What is time series classification?

Learn about time series classification, the process of analyzing multiple labeled classes of time series data and then predicting or classifying the class that a new data set belongs to

By Timothy R. Dinger, Yuan-chi Chang, Raju Pavuluri, Shankar Subramanian
Published January 26, 2022

Time series classification uses supervised machine learning to analyze multiple labeled classes of time series data and then predict or classify the class that a new data set belongs to. This is important in many environments where the analysis of sensor data or financial data might need to be analyzed to support a business decision. Accuracy of classification is critical in these situations, so data scientists work hard to ensure that their time series classifiers are as accurate as possible.

Types of time series classification

There are many algorithms that are designed to perform time series classification. Depending on the data, one type might produce higher classification accuracies than other types. This is why it's important to consider a range of algorithms when diving into a time series classification problem. The use of an automated platform that would rigorously explore the space of available algorithms and hyperparameters might save significant time in at least the initial stages of exploration by indicating the algorithmic pipeline that produces the optimized accuracy for the input data set. Such platforms with time series classification capability are expected in the near future.

Distance-based approaches

A distance measure is an objective score that summarizes the relative difference between two objects in a problem domain. The smaller the distance measure between two objects (typically data describing something), the more similar the items are. Some types of distance measures that are typically used in machine learning are:

1. Euclidian distance

2. Hamming distance
3. Manhattan distance
4. Minkowski distance

These distance measurements are used along with some well-known distance-based algorithms such as k-nearest neighbors (KNN). It measures the distance between the test object and all of the objects in the training data set. The k shortest distances are then selected, and the new object is assigned the class that is most represented in the k objects from the training set. When k is set to one, the algorithm reduces to the one-nearest neighbor, and the test object is assigned the class of the training set sample with the shortest distance.

There are other kernel-based algorithms that use distance measures at their core. Perhaps the most well-known of these algorithms is the support vector machine (SVM). This algorithm creates a hyperplane (or line in 2-dimensions) to separate objects into classes. The position of the test object is then calculated with respect to the hyperplane and the class is assigned accordingly.

Dynamic time warping (DTW) is a distance-based algorithm that is used for measuring the distance between two time series. DTW does this by calculating the distances between each point in the time series and summing these for the overall distance. The algorithm is constructed to deal with slight shifts between very similar time series. DTW in conjunction with 1-NN has been the gold standard for time series classification for the past decade and is almost always used as a comparative algorithm in benchmarking studies.

Shapelet

Time series data often exhibits characteristic data shapes that are indicative of the class of the time series. A shapelet transform algorithm can analyze the time series subsequences and generate output useful to a classifier to discriminate classes. Characteristic ECG shapes present in subsequences of the heartbeat and which are indicative of heart disease would be a perfect problem for this type of classification algorithm.

Time series data that is passed through a shapelet algorithm produces output showing a minimum distance measure between the shapelet and all subsequences in the data set. As such, it is a type of distance-based algorithm that is similar to DTW except that the shapelet transform only measures the distance for subsequences of the data and not the entire time series. Some of the highest performing time series classifiers are composed of multiple classifiers (ensembles) that use data transformed through a shapelet transform.

Model ensembles

An ensemble model for time series classification is a collection of classification models that each perform their own class discrimination on the data set. The class that results most often from the collected classifiers is then the class applied to the data set. For this approach to work, any error that is produced by the classifiers should not be correlated. If the errors are correlated, the strength of the approach is lost, and the ensemble approaches the accuracy of a single classifier.

If multiple types of algorithmic classifiers are aggregated into an ensemble, it's possible to reduce the effect of correlated error by grouping the classifiers by type and taking a single classification from the group. This is then used with the classification from the other groups for the ensemble classification. Ensembles of this kind are the basis for the HIVE-COTE algorithm, which performs very well on relatively small time series classification

Dictionary approaches

Another popular type of classifier algorithms is based on the structure of a dictionary, which is used to describe the meaning of a work in the most common use of the term. But, a dictionary approach can describe objects other than words also. A dictionary could be used to describe the numbers of occurrences of a particular shapelet in a time series.

