Neural Causal Models with TRAM-DAGs: A framework for modelling causal effects in a flexible and interpretable way and for making subsequent causal queries.

Master Thesis in Biostatistics (STA495)

by

Mike Krähenbühl

Matriculation number: 18-652-149

supervised by

Prof. Dr. Beate Sick

Prof. Dr. Oliver Dürr, HTWG Konstanz

Zurich, July 2025

Neural Causal Models with TRAM-DAGs:

Applied on real-world data and used for ITE estimation.

Mike Krähenbühl

Version June 5, 2025

Contents

Preface

In the introduction part, my aim is to give a summary of important concepts of causal inference and causal models. Further, I motivate the importance for methods that allow to draw causal conclusions from observational data in contrast to randomized controlled trials (RCTs) and that the proposed framework of tram-dags (cite sick duerr) can be used as such a tool.

In the methods section, I give a detailed description of the tram-dag framework and how it works by illustrating it on a simple simulated example. I also show for what kind of causal queries the model can be used for. Although it is not a topic for observations data, I discuss how the model can be used for estimating the individualized treatment effect (ITE).

In the results section, I will first show results from simulation studies where the ground truth is known. Second, I will show results on a real-world example in the setting of climate data. Third, I present the results of the ITE estimation.

In the final section, I will discuss the results and give a conclusion about the advantages and limitation of this framework and provide an outlook.

Mike Krähenbühl
July 2025

Chapter 1

Introduction

The important questions that studies want to answer are usually not associational but causal (Pearl, 2009a). For example, these are questions that ask for effects when making a certain intervention, like the effect of a treatment. They can ask for reasons that lead to the observed outcome, like which disease caused the given symptoms. Or what would have been different if another action was taken, like what would the gdp have benn, if the interest rates were incereased only by 25 instead of 75 Bps. To answer such questions, a causal reasoning must be applied that aims to understand the underlying data genearting mechanism, sole associational reasoning directly from the data is not sufficient.

Importance of Causal Inference on Observational Data

The gold standard to measure causal relationships between an intervention and an outcome is the randomized controlled trial (RCT) (Hariton and Locascio, 2018). The key concept of this prospective study design, is that the participants are randomly allocated due either the treatment or control group. Due to this randomization, the influence of potential confounding variables is eliminated and study groups are balanced with respect to baseline characteristics allowing for an unbiased cause-effect estimation. Disadvantages of RCTs include but are not limited to often high cost, the time for planning and executing the trial, and generalisability to the population of interest. Furthermore, RCTs typically aim to estimate an average treatment effect on a sample, which is the difference in the averages accross the treatment groups (Nichols, 2007). However, patients have individual responses to the treatment, depending on their characteristics. In personalized medicine, such individual treatment effects are crucial. Another central limitation of RCTs is that in many scenarios they can not be conducted due to ethical or practical reasons. For example, an RCT is only ethical in the case of clinical equipoise, which means that there is uncertainty about the (superiority) of one of the two treatment arms (Freedman, 1987). It is not acceptable to treat one group with the assumed inferior treatment. The same is true for obviously harmful interventions, like smoking or drinking alcohol. In these cases, it is not possible to conduct an RCT to estimate the causal effect of smoking on lung cancer.

Therefore, much of the research aims to make causal inference from observational data in a non-experimental or quasi-experimental design. In an observational setting, there are usually confounding variables that make it challenging to measure the effect between exposure and outcome. Methods for causal infreence on observational data for example aims to correctly adjust or control for confounders. Sick and Dürr (2025) proposed the framework of TRAM-DAGs to estimate the causal relationships in an observational setting and make subsequent queries. The aim of this thesis is to further analyze this method and apply it in a real-world scenario.

General Causality review (DAG, SCM, Pearls ladder, rubins rules?)

Causal relationships can be represented by a directed acyclic graph (DAG) as, for example, shown in Figure 1.1(a). The variables, or nodes, are connected by directed edges which indicate the path of causal dependence.

Usually we want to answer questions that can be assigned to one of the groups in pearls hierarchy of causation Pearl (2009b). Visual examples are presented in Figure 1.1(a)-(c). Level 1 are observational queries which are conditional probabilities $P(Y \mid X)$ and can be answered directly from the joint distribution $P(Y \cap X)$. Level 2 are interventional queries which are probabilities $P(Y \mid do(X))$ that result by a taken action do(X). Where observational queries only require to know the joint distribution, for interventional queries an additional understanding of the causal mechanism is necessary. Level 3, the analysis of counterfactuals, poses the biggest challenge. These are what-if questions. An example of this would be if a sick patient was treated with a certain treatment and then died. Death would therefore be the observed and therefore factual outcome. The counterfactual outcome is the outcome that would have occurred if the patient had received a different medication. Such counterfactual questions are often labelled as metaphysical because they can never be tested directly. However, there are important practical questions that require the analysis of such counterfactuals.

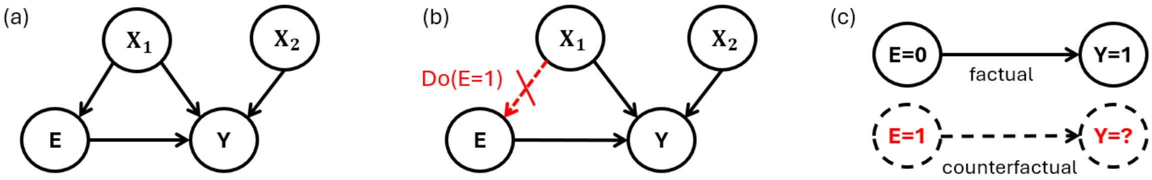

Figure 1.1: Example for the three levels of Pearl's hierarchy of causation. (a) DAG for observational data. (b) DAG when making a do-intervention by fixing the variable E at a certain value. c) Observed factual outcome and corresponding counterfactual query.

To illustrate Pearl's three levels of causality, I consider a simplified example involving the exposure Exercise (E), the outcome Heart Disease (Y), the confounder Age (X_1) and the additional covariate Smoking (X_2). I assume that exercise reduces the risk of heart disease, but both variables are also influenced by age. Figure 1.1(a)-(c) illustrates the corresponding scenarios.

Level 1: Observational ("seeing"). We observe the joint distribution of variables without intervention. Example: What is the probability of heart disease given that a person exercises?

$$P(Y \mid E = 1)$$

This can be estimated directly from data (by filtering for E=1 and calculating the fraction of Y), but does not account for confounding.

Level 2: Interventional ("doing"). We consider the effect of an intervention on the system. Example: What is the probability of heart disease if everyone were made to exercise, regardless of age or smoking?

$$P(Y \mid \text{do}(E = 1))$$

This requires assumptions about the causal structure.

