Architecture Learning of Deep Neural Networks for Counterfactual Inference

Author One, Author Two, Author Three
Authors Details
Authors Email

Abstract

We propose a novel approach for automatically inferring appropriate architectures of deep neural networks for the task of counterfactual inference over observational data. The individualised causal effect of an intervention or treatment is modelled in terms of a multi-task learning problem using a deep neural network which consists of a number of layers that are shared among the factual and counterfactual outcomes and a number of outcome-specific layers. Our approach enables automatically selecting an appropriate architecture (i.e. number of shared and outcome-specific layers) by exploiting inferred characteristics of the dataset such as the propensity score, the skewness of the data, and the complexity of the different outcome functions. We conduct experiments on a synthetic dataset allowing us to parametrize and fully control the characteristics of the data before applying our approach to a real-world observational study for which we infer the characteristics in order to derive an appropriate architecture. As shown in the experiments, our method outperforms the state-of-the-art.

Introduction

The technological advancements of recent years have resulted in an increasing availability of data in various fields such as healthcare, education, and economics. This data can be used to make predictions concerning unseen data points on the basis of statistical models. When dealing with observational studies, we are often particularly interested in the task of predicting the individualised treatment effect that certain intervention has on a given subject or context. In the case of electronic health records, for instance, a dataset typically consists of a set of patients each with individual features, a treatment assignment indicator (i.e. whether or not they received the treatment), and an observed outcome which we call the *factual outcome*. The quantity we are interested in is the *counterfactual outcome* (i.e. the outcome had the patient received a different treatment assignment) because it allows us to compute the individualised treatment effect helping us make informed decision during treatment planning.

Classical works have focused on estimating aver- age treatment effects through variants of propensity score matching (Rubin, 2011; Austin, 2011; Abadie Imbens, 2016; Rosenbaum Rubin, 1983; Rubin, 1973). More recent works tackled the problem of estimating indi- vidualized treatment effects using representation learn- ing (Johansson et al., 2016; Shalit et al., 2017), Bayesian inference (Hill, 2012), and standard supervised learning (Wager Athey, 2015).

Recent works have shown that the problem can be effectively framed in terms of a multi-task learning problem using deep neural networks. The network has a set of layers that are shared among both the factual and the counterfactual outcomes and a number of outcome-specific layers. However, the questions of how to select an appropriate architecture (i.e. the number of shared layers, and the number of outcome-specific layers) may drastically influence the expressiveness and computational complexity of the model and remains an open challenge. While there are various general approaches for model selection and architecture learning in neural networks, they do not make use of the specific nature of our causal inference problem.

In this paper, we propose a novel approach for automatically inferring appropriate architectures of deep neural networks for the task of counterfactual inference over observational data. This is achieved by exploiting inferred characteristics of the dataset such as the propensity score, the skewness of the data, and the complexity of the different outcome functions. For instance, if one of the outcomes follows a much more complex function than the other, this should be reflected in an asymmetric architecture which utilises a higher number of outcome-specific layers for the more complex outcome than for the other.

Problem Formulation

We represent each subject i in our population with a d-dimensional feature vector $X_i \in \mathcal{X}$, and two *potential outcomes* $Y_i^{(1)}, Y_i^{(0)} \in \mathbb{R}$ which are drawn from a distribution $(Y_i^{(1)}, Y_i^{(0)}) \mid X_i = x \sim \mathbb{P}(\cdot \mid X_i = x)$. This way, the *individualised treatment effect* for subject i can be expressed as

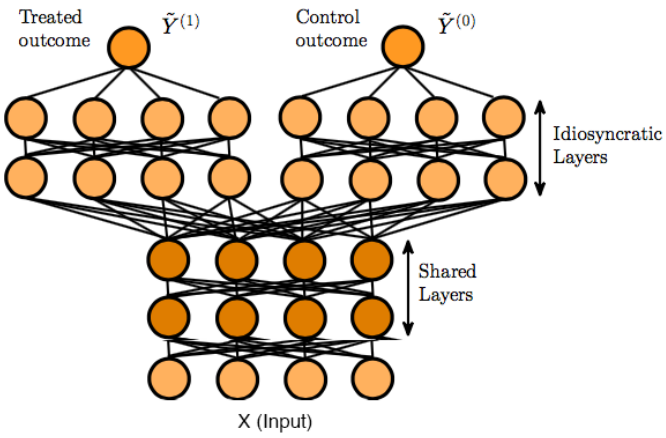

Figure 1: DRAFT: Architecture of a Deep Counterfactual Network (DCN). Objective is to learn appropriate values for number the different number of layers (here $L_s = L_{i,0} = L_{i,1} = 2$).

