Master's thesis
Master's Programme in Data Science

Causal-aware Feature Selection for Domain Adaptation

Ilkka I. Porna

October 17, 2022

Supervisor(s): Dr. Antti Hyttinen

Examiner(s): Professor Mikko Koivisto
Dr. Antti Hyttinen

University of Helsinki
Faculty of Science

P. O. Box 68 (Pietari Kalmin katu 5)
00014 University of Helsinki

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's Programme in Data Science

Tekijä — Författare — Author

Ilkka I. Porna

Työn nimi — Arbetets titel — Title

Causal-aware Feature Selection for Domain Adaptation

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Master's thesis	October 17, 2022	72

Tiivistelmä — Referat — Abstract

Despite development in many areas of machine learning in recent decades, still, changing data sources between the domain in a model is trained and the domain in the same model is used for predictions is a fundamental and common problem. In the area of domain adaptation, these circumstances have been studied by incorporating causal knowledge about the information flow between features to be utilized in the feature selection for the model. That work has shown promising results to accomplish so-called invariant causal prediction, which means a prediction performance is immune to the change levels between domains. Within these approaches, recognizing the Markov blanket to the target variable has served as a principal workhorse to find the optimal starting point.

In this thesis, we continue to investigate closely the property of invariant prediction performance within Markov blankets to target variable. Also, some scenarios with latent parents involved in the Markov blanket are included to understand the role of the related covariates around the latent parent effect to the invariant prediction properties. Before the experiments, we cover the concepts of Makov blankets, structural causal models, causal feature selection, covariate shift, and target shift. We also look into ways to measure bias between changing domains by introducing transfer bias and incomplete information bias, as these biases play an important role in the feature selection, often being a trade-off situation between these biases.

In the experiments, simulated data sets are generated from structural causal models to conduct the testing scenarios with the changing conditions of interest. With different scenarios, we investigate changes in the features of Markov blankets between training and prediction domains. Some scenarios involve changes in latent covariates as well. As result, we show that parent features are generally steady predictors enabling invariant prediction. An exception is a changing target, which basically requires more information about the changes in other earlier domains to enable invariant prediction. Also, emerging with latent parents, it is important to have some real direct causes in the feature sets to achieve invariant prediction performance.

ACM Computing Classification System (CCS):
Mathematics of computing → Probability and statistics → Probabilistic representations → Causal networks
Computing methodologies → Machine learning → Machine learning algorithms → Feature selection

Avainsanat — Nyckelord — Keywords

causality, domain adaptation, feature selection, Markov blankets, latent variables, soft interventions

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1. Introduction

During recent decades, we have seen the rise of machine learning, in many ways, as a solution in the search for more accurate predictions. For example, sophisticated algorithms run over neural networks with feasible computational power and with extremely large data sets have suddenly become available for many researchers [37]. This new emerging power of machine learning enables us to reduce the uncertainty in more complex modeling challenges. This might lead many researchers, or other types of users, to rely on this new power as is [44]. By just inputting as much data as possible and then enabling an artificial intelligence machine (A.I.) to do the actual model fitting, one might expect it to output the best-performing predictive models. This approach might lead to situations, in which the predictive models might be extremely complex. Meanwhile, the role of the predictive features in the outcome model may remain unclear, thus leaving the outcome models as black boxes for the user, and thus leaving them to lose control over the outputs. Albeit operating purely with these black boxes can often yield accurate predictions and hence be well beneficial.

However, machine learning used in this way is found to be efficient, when the training data and the data used for predictions are coming from the same source domain [38]. There is no guarantee that any modern machine learning application, even with extremely large data sets and computational power available, is capable to deal with changing conditions between the training and the prediction domains. Thus, in order to trust such an A.I. solution as a robust predictor machine, one must be sure that the assumption about the same source domain holds with the data.

In reality, many domains are vulnerable to changes, and sooner or later, a shift happens within a source, leading to a situation where the data used for prediction with a model is altered from that data used to train the model [52, 46]. The change in a domain can happen in multiple ways, leading to different types of alterations in the new data used in predictions. There are many ways to show that the alteration can lead to a situation that the modeling technique used in the training phase can not cope with the change anymore with the new data [38]. And when this happens, the uncertainty can not be removed and fixed with some brute force, for example, by increasing the data units used in the training phase or using a more flexible modeling technique. The

question is then, how an A.I. can be designed to be prepared for changes in domains?

Also, a change can happen intentionally, but still it can ruin the models trained before the change. Consider a marketer trying to improve the sales of a product *Sales*. They think that the price of the product *Price* is a factor that affects the sales figures, and also that good earlier customer experiences CX affect to the sales figures. They have the sales data and corresponding price data observed. They do not have the customer experience exactly measured, but they have some indicating information about how often the product is mentioned positively in some online discussion forums, denoted as *Discussion*. They have modeled out that so far the *Price* and *Discussion* can predict *Sales* fairly well. The marketer thinks if they boost somehow the discussion in the forum, might we get more sales in upcoming weeks? In this example, the attempt to boost the discussion would be an intervention to a system of features that would change something in the data formation. The question then is that does the original model hold after this change in a feature in the model. The answer here would depend on what is the real causal structure between these features. For example, if we consider *Price* and CX actually affect the Y, and CX affecting also *Discussion*. Then boosting *Discussion* could cause the predictive model $Pr(Sales|Price, Discussion)$ might be no more valid in the new boosted domain since the predictions are higher due to the boost to *Discussion*, but in this example scenario, *Sales* is not actually affected by the boost to *Discussion*.