Level 3: Counterfactual ("imagining"). We ask what would have happened under different circumstances, which means imagine an alternative reality. Example: For a person who does not exercise and has heart disease, would they still have heart disease if they had exercised?

$$P(Y_{E=1} \mid E = 0, Y = 1)$$

Here $Y_{E=1}$ is the outcome under the positive exposure. Counterfactual queries require a structural causal model and cannot be answered from data alone.

What is a Structural Causal Model? To answer questions from Pearl's ladder of causation, the concept of DAG's can be extended to structural causal model (SCM). A set of structural equations of the from $x_i = f_i(pa(x_i), z_i)$, $i = 1, ..., n$ forms a structural causal model

(Pearl, 2009b). $pa(x_i)$ are the direct causal parents of X_i and therefore directly determe its value. Z_i are errors that follow exogeneous noise distributions $P(Z_i)$. They can be understood as latent variables that represent unmodeled factors which can not be observed or measured directly. By convention, the Z_i are assumed to be mutually independent. The potentially non-linear function f_i determines the functional form of the parents and the noise that represents the data generating mechanism of the dependent variable X_i. Hence, in a SCM a source node X_j can be represented as $X_j = f_j(z_j)$ since it does not depend on any other variables in the system. Once all components of the structural equations are known (or assumed to be known), it is fully deterministic.

This representation makes it practical to determine interventional distributios and counterfactual queries. This will be discussed in detail in Section XXX.

In this thesis, the focus is not on finding the causal structure of a system. Such a structure can be found by structure finding algorithms, or determided by expert knowledge, for example. Instead, I assume the DAG to be known and focus on the estimation of the functional form of the relationships.

There is a variety of methods that are applied to quantify these structural equations that form the resulting SCM.

The simplest method might be the linear regression which assumes gaussian error terms Z_i and a linear f_i. Similarly other classical statistical methods can be applied they have the advantage of being typically well defined and having interpretable parameters. However, they require to make strong assumptions about the underlying data generating mechanism which can make them susceptible to bias if these assumptions are violated.

Then there is also the possibility to model the relationships with various methods based on neural networks. Because they can be very flexible, they can also model complex underlying distributions with practically no bias. But this comes at the cost of less interpretability. And often they are limited to continuous data.

The framework of TRAM-DAGs proposed by Sick and Dürr (2025) builds a bridge between these two approaches of classical statistical and deep learning methods. It allows us to model the causal relationships with interpretable or fully-flexible parts, depending on what is regarded more important at the moment. The basis of this model is to construct the structural equations as transformation models as introduced by Hothorn *et al.* (2014), which is a flexible distributional regression method. To make them even more customizable, these transformation models (TRAMS) were extended to deep TRAMS (Sick *et al.*, 2021). Applied in a causal context, these deep TRAMS form the framework of TRAM-DAGs that can be fitted on observational data and allow to tackle causal queries on all three levels of Pearl's causal hierarchy. This framework will be explained in detail in the Section XXX.

The primary goal of this thesis is to apply TRAM-DAGs on real world data where the underlying data generating process is unknown.

An application where causal inference is of particular importance is the estimation of individualized treatment effects (ITE). Chen *et al.* (2025) showed that all causal machine learning models that were trained on a train set failed to generalize to a test set.

The estimation of this with the help of TRAM-DAGs constitutes the second objective of this thesis. Although I analyze it in the setting of RCT data and not observational.

Chapter 2

Methods

In this section I will explain the necessary background needed to understand the TRAM-DAGs. Once the framework of tram dags is explained, I will present how the experiments of the simulation, the application on real data and the ITE estimation are conducted.

The goal of TRAM-DAGs is to estimate the structural equations according to the causal order in a given DAG in a flexible and possibly still interpretable way in order to sample observational and interventional distributions and to make counterfactual statements. The estimation requires data and a DAG that describes the causal structure. It must be assumed that there are no hidden confounders. TRAM-DAGs estimate for each variable X_i a transformation function $Z_i = h_i(X_i \mid pa(X_i))$, where Z_i is the noise value and $pa(X_i)$ are the causal parents of X_i. The important part here is that we can rearrange this equation to $X_i = h_i^{-1}(Z_i \mid pa(x_i))$ to get to the structural equation. The transformation functions h are monotonically increasing functions that are a representation of the conditional distribution of X_i on a latent scale. They are based on the idea of transformation models as introduced by Hothorn *et al.* (2014) but were extended to deep trams by Sick *et al.* (2021). In the following sections I review the most important ideas of these methods as they are the essential components of TRAM-DAGs.

2.1 Transformation Models

Transformation models are a flexible distributional regression method for various data types. They can be for example specified as ordinary linear regression, logistic regression or proportional odds logistic regression. But Transformation models further allow to model conditional outcome distributions that do not even need to belong to a known distribution family of distributions by model it in parts flexibly. This reduces the strength of the assumptions that have to be made.

The basic form of transformation models can be described by

$$F(y|\mathbf{x}) = F_Z(h(y \mid \mathbf{x})) = F_Z(h_I(y) - \mathbf{x}^\top \boldsymbol{\beta}) \tag{2.1}$$

, where $F(y|\mathbf{x})$ is the conditional cumulative distribution function of the outcome variable Y given the predictors $\mathbf{x}$. $h(y \mid \mathbf{x})$ is a transformation function that maps the outcome variable y onto the latent scale of Z. F_Z is the cumulative distribution function of a latent variable Z, the so-called inverse-link function that maps $h(y \mid \mathbf{x})$ to probabilities. In this basic version, the transformation function can be split into an intercept part $h_I(y)$ and a linear shift part $\mathbf{x}^\top \boldsymbol{\beta}$, where the vector $\mathbf{x}$ are the predictors and $\boldsymbol{\beta}$ are the corresponding coefficients.

If the latent distribution Z is chosen to be the standard logistic distribution, then the coefficient β_i can be interpreted as log-odds ratios when increasing the predictor x_i by one unit, holding all other predictors unchanged. This means that an increase of one unit in the predictor x_i leads to an increase of the log-odds of the outcome Y by $\boldsymbol{\beta}$. The additive shift of the transformation function means a linear shift on the latent scale (herer log-odds). The following transformation to probabilities by F_Z potentially leads to a non-linear change in the conditional outcome distribution on the original scale. This means not only is the distribution shifted, also its shape can change to some degree based on the covariates. More details about the choice of the latent distribution and the interpretation of the coefficients are provided in the appendix XXX.

For a continuous outcome Y the intercept h_I is represented by a bernstein polynomial, which is a flexible and monotonically increasing function

$$h_I(y) = \frac{1}{M+1} \sum_{k=0}^{M} \vartheta_k \, \mathrm{B}_{k,M}(y) \tag{2.2}$$

, where ϑ_k are the coefficients of the bernstein polynomial and $\mathrm{B}_{k,M}(y)$ are the Bernstein basis polynomials. More details about the technical implementation of the bernstein polynomial in the context of TRAM-DAGs is given in the appendix XXX.

