Time-series Interpretability

Borbély Bernárd

November 2024

1 Motivation and Goals

As the use of artificial intelligence becomes more and more popular, a demand appears to provide an explanation of the prediction.

It becomes important that our models not only achieve good results, but also provide an explanation using the input as information.

While in the **computer vision** and **natural language processing** domains using Transformers [**Transformers**] gives a simple tool for providing explanation, with the attention maps; in the domain of time series, it is not clear what explanation could be well interpretable.

An additional advantage of exploring interpretability methods is that, while we are trying to find easily interpretable information, and as we are trying to build a model which provides this information, we might inadvertently transform in a way that is easier to deal with for our model.

My goal is to explore and compare classical interpretability methods with deep learning-based ones.

Then, inspired by these approaches, I aim to develop my own original method(s) and compare them to the already existing ones.

2 Interpretability in machine learning

2.1 The concept of interpretability

As machine learning tools improve, they make decisions in a more complex way, which is harder for humans to interpret.

There are models that are easily interpretable by themselves, for example, decision trees, where we have access to the questions evaluated in the node, based on which the decision is made. When the models are interpretable by design, they are called **white-box** models. However, in the case of the more complex models, for example, the neural nets and transformers, it is harder to interpret which parts of the input were more important for the prediction and which parts influenced the decision-making. These kind of models are often called **black-box** models. The aim of interpretability of the machine learning models is that the decisions made by the model can be better interpreted by

humans. This interpretability could stem from the model building, so that we use components that provide interpretability. Moreover, it can stem from methods which employ subsequent interpretation, which is designed to provide insight into the functioning of an already trained model.

2.2 An important black-box model: neural nets and their interpretability

The topic of my thesis revolves around neural networks; thus in this subsection I elaborate on the questions connected to interpretability, before I discuss the main topic of the thesis, the interpretability of time series data. Dolgozatunk témája a neuronhálók köré szerveződik, ezért ebben az alfejezetben részletsebben bemutatom az ehhez kapcsolódó interpretálhatósági kérdéseket, mielőtt dolgozat fő témájára, az idősoros modellek interpretálhatóságára térek.

2.2.1 Computer vision

In the world of computer vision, there are various available interpretability tools. The most famous method is the **Grad-CAM** [3], which, after the classification of an instance, calculates the derivative for the most probable class, with respect to the input. This way it can give a *heat-map* which depicts which parts of the input were most significant for predicting this class.

The segmentation of cancerous cells by itself can also employ an interpretability method, since our task is to isolate the sick parts, and this by itself provides evidence for the disease.

3 The interpretability of time series models

Time series, in their simplest form, are sequential records of measurements in a process. Therefore, time series can be used in diverse fields. Time series can be used to describe cardiac rhythm, processes of the stock market, and even the vibrations of a drill bit, which processes differ fundamentally. These are popular and widely used modeling tools, thus, they are popular targets of the machine learning models.

As in every other domain, a demand emerged for the interpretability of time series data over the years.

3.1 Time series and associated modeling concepts

To be able to talk about the interpretability of time series in machine learning models, we first have to define a few basic concepts and understand the nature of time series.
Time series in their simplest form are sequential records of simple observations of a process.

Definition 3.1 (Time series)**.** *The series of ordered recordings $X = \{x_1, x_2, \ldots\}$ is called a time series if x_t was recorded at the specific time $t, t \in T$, where T is the* ***index set*** *of the time series. We require set T to be orderable.*

If T is discrete, it is called a discrete time series, and if it is an interval T (e.g. $[0, 1]$), then it is called a continuous time series. We usually consider the recorded x_t values to be records of a stochastic process, thus the realization of an X_t random variable.

In case of discrete time series, the records are usually taken at regular intervals, however, there are time series in which the records are taken at variable sampling/recording frequency. In the latter case, the time stamp may also be included in the time series.
Accordingly, one of the key properties of the time series is that the records are collected at consecutive time points, either discrete or continuous.