$$T(x) = \mathbb{E}[Y_i^{(1)} - Y_i^{(0)} \mid X_i = x]. \quad (1)$$

Given this definition, the objective is to approximate the function $T(x)$ using an observational dataset $\mathcal{D}$ consisting of n independent samples. Each sample is comprised of a tuple $\langle X_i, W_i, Y_i^{(W_i)} \rangle$, where X_i represents the subject's features, $W_i \in \{0, 1\}$ the treatment assignment indicator, and $Y_i^{(W_i)}$ and $Y_i^{(1-W_i)}$ the respective *factual* and *counterfactual* outcome. The treatment assignment is a random variable depending on the subjects' features, i.e. $W_i \not\perp X_i$. The assignment reflects a domain-specific policy which can be captured in terms of the probability $p(x) = \mathbb{P}(W_i = 1 \mid X_i = x)$ called the *propensity score*.

We are following the approach of (TODO CITATION) and are using a *deep counterfactual network* (DCN) to infer $T(x)$ from $\mathcal{D}$. The DCN treats the problem as a multi-task learning problem using a deep neural network with an architecture illustrated in figure **??**. The network uses a number L_s of shared layers, a number $L_{i,0}$ of idiosyncratic layers for the *treated outcome*, and a number $L_{i,1}$ for the *control outcome*. We are interested in learning an appropriate architecture of the DCN, in particular, coming up with suitable values for L_s, $L_{i,0}$, and $L_{i,1}$.

Architecture Learning

* How does architecture learning work in general?
* What are the main existing methods for this?

* What is our approach?
* Hyper-parameter search that is informed by propensity score and different metrics of the data

Relevant Characteristics

Briefly describe ...

(a) Skewness * What do we mean by that?
Depending on propensity score.
* How do we get that the PS?
* How should it be reflected in the layers?
* How does it influence the architecture?
* Qualitatively? Quantitatively?
* out1 / total, out2 / total

(b) Complexity of Response Surfaces * What do we mean by that?
* What does it depend on?
* How do we measure it?
* How should it be reflected in the layers?
* How does it influence the architecture?
* Qualitatively? Quantitatively?
* shared / total or out1 / total, out2 / total ?

(c) Interdependence between Features * What do we mean by that?
* What does it depend on?
* How do we measure it?
* How should it be reflected in the layers?
* How does it influence the architecture?
* Qualitatively? Quantitatively?
* shared / total

Deriving a suitable Architecture for the DCN

* Once we are able to quantify these characteristics, what do we do with it?
* What is the relationship between these characteristics and an appropriate architecture?

Experiments

The experiments are conducted on two different datasets. Firstly, we use a synthetic model which allows us to parametrise and fully control the characteristics mentioned in the previous section. This gives us the power to investigate how the different characteristics influence the performance of the learnt architecture. Secondly, we run the experiments on the *UNOS* dataset (consisting of information regarding patients who underwent an organ transplantation) to show how our approach generalises to a real-world dataset for which we do not have access to the characteristics directly but have to infer them from the data in order to learn a suitable architecture. In both cases, we compare the performance of a DCN whose architecture was learnt by our approach to a generic DCN and a number of other baseline approaches and architectures.

As we are dealing with counterfactual inference, we generally do not have access to the ground truth counterfactual outcomes, making it difficult to evaluate the performance of our predictions on real-world data. As a consequence, we adopt a semi-synthetic experimental setup (Hill, 2012; Johansson et al., 2016) for which we use the original covariates and treatment assignments but simulate the

outcomes according to a specific response surface described in detail below.

Since it is our objective to learn an appropriate architecture by exploiting characteristics of the data, the number of (shared and outcome-specific) layers varies across the different datasets. However, throughout both networks we are using a fully-connected architecture with 200 hidden units in all layers (ReLU activation) and we evaluate the performance in terms of the mean squared error (MSE) of the estimated treatment effect. The datasets are split into a training set (80%) and test set (20%) and evaluated exclusively on the test set, averaging over 100 experiments for which new outcomes are drawn each time.