For a discrete outcome Y the intercept h_I is represented by cut-points, which are the thresholds that separate the different levels of the outcome. For example, for a binary outcome Y there is one cut-point and for an ordinal outcome with K levels there are $K-1$ cut-points. The transformation model is given by

$$P(Y \leq y_k \mid \mathbf{X} = \mathbf{x}) = F_Z(\vartheta_k + \mathbf{x}^\top \boldsymbol{\beta}), \quad k = 1, 2, \ldots, K-1 \tag{2.3}$$

A visual representation for a continuous and discrete (ordinal) outcome is provided in Figure 2.1.

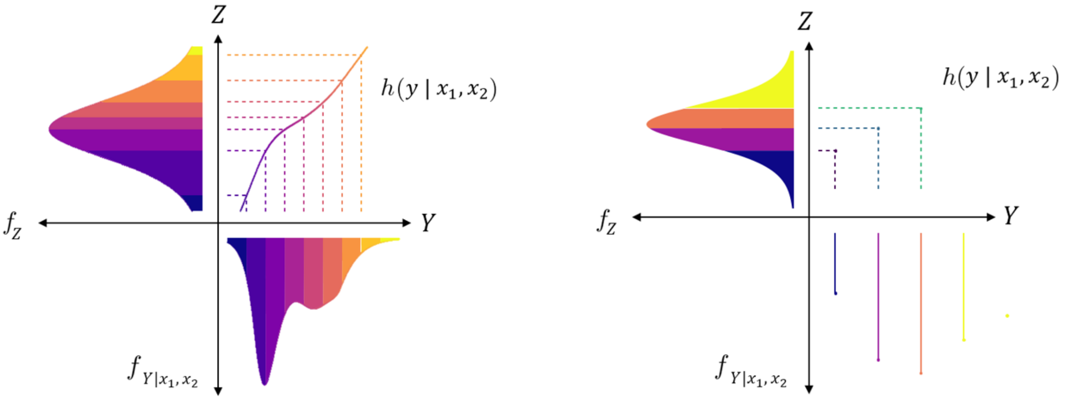

Figure 2.1: Left: Example of a transformation model for a continuous outcome Y with a smooth transformation function. **Right:** Example of a transformation model for an ordinal outcome Y with 5 levels. The transformation function consists of cut-points that separate the probabilities for the levels of the outcome. In both cases the latent distribution Z is the standard logistic and the predictors $\mathbf{x}$ induce a linear (vertical) shift of the transformation function.

To estimate the parameters $\boldsymbol{\beta}$ and $\boldsymbol{\vartheta}$ the negative log likelihood (NLL) is minimized. The NLL is defined as

$$\text{NLL} = -\frac{1}{n} \sum_{i=1}^{n} l_i(\boldsymbol{\beta}, \boldsymbol{\vartheta}) = -\frac{1}{n} \sum_{i=1}^{n} \log(f_{Y|\mathbf{X}=\mathbf{x}}(y_i)) \tag{2.4}$$

where $l_i(\boldsymbol{\beta}, \boldsymbol{\vartheta})$ is the log-likelihood of the i-th observation, $l_i(\boldsymbol{\beta}, \boldsymbol{\vartheta}) = f_{Y|\mathbf{X}=\mathbf{x}}(y_i)$ is the conditional density function of the outcome variable Y given the predictors $\mathbf{x}$ under the current parameterization. I provide the full derivation in the appendix xxx.

For the remainder of this thesis, I rely on the idea of these transformation models to model the conditional distribution functions represented by the transformation functions of the respective variables. The standard logistic distribution is used as F_Z, which results in a logistic transformation model.

2.2 Deep TRAMs

The transformation models as discussed before were extended to deep TRAMs using a modular neural network (Sick *et al.*, 2021). The goal is to get a parametrized transformation function of the form **??**.Each part, the intercept $h_I(X_i)$, the linear shift $\mathbf{x}_L^\top \boldsymbol{\beta}_L$ and the complex shift $f_C(\mathbf{x}_C)$ are assembled by the outputs of the individual neural networks. The user can specify the level of complexity the parents $pa(X_i)$ have on the transformaiton funciton. Figure 2.2 illustrates the case for a SI-LS-CS model.

$$h(y \mid \mathbf{x}_L, \mathbf{x}_C) = h_I(y) + \mathbf{x}_L^\top \boldsymbol{\beta}_L + f_C(\mathbf{x}_C) \tag{2.5}$$

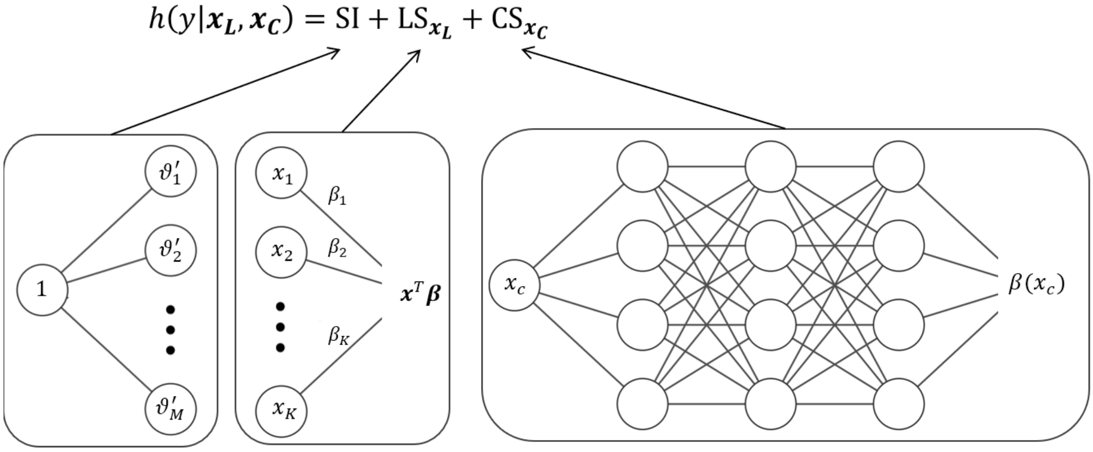
$h(y|\boldsymbol{x_L}, \boldsymbol{x_C}) = \text{SI} + \text{LS}_{\boldsymbol{x_L}} + \text{CS}_{\boldsymbol{x_C}}$
1
ϑ'_1
ϑ'_2
ϑ'_M
x_1
x_2
x_K
β_1
β_2
β_K
$\boldsymbol{x}^T\boldsymbol{\beta}$
x_c
$\beta(x_c)$

Figure 2.2: Modular deep transformation model. The transformation function $h(y \mid \mathbf{x})$ is constructed by the outputs of three neural networks.