The domain changing between training and prediction phases is known to be potentially hazardous to predictive models, and it is well studied in many branches of the science, therefore, suggestions vary in the approaches to cope with changes [38, 6, 46]. For example, a type of *change detection* tries to hang on with changing conditions by investigating the patterns in the new incoming data and then to react it by re-modeling or updating the current models with the input from the new data [6]. In this way, the models can improve a few steps after the changes with the help of the new data points. The research area called domain adaptation focuses on finding properties from the data that can be used beforehand to better estimate circumstances after changes in a domain, without involving any data points observed from the new domain [38, 34, 35, 24, 56]. These recognized properties in training data may help to make feature selections optimized to deal with the data in the new domain. In this way, the models are designed to be operable already beforehand, without the help of any new data points.

To enable the optimized feature selection, some recent work has incorporated concepts from the area of causal inference, for example, using information from anticipated causal graphs [4, 21, 29] and focusing on the Markov blankets around the outcome variable [29, 30]. The latter one is a central workhorse to find optimal feature

sets for a prediction in the area of causal feature selection [22, 14, 1, 2]. A real Markov blanket for a target is a set of features that encapsulates all relevant information about the target variable and hence makes the prediction immune to all changes outside of the blanket. An anticipated causal graph is then crucial to enable the found models to work well in cases a change hits the features in the Markov blanket itself.

On occasions, an optimal feature selection can enable invariant prediction performance in the new domain despite the size of the change [38, 34, 35, 24, 56]. In general, this kind of property can be seen as desirable property. Having such conditions enabling an invariant prediction, a data modeler can trust the model being sound in changing settings between domains, and thus, for example, relax the need for detecting and monitoring possible changes and re-modeling or updating the models. Motivated by explained reason, in this thesis, we focus on investigating changes in the features belonging to the Markov blanket to the outcome variable, and in each case, finding out which one of the feature sets performs best across different levels of changes in the domains, and point out the possible property for an invariant prediction performance.

The approach in this thesis is to use simple causal scenarios and data sets to drive the baseline conclusions about invariant prediction properties. Throughout this thesis, we use simulated causal data sets to run experiments. We start with simple experiments without any changing conditions in the data to create the ground for the experiments with different types of changes in the next phases. The simulated data is mainly holding linear relationships between the features. Then in the actual experiments, we use minimized set of features involved with the simplest causal structures. The experiments are extended to investigate some latent feature cases to widen the scope around invariant prediction.

In Chapter 2, we introduce the baseline theory and elements for the causal modeling used in this thesis, whereas the Markov blankets, without and with latent variables, are a central workhorse for this thesis. Chapter 3 is basically about defining what we mean by a change we are interested. We take a look at recent work around the change between domains to align, what is already known. We also run some experiments to show how the known phenomena look in practice, and reason and show also show how the bias should be measured in order to catch if the bias is due to the change, or due to the feature selection used. In Chapter 4, we introduce formally how the main experiments for this thesis were conducted. Then by walking through the results from several experiment scenarios the found characteristics are summarized. In Chapter 5, we conclude with some suggestions, point out some characteristics of some findings, reflect the findings to other recent works, and discuss the limitations of this work and the possible further topics.

2. Causal modelling

This chapter introduces the baseline theory and elements for the causal modeling used in this thesis. We look into structural causal models that hold the causal information between features and are also used to create the causal data sets used in the experiments, as well as, causal graphs to present causal structures. We look into the concept of the d-separation to enable reasoning with conditional independencies between the features and continue to the concept of Markov blankets to understand the predictive power of features in causal data sets. We show with some examples, how these elements emerge in practice. We also discuss causal assumptions, causal discovery and causal feature selection.

2.1 Structural causal models

Structural causal model (SCM) for a data set can be seen as a description of the mathematical mechanisms that define, how the information flows between variables in the world, and hence, how the data set in hand has formed [30]. Thus, SCM can be seen as a way to tell a causal story behind a data set. Structural causal models can be used to generate a data set in which the variables are causally connected. SCM can also be used as a theoretical ground-truth to define how the constitution of a data set happens in the real world. The idea for the structural causal model is based on the structural equation models by Wright el al. [53] used in path analyses in their work for genetics. This concept of using paths to describe causal structures relates as well to the concept of Bayesian networks [29] we will discuss in the next section 2.3.