Diverse problems can arise regarding time series. Firstly, the problems of **time series analysis** which includes the exploration and understanding of the underlying processes of the time series. Moreover, explicit **modeling** and **prediction** tasks can also be formulated.

Task 3.1 (Autoregressive prediction task)**.**
Given a time series $X = (x_1, x_2, x_3, \ldots, x_n)$ we are looking for a ***function*** *which if given $x_1, \ldots, x_{k-1} \quad (k \in \mathbb{N}_+)$ pattern, predicts x_k's value or it's distribuion.*

Autoregression means that we infer the value of x_k from previous values. A specialty of time series is that we nearly always assume a connection between the value of x_k and the previous values. The nature of the dependence is specified by the model family, although it is most frequently a linear relationship.

Task 3.2 (Time series analysis)**.**
For a given $X = (x_1, x_2, x_3, \ldots, x_n)$ time series, we aim to choose the appropriate ***model family*** *which generated the time series and to determine the characteristic properties of this family. These properties for example could be seasonal trends, their periods, and their increasing trend.*

These two tasks aren't disjoint. For that we can execute the prediction, we need to model the time series. The time series analysis could help us choose the model family, and only after we choose the model from this family which possesses the best parameter can we make predictions.

I would like to emphasize here that, while choosing the model family behind our time series, we implicitly select the prediction model as well.

In my thesis work, I refer to the time series generating process and also to the machine learning method, which executes the prediction task as a 'model'.

At first glance, these two concepts seem hardly related, however, the aforementioned perspective connects them. Once we select the model family, the fitting of the machine learning method includes the optimization of the parameters of the model family, and there will be one model that we use for the prediction.

3.1.1 Examples and their connection to interpretability

In this subsection I would like to demonstrate the diversity of time series and their connection to the topic of my thesis work. With the diversity of interpretable information I would like to demonstrate the many options for interpretability.

Example 3.1 (Daily temperature)**.** *The most natural discrete time series in the field of meteorology is the records of the temperature.*
TA seasonal trend can be observed in this time series: summers are warmer, winters are cooler.
The trigonometric functions are suitable for effective modeling of these seasonal trends: $X_t = cos(t/5) + Z_t$*, where* $cos(t/5)$ *provides the seasonality.*Z_t *is a random variable with expected value of* 0*, which is intended to account for random fluctuations.*

In the example of 3.1. we observe a periodic trend in the recorded temperatures, namely that in the summers it is hotter, and in the winters it is colder. Our interpretation of periodicity comes from our understanding of seasons, and this periodicity is what's represented by choosing a trigonometric function to be included in the model.

When we are modelling the evolution of a stock on the stock market, we often model it as a continuous time series. However, a stock's value rarely depends on which season we are in, so there is no periodic component, but there might be an increasing one.

Example 3.2 (The evolution of a stock)**.** *The evolution of a stock is often modeled with the family:* $X_t = m_t + Z_t$*, where* m_t *is some monotone increasing component, and* Z_t *is the component for the random fluctuations.*

In the 3.2. example we identify that there is a component which is monotone growing, and approximating m_t is what gives us interpretability. Although in real life a stock's price's evolution is a much more complex process, I wanted to demonstrate where interpretability can come from.

Since the processes of stock-market are more complecated, there are countless approaches and models. The first intuition is that the next value of a stock depends on its current value and a few of its previous values, since the members of the market would rather invest if the currency rate increases, and the currency rate increases further at the stock market
It is more realistic to model it by considering that the previous currency rates

influence the progression of the current currency rate. And with that, we arrived at the autoregressive models, where the current currency rate depends on the previous ones.