Intercept the shape of the transformation function at the baseline configuration $\mathbf{x}_L^\top \boldsymbol{\beta}_L = 0$ and $f_C(\mathbf{x}_C) = 0$ is determined by the intercept $h_I(y)$. For a continuous outcome the intercept is represented by a smooth bernstein polynomial and in the discrete case by cut-points. In either case the parameters ϑ are obtained as output nodes of the neural network. A simple intercept (SI) is the case where the parameters ϑ do not depend on the any explanatory variables. The neural network thereby only takes a constant as input and directly outputs the parameters ϑ. To make the intercept more flexible, the intercept can also depend on the explanatory variables. In this case the complex intercept (CI) models the intercept $\vartheta(x)$ by taking the predictors x as input to a neural network with some hidden layers. This allows the intercept to change with the value of the predictors. Depending on the assumptions, predictors can be used in the complex intercept, or only a subset of them. A detailed explanation of the construction of the bernstein polynomial is given in appendix XXX.

Linear shift If the predictors should have a linear effect on the transformation function, it can be modelled by a linear shift (LS). For this part the neural network without hidden layers and without biases takes the linear predictors $pa(X_i)$ as input and generates a single output node with a linear activation function. This results in the linear combination $\mathbf{x}_L^\top \boldsymbol{\beta}_L$ and it induces a linear vertical shift of the transformation function. The weights $\boldsymbol{\beta}_L$ are the interpretable coefficients of the linear shift. For the logistic transformation model, they are interpreted as log-odds-ratios. The interpretation is further described in the appendix XX.

Complex shift If the transformation function should be allowed to be shifted vertically in a non-linear manner, a complex shift (CS) can be applied. The predictor variables are inputed in a (deep) neural network with at least one hidden layer and a single output node with $f_C(X_C)$ is obtained. With a complex shift, also interactions between predictor variables can be captured.

Level of complexity One practical feature of these modular deep TRAMs is that one can specify, which predictors should have a linear or complex shift effect on the transformation function or that predictors are even allowed to deterimine the shape of the transformation function by a complex intercept. Herzog *et al.* (2023) predicted the ordinal functional outcome three months after stroke by using semi-structured data that included tabular predictors and images. The two data modalities can be included in a single deep TRAM by modeling the part of the images with a CNN.

The estimated distribution function is invariant with respect to the choice of the inverse-link function F_Z (scale of latent distribution) in an unconditional (Hothorn *et al.*, 2018) or fully flexible (CI) setting. However, as soon as restrictions are placed on the influence of the predictors (LS, CS), this leads to assumptions about the scale of the dependency. Which latent distribution should be chosen depends on following factors: (i) the intended complexity of the model, (ii) the assumptions about the data generating process, (iii) the conventional, widely used, scale of interpretation for the specific problem. If the coefficients β in the linear shift term should be interpreted as log odds ratios, then the standard logistic distribution is appropriate. For log hazard ratios it would be the minimum extreme value distribution. There exist plenty of other alternatives.

(The optimal scale could be found by comparing the likelihoods of the model under different latent distributions.)

Parameter estimation The parameters of the neural networks are learned by minimizing the negative log-likelihood (NLL) of the conditional deep TRAM. The learning process is started with a random parameter configuration and the outputs of the neural networks are used to assemble the NLL of the transformation model. The NLL is then iteratively minimized by adjusting the parameters by the Adam optimizer (Kingma and Ba, 2015) until they eventually converge to the optimum state. Additionally, methods to prevent overfitting — such as dropout, early stopping, or batch normalization — can be applied. These techniques are particularly important in more complex networks to ensure that the model generalizes well to out-of-sample data. In the hidden layers, non-linear activation functions such as ReLU or sigmoid are applied.

2.3 TRAM-DAGs

In TRAM-DAGs these deep transformation models are applied in a causal setting. We assume a pre-specified DAG which defines the causal dependence. Then we estimate the distribution of each node by a transformation model that is conditional on its parents. Figrue 2.3 illustrates the basic idea of a TRAM-DAG where a DAG with 3 variables, without hidden confounder, is assumed to be known. The arrows in the DAG indicate the causal dependencies between the variables. The transformation models are constructed by a modular neural network. The assumed influence from the parent variables has to be specified as SI, LS or CS. In this example, X_1 is a continuous source node that acts as parent of X_2 and X_3. For a source node the transformation function only consists of a simple intercept (SI). X_2 is also continuous and its transformation function can be shifted additively (LS) by the value of X_1. X_3 is an ordinal variable with 4 levels and its transformation function depends on the values of X_1 (LS) and X_2 (CS). The cut-points $h(x_3 \mid x_1, x_2)$ represent the cumulative probabilities on the log-odds scale of the first 3 levels of X_3, where the probability of the last level $K = 4$ is the complement of the previous levels k_{1-3}.

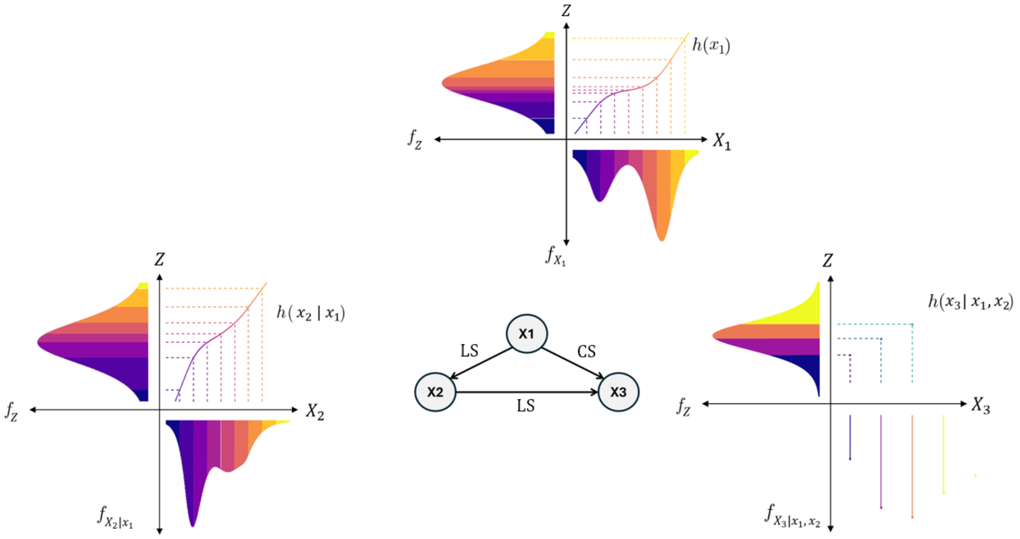

Figure 2.3: Example of a TRAM-DAG with three variables X_1, X_2 and X_3. The transformation functions are represented by the modular neural networks. The arrows indicate the causal dependencies between the variables.