Formally, Pearl et al. [30] define that a structural causal model (SCM) consists of s.c. endogenous random variables V and exogenous random variables U [30]. The exogenous variables U are the input variables affecting the endogenous variables, but they are not the measured ones in a data set, and not used in modeling. In a definition of an SCM, the value for an exogenous variable is commonly drawn randomly from a probabilistic distribution. The exogenous variables V are then the variables that are getting values based on other variables U or V as an input and a variable V specific function F_V. The function F_V defines the value for the variable V based on the input.

A function defines the outcome value deterministically from the input variables. A common way to compose an SCM is to assign at least one exogenous variable as an input for each endogenous variable, and hence, if the exogenous variable is drawn from a probabilistic distribution then V is defined non-deterministically. This practice can be seen as adding an additional error term for each variable V. We write an exogenous variable U_v for each endogenous variable $v \in V$. For example, here is a definition of an SCM including all required components U, V, and F:

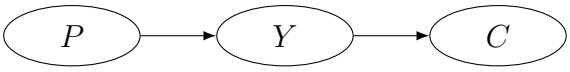
$$\begin{aligned} U &= \{U_P, U_Y, U_C\}, \\ V &= \{P, Y, C\}, \\ F &= \{f_P, f_Y, f_C\}, \\ f_P : P &= f_P(U_P), \\ f_Y : Y &= f_Y(P, U_Y), \\ f_C : C &= f_C(Y, U_C). \end{aligned} \tag{2.1}$$

In Equation 2.1 the function f_P for the variable P gets only U_P as an input variables meaning that no other variables than U_P affect the value of P. The function f_Y gets two variables as an input, P and U_Y, meaning that the value of Y is affected by the P (in addition to U_Y), as well as C is affected by Y as F_C gets Y as an input variable. From the definition of this SCM can be seen that endogenous variables in V are linked to each other so that P affects Y and Y affects C.

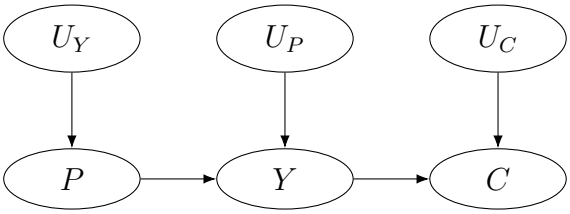
In addition to relations between the variables, a perfectly defined SCM has the content of all the functions F defined precisely allowing to define each value perfectly from other variables (only the unknown value of each U results in remaining uncertainty). In practice, this can be a challenging task to commit to perfectly. However, defining an SCM even partly allows us to test assumptions about the real world against the data sets and serves as a starting point to model out the function contents. The information about the SCM in Equation 2.1 contains only the relations in which variables are affected by other variables. However, this is not the most readable and intuitive format to read this information. For that reason, SCMs are often described with help of causal graphs which we'll look into in more detail in the next section. In this work, SCMs are used to generate s.c. ground-truth data for causally defined data sets used with examples.

2.2 Causal graphs

To describe directional connections between variables in a readable format, for example in an SCM, a causal graph is used to express how the information flows from a variable



(a) A DAG for the SCM in Equation 2.1 without exogenous variables.



(b) A DAG for the SCM in Equation 2.1 with all exogenous variables included.

Figure 2.1: Two representations of a causal graph for the SCM in Equation 2.1.

to other variables [4, 21, 29]. In this work, we define causal graphs for SCMs basically with so-called *directed acyclic graphs* (DAG) [29] (if not otherwise stated). In a DAG, variables V are shown as nodes and causal connections between them are pointed with directed arrows. In the context of SCMs, an arrow from a variable X to a variable V represents an input variable (the variable X in the origin of the arrow) for the function F_V. Thus X has a causal effect on the V. DAGs do not allow cycles in a graph, meaning that, in a DAG by following directed arrows node by node, one can not end up back to the same node again. Sometimes for clarity purposes, we also can present an exogenous variable involved in a DAG written as U_X where $X \in V$ (with a node and an arrow). Figure 2.1 shows the DAG for the SCM in Equation 2.1 represented in the both ways.

Here we describe some of the terminologies we use along with the DAGs. In Figure 2.2, as node P has an arrow pointing to node Y, we say that P is a parent to Y, and likewise, Y is a child to P, and similarly, in the graph, Y is a parent to C and C is a child to Y. All around this work, we use the following terminology from "a common family tree" to assess the roles of nodes in Figure 2.2 to each other. A PP is *a parent's parent* to Y if it has an arrow pointing to a parent node of Y. S is *a sibling* to Y if S has a common parent with Y. SP is *a spouse* to Y if it has a common child with Y. STP is *a step-parent* to Y if it has a common child with a parent of Y and STP is not a parent of Y. CC is *a child's child* to Y if a child to Y has an arrow pointing to CC. CCP is a child's child another parent if CCP has an arrow pointing to a child's child to Y. For each case, we introduce those nodes involved with a causal graph presentation.