Synthetic Model

Data Generation For the synthetic model, we draw $n = 1000$ samples in the form of a tuple $\langle X_i, W_i, Y_i \rangle$ for each subject i. The tuple $X_i = (x_{i,0}, x_{i,1}, \ldots, x_{i,24})$ consists of $d = 25$ covariates for each subject which are independently drawn from a uniform distribution, i.e.

$$x_{i,0}, x_{i,1}, \ldots, x_{i,24} \overset{iid}{\sim} \mathcal{U}(0,1)$$

keeping each covariate strictly non-negative. For the treatment assignment indicator $W_i \in \{0,1\}$, we define

$$\tilde{p}(X_i) = \frac{1}{1 + exp(-\alpha \sum_{x_j \in X_i} x_j)}$$

$$W_i \sim Bernoulli(\tilde{p}(X_i))$$

where $\tilde{p}(X_i)$ represents the subject's propensity score and the parameter $\alpha \in \mathbb{R}^0$ gives us a way to control the imbalance between the treated and untreated subjects. For $\alpha = 0$, we get $\tilde{p}(X_i) = 0.5$ corresponding to a maximum balance between number of treated and untreated subjects whereas an increasing alpha shifts the distribution towards a higher proportion of treated subjects. We compute both outcomes $Y^{(0)}, Y^{(1)} \in \mathbb{R}$ as

$$Y_i^{(0)} = f_0(X_i) + \mathcal{N}(0,1)$$

$$Y_i^{(1)} = f_1(X_i) + \mathcal{N}(0,1)$$

which are governed by their corresponding outcome functions f_0 and f_1 defined as

$$f_1(X_i) = \sum_{x_j \in X_i} \lambda_i^{(1)} x_j + \beta^{(1)} \cdot exp(\sum_{x_j \in X_i} \mu_i^{(1)} x_j^2)$$

$$f_0(X_i) = \sum_{x_j \in X_i} \lambda_i^{(0)} x_j + \beta^{(0)} \cdot exp(\sum_{x_j \in X_i} \mu_i^{(0)} x_j).$$

Each outcome function consists of a linear part governed by the coefficients in the vectors $\lambda^{(0)}$ and $\lambda^{(1)}$ respectively, and an outcome-specific polynomial part inside the exponential function governed by the coefficients in the vectors $\mu^{(0)}$ and $\mu^{(1)}$. In the case of f_1 this is a quadratic function whereas for f_0 we are using a linear function. The parameters $\beta^{(0)}, \beta^{(1)} \in \mathbb{R}^0$ let us control the weight of the outcome-specific polynomial part in comparison to the common linear part.

The vectors $\lambda^{(1)}, \mu^{(1)}$ for the treated outcome function f_1 are drawn based on the data generation process designated as the "Response Surface B" setting in (Hill, 2012). The vectors $\lambda^{(0)}, \mu^{(0)}$ of the untreated outcome function f_0 we define as

$$\lambda_i^{(0)} \sim \mathcal{N}(\lambda_i^{(1)}, \sigma) \qquad \mu_i^{(0)} \sim \mathcal{N}(\mu_i^{(1)}, \sigma)$$

Finally, we set $Y_i = W_i \cdot Y_i^{(1)} + (1 - W_i) \cdot Y_i^{(0)}$, henceforth called our factual outcome. The other (i.e. counterfactual) outcome is not used in the training set but needed later for evaluation purposes.

In summary, we receive a synthetic model which is governed by the parameters α, $\beta^{(0)}$, $\beta^{(1)}$, and γ, each corresponding to a different characteristic we are interested in. * Alpha defines skewness
* beta0, beta1 defines how much the weight we put on the different parts of the outcome surfaces
* gamma defines the difference between the two surfaces (leading to a generally more complex model)

* Want to have control over all parameters
* Skewness, Propensity Score, General Complexity, Complexity for different outcomes
* How did we generate it

Results and Discussion * Describe settings: Number of samples, experiments, hyper-params, etc.
* Show result table

UNOS Dataset

Real-world dataset to illustrate how our approach generalises

About the dataset * Who is the provider?
* What is it about?
* Some statistics: Samples, Features, etc.
* What is the treatment assignment, what the outcome?

Results and Discussion * Describe settings: Number of samples, experiments, hyper-params, etc.
* Show result table * Details of experiment * Show graph
* Explore the different parameters * Synthetic Model and Mapping it to real dataset

Conclusions and Future Research

Counterfactual inference over observational data is of great importance in various areas such as healthcare, education, and economics. Deep neural networks are highly suitable for the task and represent the state-of-the art as they are able to capture complex relations in the outcome surfaces. However, it remains and open challenge of how to select an appropriate architecture for models. This is true in particular in the case of *deep counterfactual networks* which treat

the problem as a multi-task learning problem with different numbers of shared and outcome-specific layers.

Our approach addresses this issue of model selection and provides an effective way to automatically learn a suitable architecture by inferring relevant characteristics from the data and incorporating them into the model selection. As shown in the experiments, our approach outperforms the state-of-the-art.

References

Test Test