This DAG with the assumed dependencies can be described by an adjacency matrix 2.6, where the rows indicate the source and the columns the target of the effect:

$$\mathbf{MA} = \begin{bmatrix} 0 & \text{LS} & \text{LS} \\ 0 & 0 & \text{CS} \\ 0 & 0 & 0 \end{bmatrix} \tag{2.6}$$

To apply the framework of TRAM-DAGs on this example, we assume to have observational data that follows the structure of the adjacency matrix 2.6. In practice, the DAG is either defined by expert knowledge or by some sort of structure finding algorithm (XXX cite methods). Then we want to estimate the conditional distribution function of each variable by a deep TRAM so that we can sample from the distributions and make causal queries. The conditional distribution functions are given by

$$\begin{aligned} X_1 &\sim F_Z(h_I(x_1)) \\ X_2 &\sim F_Z(h_I(x_2) + \text{LS}_{x_1}) \\ X_3 &\sim F_Z(h_I(x_3) + \text{LS}_{x_1} + \text{CS}_{x_2}) \end{aligned}$$

Construct Modular Neural network

As discussed in the section 2.2, the transformation functions are constructed by a modular neural network. The inputs are the variables in the system as well as the adjacency matrix 2.6 which controls the information flow and assures that only valid connections according to the causal dependence are made. Discrete variables with few categories are dummy encoded, and continuous variables are scaled before feeding them in the neural network. The encoding and the effect of scaling on the interpretation of parameters is discussed in the appendix (ref XXX). The outputs are the three components for the transformation function (SI, LS, CS) for each variable. These components are assembled to the transformation functions. For the complex shift and complex intercept, the structure of the neural network (depth and width) has to be defined. Finally the loss is defined as the negative log likelihood, which the model aims to optimize to estimate the optimal parameterization.

Describe interpretation quickly and refer to formal proof in the Appendix.

Finally, neural network works best if inputs are scaled. Proof that we can do that, it just changes the interpretation. For structure finding algorithms, this might be problematic, because increasing variance along the causal order would be destroyed. (why, how, interpretation change etc. check meeting notes 22.04.2025)

Fitting Betas Interpretable

The two parameters for our linear shift terms are plotted here. We can see that they converge quickly to the same values as we used in the DGP. We can interpret these parameters as log-odds ratios if changing the value of the parent by one unit.

Intercepts

Show the Discrete case with just cutpints (only K-1 parameters of outputs are used) Show the continuous case where the outputs are transformed to monotonically increasing betas for the bernstein polynomial. Also describe Bernstein polynomial construction in detail with scaling and linear extrapolation.

Here I plotted the intercepts of the 3 transformation functions. They also resemble the DGP very nicely.

Linear and complex shifts

Here in the first two plots we can see the linear shifts. And in the right plot we have the complex shift of X2 on X3. The estimated shifts match quite well with the DGP.

Complex shift (Interaction example) to show what is also possible

Here I just want to make a short input from another example. So there the true model was that of a logistic regression with the binary outcome Y and 3 predictors. The binary treatment T and the two continuous predictors X1 and X2. There was also an interaction effect assumed between treatment and X1. So this basically means that the effect of X1 on the outcome is different for the two treatment groups.

And here we can show that our TRAM-DAG specified by a complex shift of T and X1 can also capture this interaction effect quite well.

2.3.1 Sampling from TRAM-DAGs

Observational sampling Once the TRAM-DAG is fitted on data, it can be used to sample from the observational or interventional distribution or to make counterfactual queries. The structural equations $X_i = f(Z_i, \mathrm{pa}(X_i))$ are represented by the inverse of the conditional transformation functions $h^{-1}(Z_i \mid \mathrm{pa}(X_i))$ because $Z_i = h(X_i \mid \mathrm{pa}(X_i))$. The sampling process from the observational distribution for one iteration (one observation of all variables in the DAG) is described in the pseudocode 1 and illustrated in Figure 2.4. The process is repeated for the desired number of samples.

Algorithm 1 Generate a samples from the TRAM-DAG

```
Given: A fitted TRAM-DAG with structural equations $X_i = f(Z_i, \mathrm{pa}(X_i))$, where $Z_i = h(X_i \mid \mathrm{pa}(X_i))$
for each node $X_i$ in topological order do
    Sample latent value $z_i \sim F_{Z_i}$                ▷ e.g., rlogis() in R
    if $X_i$ is continuous then
        Compute $x_i = h^{-1}(z_i \mid \mathrm{pa}(x_i))$ by solving $h(x_i \mid \mathrm{pa}(x_i)) - z_i = 0$
    end if
    if $X_i$ is discrete then
        Determine $x_i$ such that $x_i = \min\{x : z_i \leq h(x \mid \mathrm{pa}(x_i))\}$
    end if
end for
```

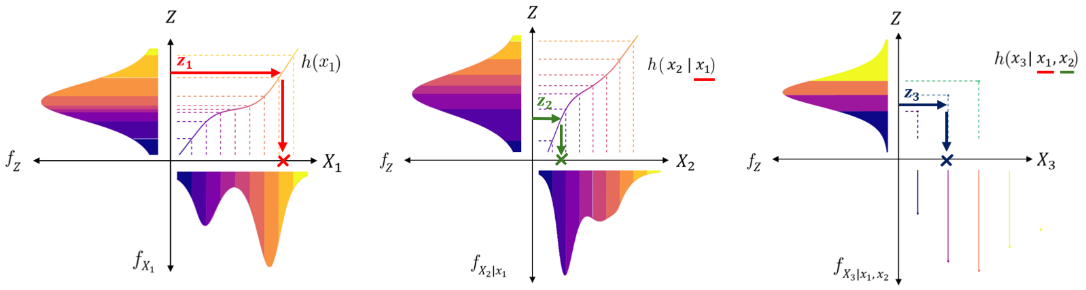

Figure 2.4: One sampling iteration for the three variables from the estimated transformation functions $h(x_i \mid \text{pa}(x_i))$. The latent values z_i are sampled from the standard logistic distribution. The values x_i are determined by applying the inverse of the transformation function for continuous variables or by finding the corresponding category for the ordinal variable.

Interventional sampling To sample from the interventional distribution, we can apply the do-operator as described by Pearl (1995) (Pearl named it set instead of do). The do-operator fixes a variable at a certain value and sample from the distribution of the other variables while keeping the fixed variable constant. For example, if one wants to intervene on X_2 and set it to a specific value α, $\text{do}(x_2 = \alpha)$ and then sample from the interventional-distribution

$$x_3 = \min \{x : z_3 \leq h(x \mid x_1, x_2 = \alpha)\}$$

with the same process as for the observational sampling, with the only difference that the intervened variable X_2 stays constant.

Counterfactual queries In a counterfactual query one wants to know what the value of variable X_i would have been if another variable X_j had a different value than what was acutally observed. Pearl (2009b) describes the three-step process to answer counterfacutal queries as follows: Given a causal model M and observed evidence e (which are the actually observed values of the variables X_i of one sample) one wants to compute the probability of $Y = y$ under the hypothetical condition $X = x$.