A causal graph holds alone less information (it does not describe the functions F anyhow) about the actual causal model than a fully defined corresponding SCM. Nevertheless, it is widely used as a tool to describe and communicate the causal directions

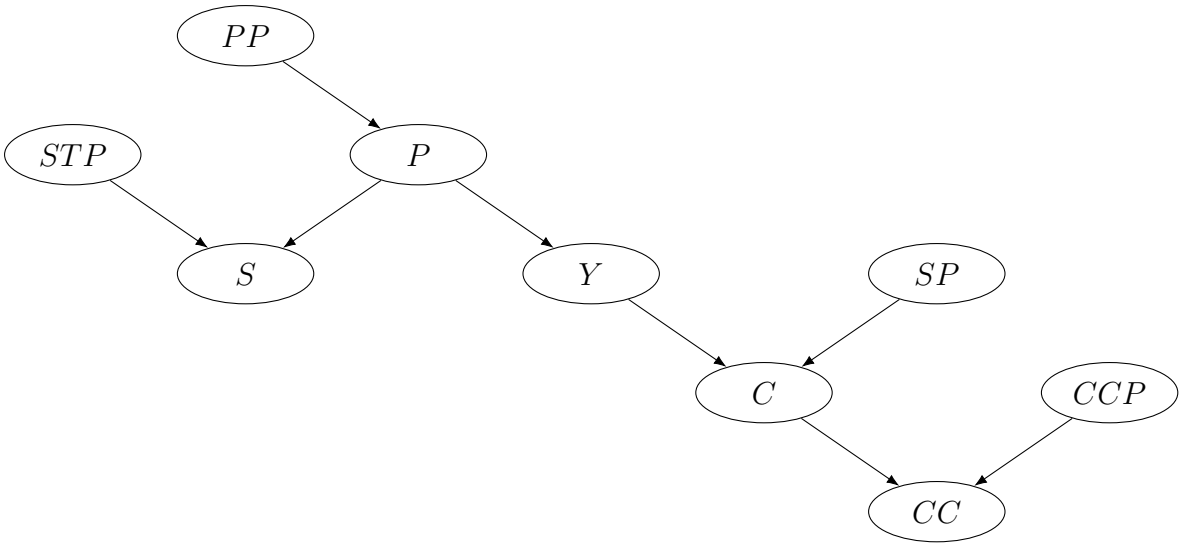


Figure 2.2: Family members to Y presented in a DAG.

between the variables. Thus alone this structural knowledge allows making further causal inferences about which nodes are affected and how big is the effect in cases of interventions to nodes. In this work, we are primarily interested how the understanding of the underlying causal structure (in form of a causal graph) can help with prediction tasks under changed settings between a model training time and model usage time. The nature of the change is discussed closely in Chapter 3. The nature of the causal graph for the underlying SCM can be enlightened from the data in hand by examining conditional independencies between the variables [30]. Many studies in causal discovery have defined algorithmic approaches for this challenge [29]. As humans are able to intuitively infer the causal directions in some cases, then by incorporating some domain knowledge about causal directions between variables can be benefited to use of the causal graph in analyses [31]. For example, we can infer the size of an ice cube from the puddle of water on the table and vice versa. But as we know the melting ice cube is the one to cause the puddle, by intervening with the size of the ice cube we know the size of the puddle is affected, but it does not work vice versa if the puddle is affected by some other cause. Either way, a causal structure is found, and the causal graph can be then benefited to estimate the true nature of underlying SCM for the data in hand.

2.3 Bayesian networks

The concept of *Bayesian network* (BN) is an alternative model to SCMs to describe causal relationships in a data set. The concept enables us to map causal graphs with data sets in hand by defining conditional independence between variables [27, 8]. By

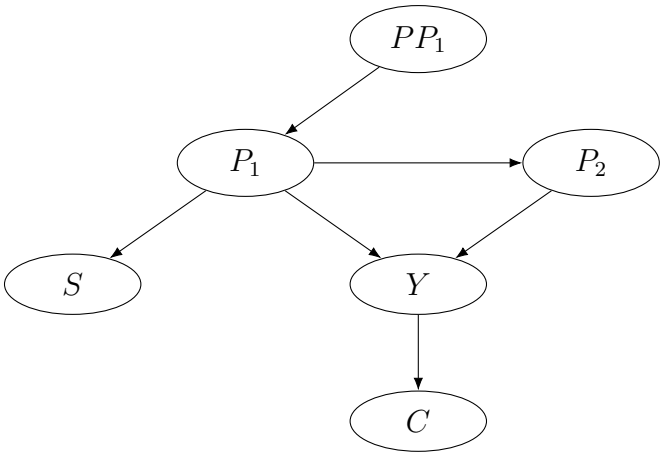


Figure 2.3: An example of a directed acyclic graph, DAG.

definition:

Definition 1 ([28]). *A Bayesian network for a set of random variables $\mathcal{V}$ is a pair $(\mathcal{G}, \Theta)$ where*

- *$\mathcal{G}$ is a DAG over $\mathcal{V}$, called the structure,*
- *Θ is a set of conditional probability distributions, one for each $V \in \mathcal{V}$ conditional on its parents $Pa(V)$ in $\mathcal{G}$ representing the parameters for the BN.*

