0.1 Generating Counterfactuals

The topic of counterfactual generation as explanation method was introduced by wachter$_{C}ounterfactualExplanationsOpening_{2}018inwachter_{C}ounterfactualExplanationsOpening_{2}018[?].Th$

A newer approach by dandl$_{M}ultiObjectiveCounterfactualExplanations_{2}020incorporatesfourmaincriter$
$objectivefitnessfunction[?].Thisapproachstronglydiffersfromgradient-basedmethods, asitdoesnotrequire$

0.2 Generating Counterfactual Sequences

When it comes to sequential data most researchers work on ways to generate counterfactuals for natural language. This often entails generating univariate discrete counterfactuals with the use of Deep Learning techniques. martens$_{E}xplainingdatadrivendocument_{2}014an$

Instead, robeer$_{G}eneratingRealisticNatural_{2}021showedthatitispossibletogeneraterealisticcounterfactual$
$decoderpatterninwhichtheencoderencodesadatainstanceintoalatentvector, whichwillbepeturbedandthendecod$

Even though, a single latent vector model can theoretically produce multivariate sequences, it may still be too restrictive to capture the combinatorial space of multivariate sequences. Hence, most of the models within Natural Language Processing (NLP) were not used to produce a sequence of vectors, but a sequence of discrete symbols. For process instances, we can assume a causal relation between state vectors in a sequential latent space. We call models that capture a sequential latent state-space, which has causal relations, *dynamic*[**?**]. Early models of this type of dynamic latent state-space models are the well-known *Kalman-Filter* for continuous states and Hidden Markov Model (HMM) for discrete states. In recent literature, many techniques use Deep Learning to model complex state-spaces. The first models of this type were developed by krishnan$_{S}tructuredInferenceNetworks_{2}017[?, ?].TheirDeepKalmanFilter(DKF)andsubsequentDeepMark$
$spacebymodellingthelatentspacegiventhedatasequenceandallpreviouslatentvectorsinthesequence.Therearem$
$Bound(ELBO)oftheposteriorforthecurrent$Z$_t$ given all previous $\{Z_{t-1}, \ldots, Z_1\}$ and X_t[**?**].

0.3 Generating Counterfactual Time-Series

Within the *multivariate time-series* literature two recent approaches yield ideas worth discussing.

First, delaney$_{I}nstanceBasedCounterfactualExplanations_{2}021introduceacase-$
$basedreasoningtogeneratecounterfactuals[?].Theirmethodusesexistingcounterfactualinstances, orprototype$
$basedapproachesstronglydependontherepresentativenessoftheprototypes[?, p.192].Inotherwords, ifthemodeld$
$basedtechniqueswillfailtoprovideviablecounterfactuals.Thelikelihoodofsuchabreak-$
$downincreasesduetothecombinatorialexplosionofpossiblebehavioursifthetrueprocessmodelhascyclesorconti$
$basedapproachesmayactasavaluablebaselineagainstothersophisticatedapproaches.$

The second paper within the multivariate time series field by ates$_{C}ounterfactualExplanationsMultivariate$
$basedapproach[?].However, itcontrastsfromotherapproaches, asitdoesnotspecifyaparticularmodelbutpropose$

0.4 Generating Counterfactuals for Business Processes

So far, none of the techniques have been applied to process data.

Within Process Mining (PM), Causal Inference has long been used to analyse and model business processes. Mainly, due to the causal relationships underlying each process. However, early work has often attempted to incorporate domain-knowledge about the causality of processes in order to improve the process model itself[?,?,?,?]. Among these, narendra$_{C}ounterfactualReasoningProcess_{2}019approachisoneofthefirsttoinclu$

Within the XAI context, tsirtsis$_{C}ounterfactualExplanationsSequential_{2}021developthefirstexplanationm$ $learningorSARSA.However, thisoftenrequiresadditionalassumptionssuchasagivenrewardfunctionandana$ $space.Forcounterfactualsequencegeneration, thereisnoobviouschoicefortherewardfunctionortheaction-space.$

Nonetheless, both tsirtsis$_{C}ounterfactualExplanationsSequential_{2}021andoberst_{C}ounterfactualOffPolic$

For this reason, this thesis highlights some key differences and similarities. However, to understand the differences and similarities, we first have to establish some core concepts. In this section, we only discuss their approach, briefly.

The authors recognised that some processes have critical events which govern the overall outcome. Hence, by simply avoiding the undesired outcome from critical event to critical event, it is possible to limit the search space and compute viable counterfactuals. They use an extension of DiCE[?] to generate counterfactuals. However, their approach requires concrete knowledge about these critical points. We propose a Framework that avoids this constraint.

To our knowledge, the authors are also the first authors that try to optimize their counterfactual process generation based on criteria that ensure their viability. However, in our approach, we use different operationalisations to quantify the criteria.