Step 1 aims to explain the past (Z) by knowledge of the evidence e; Step 2 amends the past to the hypothetical condition $X = x$ Step 3 predicts the future (Y) based on our new understanding of the past and our newly established condition, $X = x$

Pearl named these three steps, (1) abduction, (2) action and (3) prediction. The procedure is described in the pseudocode **??** and illustrated in Figure.

Algorithm 2 Answer a Single Counterfactual Query

```
Given: A structural model $X_k = f(Z_k, \text{pa}(X_k))$, with inverse noise map $Z_k = h(X_k \mid \text{pa}(X_k))$
Input: Observed sample $x$, intervention $X_i := \alpha$, target variable $X_j$
Step 1: Abduction Infer latent variable $Z_j = h(x_j \mid \text{pa}(x_j))$ using the observed values
Step 2: Action Replace the value of $X_i$ with $\alpha$ in the set of parent variables
Step 3: Prediction Compute the counterfactual value $x_j^{cf} = h_j^{-1}(Z_j \mid \text{pa}(x_j)^{cf})$
```

While the probability of Y under the hypothetical condition $X = x$ can be determined in any case, the actual counterfactual value of Y is only defined for a continuous outcome but not for discrete outcomes.

(What pearl writes: Likewise, in contrast with the potential-outcome framework, counterfactuals in the structural account are not treated as undefined primitives but rather as quantities to be derived from the more fundamental concepts of causal mechanisms and their structure.)

ITE how it is applied in our model?

Rubins potential outcomes framework.

simulation studies "The setup was such that development and test sets were generated from the same data generating mechanism. In practice, there may be differences between these two settings that are not captured by the models, and the uncertainty that accompanies these unknowns may overshadow relatively small gains realized by more complex models."

"This could include the analysis of individual patient data from multiple randomized trials, or even the use of nonrandomized studies for the estimation of outcome risk under a control condition." this motivates the need for observational modeling.

Maybe it is the methods section. Here however, we give a couple hints. Note that you can wisely use `preamble`-chunks. Minimal, is likely:

Problems with ITE: (in an RCT setting) - to estimate the ITE we must assume un-confoundedness. Does this also apply to itneractions (effect modifiers)? Check how this is handled in the literature. - when there are treatment covariate interactions and these covariates are in the DGP but dropped from the dataset (so unobserved), then the ITE Estimation failed in the simulations. At least when there is only 1 strongly interacting variable and we drop this one. An example could be the psychological condition of a patient which might also affect how the treatment works, this is not a confounder but an effect modifier, and i would assume that this variable is rarely recorede or measured.

- Maybe a good conclusion: because this problematic with missing effect modifiers in RCT data can be a motivation to work with observational data where the dag is very detailed specified with all confounders and interactions, then a tram-dag can be applied. However, there we also have the problem, that important variables are probably also not known/measured...

- question still to answer: the estimated ITE on the train vs test set is equally bad (in terms of scatterplot and RMSE), so why does the ITE-cATE plot and the ITE Outcome plot looks like it discriminates good in the train set but not in the test set? Could the answer be, that the model is overfitting, hence tries to really model the observed outcomes and not the true probabilities, hence when an inportant variable is missing, it could still reasonably well predict the outcome (probability) but these are not the causal relationships anymore, so therefore the ITE estimation is bad on the train and the test set. But the ITE-cATE plot still looks good in the train set, because at least the observed outcomes could be predicted very well.??? still not sure if this is the case and how to proof.

- another point is the effect of the correlation of the variables. If the X's are strongly correlated, and one X with interaction effect is dropped, can the info then still be retreived from the other variables? maybe the effect is then attributed to another correlated variable. –> check with simulations and or theroretical proof.

- maybe also make propensity score estimation on IST stroke trial to check if possibly confounded.

- also talk about propensity score Rubin(2007) to basically estimate an RCT...and overcome the problem of confounding. but this might work for ATE but not really for ITE, direct modelling of the outcome is necessary

Models for ITE Estimation

T-learner vs s-learner, metalearner,

The ITE for a binary endpoint is estimated as the difference of two probabilities (the risk under treatment minus the risk under control). It is essential that the model used to estimate these probabilities is well calibrated and generalizes to new (unseen) data. When using models that are estimated with conventional methods such as ordinary least squares or standard maximum likelihood, they tend to overfit on the training data and make too extreme predictions on the test data. This problem increases with reduced sample size, low event rate or large number of predictor variables. To prevent such overfitting, penalization (shrinkage) methods are proposed as they shrink the estimated coefficients towards zero to reduce the variance in predictions on new data (Riley *et al.*, 2021).

Logistic regression, penalized logistic regression (shrinkage, lasso Shrinkage methods should provide better predictive perfomance on average (cite articles). Calster *et al.* (2020) analyzed different regression shrinkage methods with a binary outcome in a simulation study. They concluded, although the calibration slope improved on average, shrinkage often worked poorly on individual datasets. With small sample size and low number of events per variable the performance of most of these methods were highly variable and should be used with caution in such settings. Riley *et al.* (2021) obtained to similar results in their simulation study. Problems occur, because tuning parameters are often estimated with large uncertainty on the training data and fail to generalize. In both studies the autors pointed out that these penalization methods are more unreliable when needed most, that is when the risk of overfitting may be large.

In this thesis , I will apply Lasso regression on the IST stroke trial and simulation studies, where the sample size is relatively large.

2.4 Experiments

2.4.1 TRAM-DAG simulation

Show easy simulation with 3 variables and in the results the plots of the loss function, the coefficient learning, intercepts, shifts, and the sampling results. The sampling results should show that the sampled data matches the DGP very well. Also show the estimated parameters of the linear shifts and the intercepts. The complex shift can be shown by plotting the transformation function of X3 with respect to X2. also some queries for observational, interventional and counterfactual.

2.4.2 TRAM-DAG real data

maybe the Weather data case, or another if we find a practical observational data example.

2.4.3 ITE simulations

Show types of models that will be applied and dgp and when problems occure.

2.4.4 ITE real data

Results on IST trial wiht the interpretation in the discussion part.

show results of different models including tram dag.