Consider a data set with variables $\mathcal{V} = V_1, ..., V_n$. In an attempt to calculate a joint probability distribution Pr over all states in $\mathcal{V}$, a Bayesian network helps to reduce the number of parameters needed by ruling out some parameter combinations based on known conditional independencies. A DAG can be seen as a way to describe these conditions. Based on Def. 1, as DAGs are always acyclic, a Bayesian network represents a distribution that factorizes according to a given DAG with s.c.*chain rule for BNs* [27]:

$$Pr(\mathcal{V}) = \prod_{i=1}^{n} Pr(V_i|Pa(V_i)), \tag{2.2}$$

where $Pa(V_i)$ is all the parents for variable V_i in a causal graph. For example, the causal graph in Figure 2.3 can be factorized as

$$\begin{aligned} Pr(Y, PP_1, P_1, P_2, C) = Pr(PP_1) \times Pr(P_1|PP_1) \times \\ Pr(P_2|P_1) \times Pr(Y|P_1, P_2) \times Pr(S|P1) \times Pr(C|Y). \end{aligned} \tag{2.3}$$

Local Markov condition is used to define precisely the link between statistical independencies in data to an anticipated graph structure:

Definition 2 ([29, 47]). *Distribution Pr over variables $\mathcal{V}$ satisfies the local Markov condition with regard to DAG $\mathcal{G}$ if and only if every node V_i is conditionally independent of its non-descendants given the set of its parents $Pa(V_i)$.*

If Pr is modeled by a Bayesian network with the structure $\mathcal{G}$ then it requires these independence assumptions holds given by local Markov, then denoted as $Markov(\mathcal{G})$ [29]. In the other words, in order to make a valid inference about causation then the definition of the anticipated SCM must have a causal graph structure that holds $Markov(\mathcal{G})$. If a data set is generated with a SCM as defined in Equation 2.1 then it holds local Markov condition [30].

However, $Markov(\mathcal{G})$ does not include all of the conditional independence relations within the graph. These relations are looked at closely in the next section.

Two other assumptions are crucial to note if they hold in both the graph and the data in hand to understand what they are capable to explain about the world they represent. S.c. *faithfulness condition* demands:

Definition 3 ([29, 47]). *Given a Bayesian network $\mathcal{V}, \mathcal{G} and Pr(\mathcal{V})$, DAG $\mathcal{G}$ is faithful to $Pr(\mathcal{V})$ if and only if every conditional independence present in P is entailed by $\mathcal{G}$ and the Markov condition. $Pr(\mathcal{V})$ is faithful if and only if $\mathcal{G}$ is faithful to $Pr(\mathcal{V})$.*

This is basically important as with finite data points it is possible that some set of parameters (for example, correlation coefficients for a linearly defined SCM) in an SCM can cause by a chance that relations between variables do not hold as assumed in the resulting data set, making it *unfaithful* for the model. For this work, we use randomized parameter values in the data simulations and several trials to avoid such misleading conclusions.

S.c. *causal sufficiency assumption* of the data states the following:

Definition 4 ([29, 47]). *Every direct common cause for two variables in $\mathcal{V}$ is also in $\mathcal{V}$.*

Causal sufficiency often appears as a strong assumption, as it is generally considered impossible to ensure that all possible common causes are measured [33]. In this work, we occasionally relax this assumption by allowing s.c.*a latent variable* (unobserved in the data set) to be a parent to two observed variables, and hence not forced to make such a strong assumption about knowing all common causes.

2.4 Conditional independence and d-separation

Typically causal models represented with DAGs involve multiple paths between variables connecting them and paths traverse through other variables. A direct arrow between two variables ($X \to Y$) means variables X and Y are likely dependent. For example, in the definition of SCM that generates the model data set, a value from X is included as a parameter in the function f_Y that defines the value for Y [29]. However,

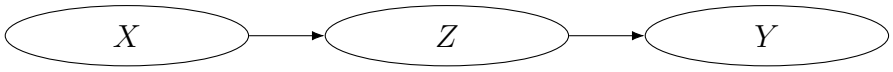


Figure 2.4: A chain structure represented as a causal graph.

if the nodes X and Y are connected indirectly, without any direct path, but with some other nodes Z in between them in a path, then X and Y might be as well conditionally independent or conditionally dependent given Z based on directions in the graph. In a DAG, a path can consist of a sequence of any number of three types of structures between two variables having a variable in between them [29]. Next, we introduce these three types of structures between variables and show in which cases the conditional independence between X and Y given Z occurs or not.

2.4.1 Chain, fork and collider structures

Definition 5 (**Rule with chains** [29, 30])**.** *Variables X and Y are conditionally independent of given Z, if there is only one directed path from X to Y, and Z is any set of variables that intercepts that path.*

In the case of a chain structure, an example in Figure 2.4, when making predictions about Y, then $Pr(Y|X,Z) = Pr(Y|Z)$. If Z is observed then X is not relevant in prediction. If Z is unobserved then $Pr(Y|X)$ is supposedly better predictor than $Pr(Y|\emptyset)$. An example is shown in Figure 2.7.

