can classify between groups and handle any prediction value optimally, many features are required. This means, some may need to be engineered from the real-time data already gathered.

What is feature engineering?

Feature engineering is the process of dimensionality reduction in which a set of raw data is reduced and combined to a more manageable group for processing. It is very useful to simplify and speed up data transformations while enhancing model accuracy. This method allows large datasets to be downsized for more readability and easier descriptions. This is useful when it is necessary to process many data points without losing important information. In order to reduce dimensionality, a section of the given time series data (a window) is taken and then engineered. An example is when we have one sliding window with (100) elements, and we downsize it with a mathematical or statistical operation, calculating a singular value that describes all the data points in some way. One example would be taking the mean of a time window. This is an engineered feature that is extracted from the given dataset via an extracted window. This considerably reduces the data size and complexity, while also providing more data categories to distinguish between to emphasize the class being observed.

How does sliding window relate to feature selection?

In machine learning, feature engineering is the best way to reduce a large dataset into new features that define it more closely. This is done via a method called feature engineering- the process of selecting, manipulating, and transforming raw data into features to use in supervised machine learning. These features accurately reflect the larger dataset, and include mean, median, mode, standard deviation, and more measurements. A sliding window must be first implemented to properly engineer features, so an even amount of data is extracted with levels of continuity.

What is a machine learning model?

A machine learning model is an algorithm that is trained to recognize patterns in data. The model is trained using a set of real-life data, in which it learns and reasons between patterns. The different patterns get signified as classes or groups, and certain data reflects each group. The goal of machine learning models is to accurately predict any incoming data and classify it appropriately based off how the input data trained the network. This is very useful in scenarios that require repeated decisions or evaluations. Supervised machine learning datasets are most used with time series data.

An input dataset is provided to the network and is rewarded/optimized to meet the desired outputs or classes. This is known as classification. The input data has a clear goal- classifications that are provided alongside the input data- that is aspires to meet. The algorithm, consequently, adjust/creates weights to more important aspects of the network, giving weightage to the details that ultimately best determine the group the object/data point belongs in. There are many types of supervised machine learning networks, including Logistic Regression, Support Vector Machines, Naïve Bayes, Decision Trees, and Linear Regression.

Different Classification models?

Different classification models perform classification analysis differently. Therefore, it is important to closely analyze the input dataset to determine which model will be most optimal to create the best classification model, with the most accurate predictions. A logistic regression model is used to determine if an input belongs to a certain group. A Support Vector Machine (SVM) creates coordinates for each object in a class and uses a hyperplane to group object by similar features. SVM maps inputs into high-dimension feature spaces, and is effective when dealing with more than two features, can be an N-dimensional model, where N is the number of features. A naïve bayes model is an algorithm that assumes independence between variable and uses probability to classify objects with the given features. There are many classification networks available, and more research can be done into each type to determine if the model best suits your data's needs.

Model Selection

When dealing with time series data, features must often be engineered from interval-based windows. Engineered features usually become numerous, so with a time series dataset, it is best to pick a model that can take numerous amounts of features and can create a clear boundary between groups. This way, any segment of data inserted, can be feature engineered and then categorized correctly. Sometimes, the features may also have a dependence on each other, so it is best to map it in space where each feature somewhat is reliant on the rest. Consequently, using a SVM algorithm with time series data is one of the best ways to create an optimal classification model for further predictions. However, to check if the model is appropriate for the given dataset, metrics can be used to evaluate the model performance. Four of the main metrics used to check the model's performance are accuracy, precision, recall, and F1. Accuracy is the percentage reflectin how much of a model output is equivalent to real output. A confusion matrix is plotted by the model, describing how many values are accurately classified (True Positives + True Negatives) and how many are falsely classified (False Positives + False Negatives). Precision is the ratio of true positives by all the positives, describing how able the model is to describe negatives as negatives. Recall is another metric describing the ratio of true positives by true positives and false negatives. This describes the ability of the model to describe positive samples as positives. F1, or F-score is the last metric used to define model performance. This is the mean of the precision and recall. A value of one is optimal, and a value of zero is worst. It is calculated by the formula:

$$F1 = 2 * \frac{Precision * Recall}{(Precision + Recall)}$$

These metrics should be calculated to ensure the model is performing well and it can show whether the selected classification model should be pursued, or another should be looked into.

References

(1) “What Is Time Series Data?: Definition, Examples, Types & Uses.” *InfluxData*, 1 May 2022, https://www.influxdata.com/what-is-time-series-data/.
(2) DeepAI. “Feature Extraction.” *DeepAI*, DeepAI, 17 May 2019, https://deepai.org/machine-learning-glossary-and-terms/feature-extraction.
(3) “What Are Machine Learning Models?” *Databricks*, 18 Jan. 2022, https://databricks.com/glossary/machine-learning-models.