2.5 Software

All code was done in R wiht packages xx used for yy.

```
library(knitr)
opts_chunk$set(
    fig.path='figure/ch02_fig',
    self.contained=FALSE,
    cache=TRUE
)
```

Defining figure options is very helpful:

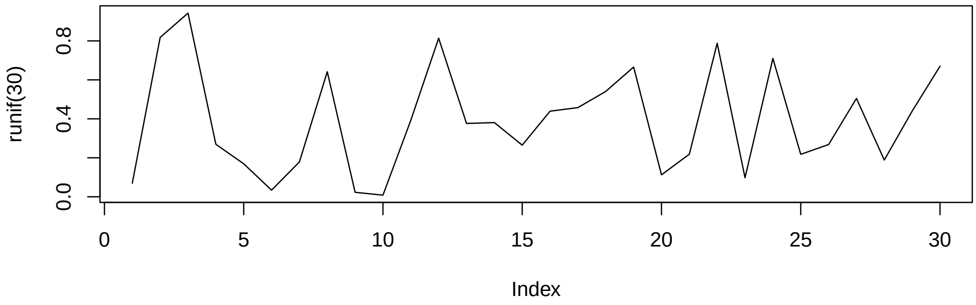

Figure 2.5: Test figure to illustrate figure options used by knitr.

```
library(knitr)
opts_chunk$set(fig.path='figure/ch02_fig',
               echo=TRUE, message=FALSE,
               fig.width=8, fig.height=2.5,
               out.width='\\textwidth-3cm',
               message=FALSE, fig.align='center',
               background="gray98", tidy=FALSE, #tidy.opts=list(width.cutoff=60),
               cache=TRUE
)
options(width=74)
```

This options are best placed in the main document at the beginning. Otherwise a `cache=FALSE` as knitr option is necessary to overrule a possible `cache=TRUE` flag.

Notice how in Figure 2.5 everything is properly scaled.

2.6 Citations

Recall the difference between `\citet{}` (e.g., Chu and George (1999)), `\citep{}` (e.g., (Chu and George, 1999)) and `\citealp{}` (e.g., Chu and George, 1999). For simplicity, we include here all references in the file `biblio.bib` with the command `\nocite{*}`.

Chapter 3

Results

Chapter 4

Discussion and Outlook

Chapter 5

Conclusions

Bibliography

Calster, B. V., van Smeden, M., Cock, B. D., and Steyerberg, E. W. (2020). Regression shrinkage methods for clinical prediction models do not guarantee improved performance: Simulation study. *Statistical Methods in Medical Research*, **29**, 3166–3178. 13

Chen, H., Aebersold, H., Puhan, M. A., and Serra-Burriel, M. (2025). Causal machine learning methods for estimating personalised treatment effects – insights on validity from two large trials. 3

Chu, E. and George, A. (1999). *Inside the FFT Black Box: Serial and Parallel Fast Fourier Transform Algorithms*. CRC Press. 14

Collett, D. (2014). *Modelling Survival Data in Medical Research*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, third edition.

Freedman, B. (1987). Equipoise and the ethics of clinical research. *The New England journal of medicine*, **317**, 141–145. 1

Hariton, E. and Locascio, J. J. (2018). Randomised controlled trials - the gold standard for effectiveness research. *BJOG: An International Journal of Obstetrics & Gynaecology*, **125**, 1716 – 1716. 1

Herzog, L., Kook, L., Götschi, A., Petermann, K., Hänsel, M., Hamann, J., Dürr, O., Wegener, S., and Sick, B. (2023). Deep transformation models for functional outcome prediction after acute ischemic stroke. *Biometrical Journal*, **65**, 2100379. 8

Hoogland, J., IntHout, J., Belias, M., Rovers, M. M., Riley, R. D., E. Harrell Jr, F., Moons, K. G. M., Debray, T. P. A., and Reitsma, J. B. (2021). A tutorial on individualized treatment effect prediction from randomized trials with a binary endpoint. *Statistics in Medicine*, **40**, 5961–5981.

Hothorn, T., Kneib, T., and Bühlmann, P. (2014). Conditional transformation models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, **76**, 3–27. 3, 5

Hothorn, T., Möst, L., and Bühlmann, P. (2018). Most likely transformations. *Scandinavian Journal of Statistics*, **45**, 110–134. 8

Hurrell, J. W., Kushnir, Y., Ottersen, G., and Visbeck, M. (2013). *An Overview of the North Atlantic Oscillation*, 1–35. American Geophysical Union.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. 8

Kratzer, G. and Furrer, R. (2018). *varrank: an R package for variable ranking based on mutual information with applications to observed systemic datasets*. R package version 0.3.

Nichols, A. (2007). Causal inference with observational data. *The Stata Journal*, **7**, 507 – 541. 1

Pearl, J. (1995). Causal diagrams for empirical research. *Biometrika*, **82**, 669–688. 11

Pearl, J. (2009a). Causal inference in statistics: An overview. *Statistics Surveys*, **3**, 96 – 146. 1

Pearl, J. (2009b). *Causality: Models, Reasoning and Inference.* Cambridge University Press, USA, 2nd edition. 2, 3, 11

Porcu, E., Furrer, R., and Nychka, D. (2020). 30 years of space-time covariance functions. *Wiley Interdisciplinary Reviews. Computational Statistics*, **13:e1512**, 1–24.

Riley, R. D., Snell, K. I., Martin, G. P., Whittle, R., Archer, L., Sperrin, M., and Collins, G. S. (2021). Penalization and shrinkage methods produced unreliable clinical prediction models especially when sample size was small. *Journal of Clinical Epidemiology*, **132**, 88–96. 13

Sick, B. and Dürr, O. (2025). Interpretable neural causal models with tram-dags. Accepted at the CLeaR 2025 Conference. 1, 3

Sick, B., Hathorn, T., and Dürr, O. (2021). Deep transformation models: Tackling complex regression problems with neural network based transformation models. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 2476–2481. 3, 5, 7

Wang, C. and Furrer, R. (2020). Monte Carlo permutation tests for assessing spatial dependence at different scales. In La Rocca, M., Liseo, B., and Salmaso, L., editors, *Nonparametric Statistics. ISNPS 2018. Springer Proceedings in Mathematics & Statistics*, volume 339, 503–511. Springer.

Chapter 6

Appendix

6.1 Negative Log Likelihood

6.1.1 Continuous Outcome

For a continuous outcome Y the CDF is given by:

$$F_{Y|\mathbf{X}=\mathbf{x}}(y) = F_Z(h(s(y) \mid \mathbf{x})) \tag{6.1}$$

where in our case F_Z is the cumulative distribution function of the standard logistic distribution

$$F_Z(z) = \frac{1}{1+e^{-z}}, \quad z \in \mathbb{R} \tag{6.2}$$

and h is the conditional transformation function that maps the scaled outcome $s(y)$ to the latent scale Z (log-odds).