Definition 6 (**Rule with forks** [29, 30])**.** *If a variable Z is a common cause of variables X and Y, and there is only one path between X and Y, then X and Y are independent conditional on Z.*

In the case of a fork structure, an example in Figure 2.5, when making predictions about Y, then $Pr(Y|X,Z) = Pr(Y|Z)$. If Z is observed then X is not relevant as a predictor. If Z is unobserved then $Pr(Y|X)$ is valid and is supposedly better predictor than $Pr(Y|\emptyset)$. An example is shown in Figure 2.8.

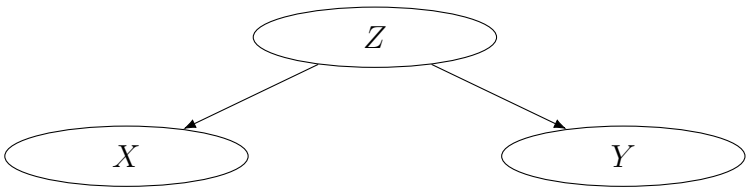


Figure 2.5: A fork structure (a common cause) represented as a causal graph.

Definition 7 (**Rule with colliders** [29, 30])**.** *If a variable Z is a collision node between X and Y, then X and Y are unconditionally independent but are dependent conditionally on Z and on any descendants D of Z.*

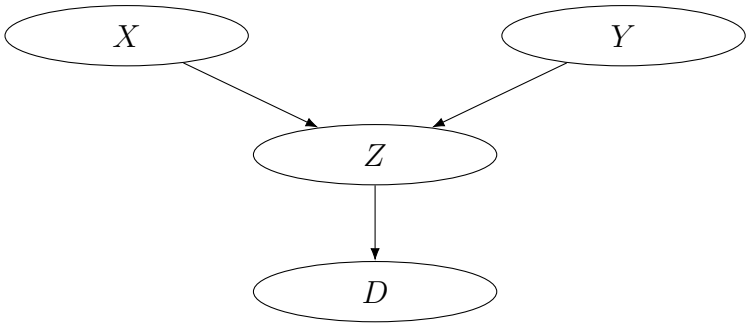


Figure 2.6: A collider structure (a common effect) represented as a causal graph. D is the descendant node for the collider node Z.

In the case of a collider structure, an example in Figure 2.4, when making predictions about Y, then $Pr(Y|X) = Pr(Y|\emptyset)$. If Z is unobserved then X alone does not transmit any information about Y. If both Z and X are observed then $Pr(Y|Z,X)$ is supposed to yield the best prediction since both X and Z together transmit information about Y. If X is unobserved then $Pr(Y|Z)$ is supposed to give better predictions than $Pr(Y|\emptyset)$ since X alone can transmit information about Y. An example is shown in Figure 2.9.

2.4.2 d-separation

A pair of nodes $\{X,Y\}$ in the graph are *d-separated* if all the paths in the graph between them are blocked [29]. Even if one path is not blocked then the pair $\{X,Y\}$ is said to be d-connected. How a path can be blocked links to conditional independence rules with the three types of connecting nodes: chain, fork, and collider. In the case of chains and forks, a set of nodes Z in between makes X and Y independent given Z thus blocking information to flow from X to Y or vice versa. In these cases, we can think that information from X to Y (and vice versa) becomes irrelevant in presence of Z in between. In the case of colliders, Z makes X and Y dependent, and such unblocks the path between them. We can think that X becomes relevant to infer about Y in presence of Z (and vice versa).

Formally the blocking that causes the d-separation is defined as follows:

Definition 8 ([28])**.** *A path p is blocked by a set of nodes Z if and only if*

1. *p contains a chain of nodes $A \rightarrow B \rightarrow C$ or a fork $A \leftarrow B \rightarrow C$ such that the middle node B is in Z (i.e., B is conditioned on), or*

2. *p contains a collider $A \rightarrow B \leftarrow C$ such that the collision node B is not in Z, and no descendant of B is in Z.*

When a pair of nodes are d-separated, then the variables they represent are definitely independent, when a pair of nodes are d-connected, then they are possibly, or most likely, dependent [29]. (The d-connected variables will be dependent for almost all sets of functions assigned to arrows in the graph, one exception being certain intransitive cases [30].)

A judgment for d-separation between variables comes only from interpretations from a causal graph, based on the conditional independence rules defined earlier. The link to conditional independence can be seen as a fact to check that must be held in order the make a justification valid. Given just a data set and finding some conditional independence between variables given some other variable does not allow one to make any judgment about the d-separation without any interpretation from a graph.

Given a data set and a graph, examining whether some nodes are d-separated or d-connected helps to clarify how these nodes relate with other nodes involved. As for this work, we are mainly interested in causal structures in prediction tasks, thus for this purpose, a note of d-separation over an anticipated causal graph helps to define if some variable has some predictive power or not to a target variable.