Example 3.3 (Linear autoregression model)**.** *Let $X_1, X_2, \ldots$ be random variables and $p \in \mathbb{N}_+$. Let the following relationship hold true between the variables:*

$$X_t = \sum_{i=1}^{p} \varphi_i X_{t-i} + \epsilon_t,$$

where $\varphi_1, \ldots, \varphi_p \in \mathbb{R}$ are the model's variables, and ϵ_t is a white noise variable (with 0 mean, and finite standard deviation). We call this the autoregressive model of order p and we denote it with $AR(p)$.

In this model family, the model fitting is the optimization of the parameters of the family. One method for this is the maximum likelihood and the expectation maximization algorithm, in which the parameter approximation is performed iteratively.

Though a complex model like this can better describe the behavior of time series, it is less clear from the perspective of human interpretability what information would be easily understandable. One approach would be to determine the parameters, but these can't be understood intuitively by humans, hence we should seek an other method for interpretability.

Instead of a linear we could assume a more complex relationship between the variables, and under certain circumstances, there is a use case for the fitting of these model families. Nevertheless, to be able to approximate the parameters of these more complex models, we need increasingly more complex and computationally more demanding algorithms, which substantially slow the fitting of these models.

It is noteworthy to mention some of models based in machine learning.

Example 3.4 (Machine learning based autoregressive models)**.** *The machine learning based autoregressive models are described by:*

$$X_t = f(X_{t-1}, X_{t-2}, \ldots, X_{t-p}) + Z_t,$$

where the function f can be a neural net, or even a Transformer model, and $Z_t \sim N(0, \sigma^2), \sigma < \infty$ is the usual white noise variable.
The parameter p determines how many previous observation affects the current one.

These models are often considered to be black box models, and the model fitting is done by gradient descent-based methods. In case of these complex models, the problem of interpretability is not that clear, therefore, we try to

gain an understanding of the prediction of the fitted model by using numerous interpretability methods. It is noteworthy that modeling with neural nets is well-established in multiple ways. Firstly, the aforementioned neural net-based autoregressive models (**??** could be the generative model that generated the time series. Another reason to use neural nets for modeling time series is that there are theoretical findings indicating that neural networks are good approximators. This means that neural nets can effectively approximate any function given that the neural net can have an infinite width and there is an infinite amount of data.

3.1.2 Event Sequences as Progressions of Snapshots of Time Series

From a time series, especially a continuous time series, one can create a new one by subsampling at given time points. In real-world monitoring, it is often not feasible to perform continuous sampling; therefore, most available time series are actually discretely sampled versions of continuous processes.

In some cases, the exact value of a measurement might not matter, or we may not be able to measure it precisely. As a result, the domain of the variable X_t may not be continuous, but rather a discrete space. In modeling such processes, we assume the system is in some state, and the next observed state depends on the current state and possibly a few preceding ones. If the next state only depends on a few fixed numbers of previous states, we get some kind of Markovian process.

In this section, I just present some ideas, and we define them more rigorously in later chapters.

We call these model event sequences, or event-based modeling, and in my thesis, they play a central role.

The main challange in this area is to fint well interpretable information.

3.2 Attitudes Toward Interpretability in Time-Series Machine Learning Models

In this section, I explore different time series interpretability methods based on the workings of Theissler et. al [4]. Since time series data can be highly diverse, and it is not always obvious which pieces of information are easily interpretable, interpretability methods can have a wide variety. In this chapter, I review various approaches — some of which were already introduced in 2. — and propose a classification of these methods based on several different perspectives.

3.2.1 Ante-hoc and Post-hoc Methods

Interpretability can appear during model construction or through the analysis of an already trained model. The former is referred to as ante-hoc, while the latter is known as post-hoc interpretation.

Ante-hoc models are inherently interpretable due to their structure. A good example is decision trees. Post-hoc interpretability methods, on the other hand,

are applied to already trained models and aim to explore what the model has learned.

A well-known *post-hoc* method is the calculation of *SHAP* values.