The outcome y has to be scaled onto the range $[0,1]$, because the Bernstein polynomial is bounded:

$$s(y) = \frac{y - \min(y)}{\max(y) - \min(y)} \tag{6.3}$$

This scaling also has to be considered when taking the derivative to get the PDF with the change of variables formula:

$$f_{Y|\mathbf{X}=\mathbf{x}}(y) = f_Z(h(s(y) \mid \mathbf{x})) \cdot h'(s(y) \mid \mathbf{x}) \cdot s'(y) \tag{6.4}$$

Where f_Z is the PDF of the standard logistic distribution:

$$f_Z(z) = \frac{e^z}{(1+e^z)^2}, \quad z \in \mathbb{R} \tag{6.5}$$

Finally, the NLL-contributions are then given by the negative log-densities evaluated at the observations.

$$\text{NLL} = -\log(f_{Y|\mathbf{X}=\mathbf{x}}(y)) \tag{6.6}$$

The full formula is given by

$$\begin{aligned}\text{NLL} = -\log f_{Y|\mathbf{X}=\mathbf{x}}(y) &= -h(s(y) \mid \mathbf{x}) - 2\log(1+\exp(-h(s(y) \mid \mathbf{x}))) \\ &\quad + \log h'(s(y) \mid \mathbf{x}) - \log(\max(y) - \min(y))\end{aligned} \tag{6.7}$$

6.1.2 Discrete Outcome

The for a discrete outcome (binary, ordinal, categoric) with categories y_k, $k = 1, \ldots, K$, the CDF is given by:

$$F(Y_k \mid \mathbf{X}) = F_Z(h(y_k \mid \mathbf{x})) \tag{6.8}$$

The likelihood contributions are then given by

$$l_i(y_k \mid \mathbf{x}) = f_{Y_k|\mathbf{X}=\mathbf{x}}(y_k) = \begin{cases} F_Z(h(y_k \mid \mathbf{x})) & k = 1 \\ F_Z(h(y_k \mid \mathbf{x})) - F_Z(h(y_{k-1} \mid \mathbf{x})) & k = 2, \ldots, K-1 \\ 1 - F_Z(h(y_{k-1} \mid \mathbf{x})) & k = K \end{cases} \tag{6.9}$$

from which the NLL-contributions are derived

$$\text{NLL} = -\log(f_{Y_k|\mathbf{X}=\mathbf{x}}(y) \tag{6.10}$$

6.1.3 Encoding of discrete variables

In the TRAM-DAG a variable X_i can act as a predictor variable for a child node, or as a outcome (child node) that depends on some parent nodes. When X_i is acting as an outcome, the distribution of the variable X_i represented by the transformation function h which estimates a cut-point for each variable. So different form of intercept h_i is used compared to a continuous outcome variable.

If a discrete variable X_i with K categories is used as a predictor variable, it should be dummy encoded. This is done by creating $K - 1$ binary variables, where each variable indicates whether the observation belongs to this specific category/level or not. The first category/level is used as the reference and is not explicitly included in the model.

Example: for an ordinal variable X_i with three levels (1, 2 3), we create two binary variables:

- $X_{i,1}$: 1 if $X_i = 2$, 0 otherwise
- $X_{i,2}$: 1 if $X_i = 3$, 0 otherwise

Assume a continuous outcome Y that depends on the ordinal variable X with 3 levels, the CDF for Y is given by: $F(Y \mid X = 1) = F_Z(h_I(y) + x_1\beta_1 + x_2\beta_2)$

For $X = 1$, the reference level, the CDF simplifies to: $F(Y \mid X = 1) = F_Z(h_I(y))$

For $X = 2$, the CDF becomes: $F(Y \mid X = 1) = F_Z(h_I(y) + \beta_1)$

For $X = 3$, the CDF becomes: $F(Y \mid X = 1) = F_Z(h_I(y) + \beta_2)$

The coefficients β_1 and β_2 can be interpreted as the additive shift in the latent scale $h_I(y)$ when moving from the reference level (1) to levels 2 and 3, respectively.

Scaling of continuous variables

Neural networks work best when the input variables are standardized. A linear, monotonic and invertible transformation of a predictor variable changes the interpretation of the coefficient. Scaling a predictor variable X as $X_{\text{std}} = (X - mean(X))/sd(X)$ will imply that the coefficient $\tilde{\beta}$ is interpreted as the change in log-odds for a one standard deviation increase in the predictor variable or equivalently, for a one unit increase in the standardized predictor. This is different from the interpretation of the coefficient β in the original scale, which represents the change in log-odds for a one unit increase in the predictor variable.

In contrast, the standardization of the outcome variable has no effect on the interpretation (because the scale invariance of the log-odds). Consider, we standardize the outcome Y as follows:

$$Y_{\text{std}} = \frac{Y - \mu_Y}{\sigma_Y}$$

This transformation is linear, monotonic, and invertible:

$$Y = Y_{\text{std}} \cdot \sigma_Y + \mu_Y$$

Therefore, for any threshold y, we have the equivalence:

$$P(Y < y \mid X) = P\left(Y_{\text{std}} < \frac{y - \mu_Y}{\sigma_Y} \mid X\right)$$

This means that the probability is the identical when evaluating the same quantile in the standardized outcome as in the raw outcome. Furthermore, the interpretation of coefficients in a continuous outcome logistic regression remains unchanged. In particular, the log-odds ratio:

$$\log\left(\frac{P(Y < y \mid X+1)}{1 - P(Y < y \mid X+1)}\right) - \log\left(\frac{P(Y < y \mid X)}{1 - P(Y < y \mid X)}\right)$$

is equal to:

$$\log\left(\frac{P\left(Y_{\text{std}} < \frac{y-\mu_Y}{\sigma_Y} \mid X+1\right)}{1 - P\left(Y_{\text{std}} < \frac{y-\mu_Y}{\sigma_Y} \mid X+1\right)}\right) - \log\left(\frac{P\left(Y_{\text{std}} < \frac{y-\mu_Y}{\sigma_Y} \mid X\right)}{1 - P\left(Y_{\text{std}} < \frac{y-\mu_Y}{\sigma_Y} \mid X\right)}\right)$$

as long as the same quantile (i.e. probability threshold) is used. Thus, the coefficient β reflects the same change in log-odds for a one-unit increase in the (standardized) predictor, regardless if the outcome is standardized or not. This property is also crucial for the evaluation of the bernstein polynomial, since the outcome has to be scaled on a range between 0 and 1.

The general formula of the transformation model is

$$P(Y < y \mid X = x) = F_z\left(h(Y) + \beta \cdot X\right)$$

but the model is fitted with standardized outcome and predictors

$$P(Y_{\text{std}} < y_{\text{std}} \mid X_{\text{std}} = x_{\text{std}}) = F_z\left(\tilde{h}(Y_{\text{std}}) + \tilde{\beta} \cdot X_{\text{std}}\right)$$

where $\tilde{h}$ and $\tilde{\beta}$ represent the estimated transformation function and coefficients after standardizing the outcome and predictors.

For example, if we want to know the probability $P(Y < 20 \mid X = 3)$ with standardized variables, the model is specified as

$$P\left(\frac{Y - \mu_Y}{\sigma_Y} < \frac{20 - \mu_Y}{\sigma_Y} \middle| X_{\text{std}} = \frac{3 - \mu_X}{\sigma_X}\right) = F_z\left(\tilde{h}\left(\frac{20 - \mu_Y}{\sigma_Y}\right) + \tilde{\beta} \cdot \frac{3 - \mu_X}{\sigma_X}\right)$$