2.4.3 Conditional independence check with linear regression models

In the case of continuous variables, one approach to test the conditional independence between variables is to fit a linear regression model to the data and infer the conditional independencies from the coefficients of the outcome model [51, 30]. For example, to estimate $Pr(Y|X, Z)$ consider a linear model for the outcome in the following form:

$$\hat{Y} = \hat{r_X} X + \hat{r_Z} Z + \hat{\beta}, \tag{2.4}$$

where $\hat{Y}$ is the prediction to Y. $\hat{r_X}$ and $\hat{r_Z}$ are the estimated coefficients for X and Z, and $\hat{\beta}$ is the estimated intersection value for the best fitting model. Estimation is done by minimizing the sum of squared error SE:

$$SE = [Y - \hat{Y}]^2. \tag{2.5}$$

If some of the estimated coefficients for the variables approach zero value then the conclusion is that this variable is conditionally independent given the rest of the variables whose coefficient estimates are not approaching zero value. In other words, only those estimated coefficients that affect the value $\hat{Y}$ somehow are relevant in the model.

Next, we show examples of how conditional independencies and therefore d-separations in DAGS can be revealed in chain, fork, and collider structures. Consider

a situation with a chain structure defined by the following SCM:

$$\begin{aligned} Z &= \mathcal{N}(\mu_Z, \sigma_Z^2), \\ X &= r_Z Z + \mathcal{N}(\mu_X, \sigma_X^2), \\ Y &= r_X X + \mathcal{N}(\mu_Y, \sigma_Y^2), \end{aligned} \tag{2.6}$$

that generates a data set $\{Z, X, Y\}$ and the corresponding DAG is then $Z \rightarrow X \rightarrow Y$. The parameters $\mu_{Z,X,Y}$ and $\sigma^2_{\{Z,X,Y\}}$ are the means and the variances for random variables drawn from normal distributions $\mathcal{N}$. The parameters $r_{\{Z,X\}}$ are the real coefficients in the model. Consider these parameter values are fixed with following values: $\mu_Z = 1$, $\mu_X = 1$, $\mu_Y = 1$, $\sigma^2_{\{X,Y,Z,W\}} = 1$, $r_Z = 2$, $r_X = 3$.

First, we fit with linear regression for $\hat{Y}$ by using X and Z as the predictors:

$$\hat{Y}_{\{Z,X\}} = \hat{r}_Z Z + \hat{r}_X X + \hat{\beta}. \tag{2.7}$$

The resulting model with estimated parameters $\hat{r_Z}$, $\hat{r_X}$, $\hat{\beta}$ is the one that minimizes the mean squared error (MSE) in the training set. In comparison we fit also a model by using only Z as a predictor:

$$\hat{Y}_{\{Z\}} = \hat{s}_Z Z + \hat{\beta}. \tag{2.8}$$

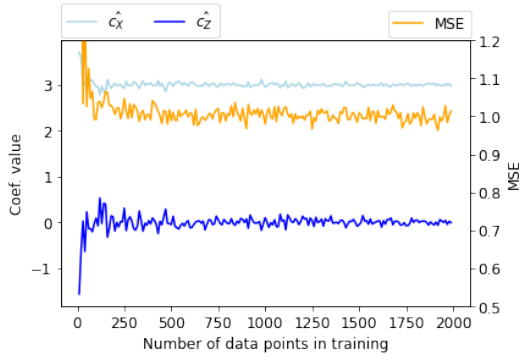
From the results in Figure 2.7a we can see that, as the number of data points in the training phase increases, then $\hat{r_X}$ approaches r_X, but $\hat{r}_Z$ approaches the zero while in the original SCM the value of $_Z$ is 2. This is exactly due to Z becoming independent from Y as X is available in the middle of the chain. As an opposite, in Figure 2.7b Z is the only predictor and the coefficient $\hat{s}_Z$ for Z gets values close to 6 meaning now Z and Y are dependent as X is unavailable. The resulting value which is approximately 6 actually comes by multiplying original coefficients: $r_Z r_Z = 2 \cdot 3 = 6$ [47]*. However, the error level with Z alone is ten times higher (MSE around 10) than with X. The value Mean Y MSE shows the error levels (around value 47) if the mean of Y would be the predicted value all over. Therefore we conclude Z alone is a weaker predictor than X but better than relying purely on the mean value of Y. If X is available Z becomes useless in the model and therefore the best predictor set is $\{X\}$.

Consider now a fork structure created with the following SCM:

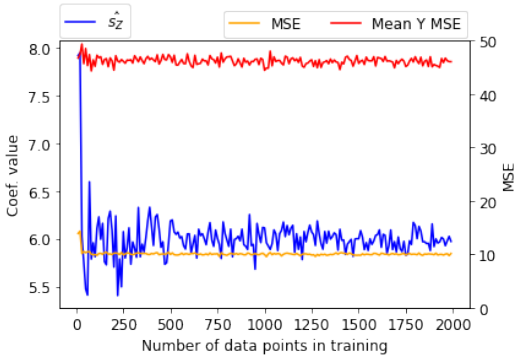
$$\begin{aligned} X &= \mathcal{N}(\mu_X, \sigma_X^2), \\ Z &= r_X X + \mathcal{N}(\mu_Z, \sigma_Z^2), \\ Y &= r_X X + \mathcal{N}(\mu_Y, \sigma_Y^2), \end{aligned} \tag{2.9}$$

that generates again a data set $\{Z, X, Y\}$ and now the DAG is $Y \leftarrow Y \rightarrow Z$. All parameters are as in the example with a chain. We make again two linear regressions

