

0.1 Choosing the Prediction Model and Feasibility Model

Before testing any model we have to establish two crucial components of the viability measure. First, we require a prediction model which we want to explain using counterfactuals. This is relevant for determining the improvement that a counterfactual yields in contrast to the factual. Second, we need to know to what extent any given counterfactual is feasible given the dataset at hand. Therefore, we will dedicate the first set of experiments to establishing these components.

0.1.1 Prediction Model

We use counterfactuals primarily to explain predictive models. This explanation requires a to define the prediction model we use in this thesis.

Model Description

As explained in ??, we use a Long Short-Term Memory (LSTM). The architecture of the model is shown in Figure 1. We encode embed each activity in the sequence to an [n]-dimensional space. We also create positional embeddings. Then we concat activity embedding, positional embedding and the event attribute representation to a final vector representation for the event that occurred. Afterwards, we pass the embedding through a LSTM module. We use the output of the last step to predict the outcome of a sequence using a fully connected neural network layer.

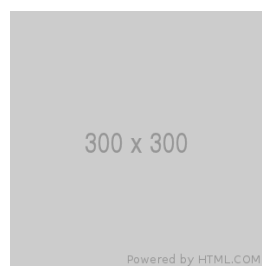


Figure 1: Shows the different components of the LSTM architecture.

Results

[Show how this model is fine to use by reporting training and validation scores.]

Discussion

0.1.2 Feasibility Model

To compute the viability of a counterfactual we need to determine its feasibility. In other words, we have to determine the possibility or impossibility of the counterfactual. We can use the data log to gauge the feasibility, by estimating the data distribution.

Model Description

There are many ways to estimate the density of a data set. For our purposes, we incorporate the sequential structure of the log data and make simplifying assumptions. First, we consider every activity as a state in the case. Second, each state is only dependent on its immediate predecessor and neither on future nor on any any states prior to its immediate predecessor. Third, the collection of attributes within an event depend on the activity which emits it. The second assumption is commonly known as *Markov Assumption*. With these assumptions in place, we can model the distribution by knowing the state transition probability and the density to emit a collection of event attributes given the activity. The probability distributions are shown in Equation 1.

$$some formula = 4 * 2 \tag{1}$$

$$another formula = 1 * 3 * 3 * 7 \tag{2}$$

Hence, the probability of a particular sequence is the product of the transition probability multiplied with state emission probability for each step.

Results

[Show how this model is fine to use by testing whether the sampling distribution matches the observed data distribution.]

Discussion

It is important to note that the proposed way of estimating the data distribution is one of many. The markovian approach explicitly removes the effect of past and future states. It is needless to say, a process step does not have to depend on its immediate previous state. A process outcome may be influenced by all past or future events. For instance, if one has to approve a loan in a second stage, one might be more inclined to approve something

that a trusted employee already approved. Likewise, one might apply more scrutiny, knowing that a certain supervisor is going to approve the third stage.

Furthermore, this approach assumes strictly sequential processes. If the sequence has events running in parallel, we also have to record in greater detail which event has triggered a subsequent event in a given sequence. Often this knowledge is not even available.

[Discuss issue with underflow]

0.1.3 Discussion