A structure very similar to a shapelet, called a kernel, is used in pattern analysis and can be the source of the dictionary for subsequent classification. An algorithm that uses this approach is now quite popular and can be accessed through sktime ROCKET. ROCKET uses random convolutional kernels to generate a dictionary that is subsequently used to train a Ridge classifier (also available as part of scikit-learn) for small data sets, or a linear regression classifier for large data sets.

For text classification, a highly useful dictionary approach is the Bag-of-Words (BoW) algorithm that counts the occurrence of words within a document, and this information is then used to train a BoW model. The intuition here is that documents are similar if their features (word counts) are similar. A similar approach that can be used on time series data uses the Bag-of-Patterns (BoP) algorithm that, instead of counting words, looks at the amplitude of a time series signal within some specified window of the data and applies a simple transformation to the signal to represent the mean of the signal within the window. This feature then becomes the basis for the dictionary that is used to train a classifier.

Interval-based approaches

Interval-based methods are based on splitting the time series into distinct intervals, similar to the Bag-of-Patterns method previously discussed. Each interval is then used to train an individual machine learning model (classifier). This approach generates an ensemble of classifiers, with each acting on its own subsequence/interval. The final classification is assigned based on the most common class that is generated by the individual classifiers.

The most common interval-based algorithm is the time series forest (TSF). This method uses a decision tree for each interval, with the aggregated decision trees being the forest. Each decision tree is a machine learning model that then assigns a class to its interval of the data. Because the decision trees are training on a different interval of the overall time series, they might not produce the same classification, which is why the ensemble voting process is needed.

One thing to keep in mind with this approach is that the runtime goes up linearly with the length of the time series sample, the number of time series samples, and the number of features per interval. So this can lead to a rather long computation for large data sets.

Deep learning

Deep learning models are a type of neural network that has multiple layers of neurons or perceptrons. These models are typically much more complex with many more parameters than other types of algorithmic models. The initial layers of the deep learning network encode fundamental shapes within the time series data, while latter layers encode representations that can then be discriminated into classes in the final layer.

While this type of learning algorithm has had enormous success in classification of language and images, work using deep learning approaches to classify time series data has lagged but is now gaining momentum due to trials showing similar accuracy to conventional algorithms but with training times orders of magnitude faster on similar compute environments. Accuracy from techniques such as the sktime ROCKET transform compare favorably to the best ensemble approaches such as HIVE-COTE when compared using the typical time series benchmark data sets from the UCR/UEA archive.

Process of classifying time series data

For time series classification in a supervised setting where all of the data has class labels, the data set is typically split into three sets of data: the training set, the holdout or validation set, and the test set. The training set is used initially to set the parameters of the algorithms that are chosen to attack the problem. The validation set is then used to choose which algorithm performs the best. Finally, the chosen algorithm is used to score the test set and determine the quality of the classifier. Because time series data is temporally ordered, this data property must be used by a good classification algorithm.

Often with real-world data collection, the length of the samples in a time series is different. This can be problematic for some classifiers such as dynamic time warping when a warping window smaller than the difference in sample length is used, or nearest neighbor using Euclidian distance. In the case of varying length samples, multiple techniques are available for scaling the data to equal lengths, adding low-noise prefixes or suffixes to the shorter samples, for example, while at the same time preserving the class of the data sample.

In addition to unequal sample lengths, gaps in data are not handled well by most time series classification algorithms. Null values or not-a-number (NaN) values should be replaced manually in the data with a value that is based on linear interpolation from the immediately adjacent values.

Finally, it might be advantageous to identify new features that can be added to the data set. These features should be generated based on domain knowledge of the problem together with the objective of the classification work. Such feature engineering could be in the form of data correlations, increasing or decreasing data trends, rolling window mean or weighted average, or the inclusion of exogenous data and features not present in the original data set.

The final step of the process is to use the various models that are created to score the validation data set to choose the best model to use against the test data.

Following the model building phase, a time series classifier is then put to work in analyzing or classifying a time series sequence that it has not been trained on. This scoring phase can be done in real time on streaming data or it can be done in a batch phase against a fully collected data set. In either case, the quality and accuracy of the model is determined by the type and quality of the training data and how closely it matches the data being tested. No single algorithm always creates the best results, which is to say that there is still a lot of data science tradecraft involved in formulating a time series classifier.