3.2.2 Global vs. Local Interpretability

Interpretability can also be categorized as global or local.
Global explanations aim to find generalizable rules that apply across the entire dataset. Local explanations focus on how the model makes a prediction for a specific instance x.

3.2.3 Model-Specificity

A third option of classification is between model-specific and model-agnostic methods. Model-agnostic methods can be applied regardless of whether the model is a decision tree, a black-box model, or a regression-based model. These are the **model agnostic** methods.

On the other hand, model-specific methods rely on the internal structure of the model. For example, Grad-CAM can only be used with differentiable models, making it model-specific.

3.2.4 Saliency-Based Approaches

Saliency-based approaches aim to determine which parts of the input are most important for the model's decision. There are several methods using this idea.

One of the most popular method is using **Shap values** [1] It uses Shapley values from game theory to evaluate the contribution of each feature. An important advantage is that SHAP is *model-agnostic*, because it uses the classification function only as a black-box. However, calculating the exact SHAP values is computationally expensive since it would require evaluating all subsets of features—therefore, approximations are usually applied.

There are *attention*-based *ante-hoc* methods, if the attention mechanism appears somewhere in the model. he lengths or magnitudes of the attention vectors can be interpreted as importance weights for the input elements. These methods are *model-specific* because they only work if the attention mechanism is present in the model.

The **gradiens based** methods, like textbfGrad-CAM [3], or **Saliency** calculate the gradient with respect to the input to measure the importance of the features and points. A big drawback in these methods is that to be applicable, the model needs to be differentiable.

Since saliency-based methods identify important input regions, they often require domain knowledge for meaningful interpretation and may remain challenging to understand for an average user.

3.2.5 Subsequence-Based Approaches

An alternative direction for interpretability focuses on identifying important subsequences.

Maletzke et. al. [2] look for so-called **motifs**, these are fixed-length subsequences which appear frequently in the dataset. They use these *motifs* for classification, with the help of a decision tree. Thus, this is a globally interpretable, ante-hoc technique.

Their advantage lies in the fact that classification decisions can be explained by the presence of coherent subsequences. However, they do not achieve state-of-the-art performance, meaning that interpretability comes at the cost of reduced predictive accuracy.

Similar in manner to motifs, **shapelets** were introduced [5]. For shapelets, we do not require them to be subsequences for an instance. Typically in shapelet-based methods, the goal is to find k shapelet, which discriminates the classes well.

After finding the best k shapelet, they use a distance function to transform each instance to a new view. For each shapelet an instance's distance from the shapelet is measured, and these distances are the new representation of the data.

The classification is then executed using the transformed data.

For both motif- and shapelet-based methods, selecting the optimal subsequence length is a major challenge, as it significantly impacts classification performance.

3.2.6 Decomposition of Event Sequences

In this thesis, I distinguish a specific class of time series called event sequences. In such cases, we assume that the observed sequence consists of interleaved states from multiple underlying generative processes.

A possible interpretation of an event sequence is to identify which subsequences originate from the same generative process. The advantage of this approach is that the resulting subsequences may become interpretable individually.

However, its drawback is that not all time series can be reasonably assumed to consist of multiple interleaved processes—thus, this approach loses generality.

Event sequences and their interpretability are further elaborated on in 4.

4 Decomposition of Event Sequences

4.1 General Concepts

The decomposition of event sequences is a fascinating topic because, if we can successfully divide the sequence into smaller subsequences, each may become interpretable on its own. An interesting example of this is the tracking of a patient's medical history.

Imagine that a patient regularly visits their general practitioner for blood pressure issues, and in the meantime, they contract the flu. Therefore, visits and prescriptions related to both conditions are recorded consecutively in their medical history, despite being unrelated

If the doctor later wants to analyze the progression of the patient's blood pressure condition, they would isolate only the relevant entries. In other words, they would extract a relevant subsequence from the full medical history, which then enables interpretation.