The statistical metrics of precision, recall, and accuracy have been traditionally used to gauge the quality of a classifier's performance. In some cases, such as healthcare, false negative results can be deadly, so the quality of a classifier in such environments is often measured by using an ROC curve (receiver operating characteristic). The ROC curve for the classifier is generated by plotting the true positive rate (TPR or sensitivity) against the false positive rate (FPR or specificity). The quality of the classifier using this method is the area under the ROC curve or the AUC (area under the ROC curve). A measure of 1.0 for AUC signifies perfect classification, while a value of 0.5 means that the classifi
previous quality metrics can be use

Use cases for time series classification

Failure prediction (predictive maintenance or power grid fluctuation) is a broad class of use cases that can benefit from the application of time series classification. Typically, the classifier is trained to discriminate between two classes (failure or nonfailure) based on the features of the time series at the current time. An example of this might be classification of the acoustic signature of a bearing in a piece of heavy equipment.

Anomaly detection (network intrusion or bank fraud) is a heavily investigated field because it can use time series classification to identify changes in the behavior of a monitored time series based on training against time series that are labeled as normal behavior. A classic example is detecting a network intrusion based on network activity that had not been previously observed as normal.

Pattern recognition (ECG, face, or sign language) also constitutes a large class of problems against which time series classification can be applied. A good example of this is the change that occurs in the shape of an ECG signal when diseases such as myocardial infarction or atrial fibrillation manifest. Time series classifiers can be deployed in real time to alert when such patterns are discriminated.

Alert generation (IoT or BPM) can play a large role in the management of many modern businesses that use alerts to identify potential loss of manufacturing capability, business transaction capability, or supply chain issues. Many of these problems are prediction problems, but they can often be formulated as a classification problem with abnormal signatures resulting in the alert protocol.

Advanced reading

If you would like to delve further into this topic, this section provides a list of publications for advanced reading.

Time series classification systems

- Weiss, Sholom M., and Casimir A. Kulikowski. Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems. Morgan Kaufmann Publishers Inc., 1991.
- Dean, Jeff, David Patterson, and Cliff Young. "A new golden age in computer architecture: Empowering the machine-learning revolution." IEEE Micro 38.2 (2018): 21-29.
- D Patel, S Shrivastava, W Gifford, S Siegel, J Kalagnanam, C Reddy: Smart-ML: A System for Machine Learning Model Exploration using Pipeline Graph, IEEE BigData 2020

Time series classification applications

- Susto, Gian Antonio, Angelo Cenedese, and Matteo Terzi. "Time-series classification methods: Review and applications to power systems data." Big data application in power systems (2018): 179-220.
- Fawaz, Hassan Ismail, et al. "Deep learning for time series classification: a review." Data mining and knowledge discovery 33.4 (2019): 917-963.
- Oleg A. Prokopyev, and Panos M. Pardalos. "Electroencephalogram (EEG) time series classification: Appli

- L. Vu et al., "A Large-Scale System for Real-Time Glucose Monitoring," 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2018, pp. 34-37, doi: 10.1109/DSN-W.2018.00020. Time series classification methods
- Zheng, Yi, et al. "Time series classification using multi-channels deep convolutional neural networks." International conference on web-age information management. Springer, Cham, 2014.
- Wang, Zhiguang, Weizhong Yan, and Tim Oates. "Time series classification from scratch with deep neural networks: A strong baseline." 2017 International joint conference on neural networks (IJCNN). IEEE, 2017.
- Abanda, Amaia, Usue Mori, and Jose A. Lozano. "A review on distance based time series classification." Data Mining and Knowledge Discovery 33.2 (2019): 378-412.

Summary

Time series data is ubiquitous throughout the world from both the activities of humans to collection of historical data in the natural world. This learning path provides an overview of time series classification, the process used for building a time series classifier, a sampling of the large variety of algorithms available to build time series classifiers, and a survey of typical use cases from the healthcare space.