Similarly, when a historian analyzes the Roman Empire, they pick relevant records from historical sources to explain the decisions of the ruling emperor. In both cases, interpretability is achieved by focusing on a relevant subsequence of a broader event sequence.

This gives us the idea that by segmenting an event sequence into meaningful subsequences, the overall sequence can become more interpretable.

4.2 Mathematical Model

To discuss the decomposition of event sequences, we need a mathematical model that describes how the elements of the sequence are generated.

The choice of the generative model influences how we approach the problem of desomposition.

While more complex models may describe the sequences more accurately, they may also complicate the decomposition process.

I aim to use a model in which the decomposition itself serves as the interpretability tool. With this goal in mind, I explore the mathematical toolbox for appropriate modeling approaches.

4.3 A Specific Case: Markov Mixtures

We want to introduce randomness into the generation since we aim to capture non-deterministic processes. A simple yet well-understood class of probabilistic models are Markov chains and their variants. These allow us to model random but structured processes.

To model that our event sequence is the interleaving of multiple evolving processes, we need multiple Markov-chains, which will generate the event sequence, based on some distribution. Let's denote them by $M_1, \ldots, M_k$.

Before exploring different approaches in the literature, let us first define the generative procedure:

There is a main controller which selects the next Markov-chain M_i, which will make a "step" (performs a transition). The label of the new state will be the next character in the event sequence.

One can distinguish two versions of the model based on how the Markov chain is selected:.

- In the simpler **probabilistic** version, each M_i is selected with probability p_i, where $\sum p_i = 1$.

- In the more sophisticated, **chain-dependent** model, the next n is selected based on the currently active chain, using transition probabilities p_{ij}.

Another modeling decision concerns the relationship between the state spaces of the Markov chains. These can be:

- pairwise disjoint
- Overlapping (sharing some states between chains).

In most of our experiments, we use disjoint state spaces and the simpler probabilistic controller model. However, for future research and deeper understanding, it's worth considering the more general models as well.

4.4 Methods for Decomposing Markovian Event Sequences

Once we choose our generative model, our goal is to provide interpretability by identifying those subsequences that originated from the same Markov chain.

This task includes determining both the states of each Markov-chain, as well as the probabilities of choosing each Markov-chain in the mixture model.

In the **probabilistic** model this means the values of each p_i, and for **chain-dependent** this means each p_{ij}

It is noteworthy, that in the case of disjoint state spaces, the task reduces to partitioning the space of all possible states, because once we have the state space for a Markov-chain, we can approximate the transition probabilies of the chain, based on the subsequence of the relevant states.

5 Summary

In my thesis work, I examined the questions of interpretability in the case of time series modeling. I presented the potential approaches and a possible way of their classification. I elucidated the questions surrounding the interpretation of event-like time series data. I demonstrated the classical methods known in literature.
Further research could focus on creating my own method, focusing on event sequence decomposition.

Irodalom

[1] Scott M. Lundberg and Su-In Lee. "A unified approach to interpreting model predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 4768–4777. ISBN: 9781510860964.

[2] André Maletzke et al. "Time Series Classification using Motifs and Characteristics Extraction: A Case Study on ECG Databases". In: Oct. 2013. ISBN: 978-90-78677-86-4. DOI: 10.2991/.2013.40.

[3] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* 128 (2016), pp. 336–359. URL: https://api.semanticscholar.org/CorpusID:15019293.

[4] Andreas Theissler et al. "Explainable AI for Time Series Classification: A Review, Taxonomy and Research Directions". In: *IEEE Access* 10 (2022), pp. 100700–100724. DOI: 10.1109/ACCESS.2022.3207765.

[5] Lexiang Ye and Eamonn Keogh. "Time series shapelets: a new primitive for data mining". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. Paris, France: Association for Computing Machinery, 2009, pp. 947–956. ISBN: 9781605584959. DOI: 10.1145/1557019.1557122. URL: https://doi.org/10.1145/1557019.1557122.