*This is based on the concept of the trek rules by Sprites et al. [47]

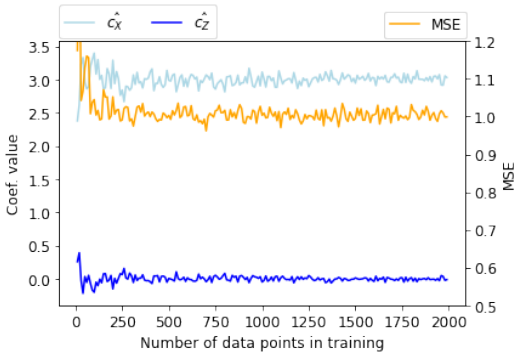


(a) Estimates for the coefficients $\hat{r}_X$, $\hat{r}_Z$ and the resulting MSE in the function of the training data points.

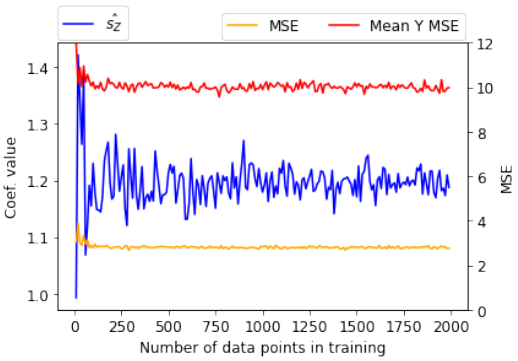


(b) Estimates for the coefficients $\hat{r}_Z$ as Z being the only predictor.

Figure 2.7: Linear model parameter estimates in the case of a chain.



(a) Estimates for the coefficients $r\hat{_X}$, $r\hat{_Z}$ and the resulting MSE in the function of the training data points.



(b) Estimates for the coefficients $r\hat{_Z}$ as Z being the only predictor.

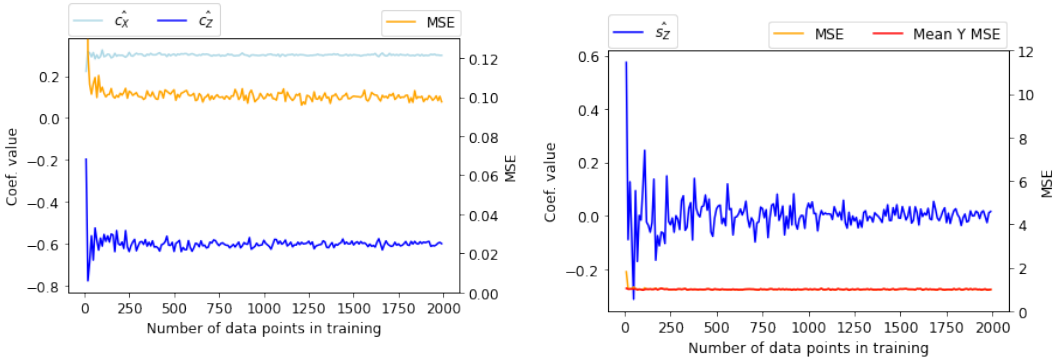
Figure 2.8: Linear model parameter estimates in the case of a fork.

to estimate parameters for $\hat{Y_{\{Z,X\}}}$ and $\hat{Y_{\{Z\}}}$. The results from Figure 2.8a shows that when predicting $\hat{Y_{\{Z,X\}}}$ then $r\hat{_Z}$ approaches zero and $r\hat{_X}$ approaches r_X meaning Z is independent to Y given X in the model. Figure 2.8b shows that when predicting $\hat{Y_{\{Z\}}}$ then in absence of X in the model $s\hat{_Z}$ gets value close to 6 meaning Z and Y are dependent. However, the prediction performance in MSE is weaker with Z than with X, but still better than Mean Y MSE.

Then consider a collider structure created with the following SCM:

$$\begin{aligned} X &= \mathcal{N}(\mu_X, \sigma_X^2), \\ Z &= r_X X + \mathcal{N}(\mu_Z, \sigma_Z^2), \\ Y &= r_X X + \mathcal{N}(\mu_Y, \sigma_Y^2), \end{aligned} \tag{2.10}$$

that generates again a data set $\{Z, X, Y\}$, but now the DAG is $Z \rightarrow X \leftarrow Y$. The results from Figure 2.9a shows that when predicting $\hat{Y}_{\{Z,X\}}$ then both coefficients r_Z and r_X approach non-zero values, hence affecting the prediction of $\hat{Y}$, meaning that in



(a) Estimates for the coefficients $\hat{r}_X$, $\hat{r}_Z$ and the resulting MSE in the function of the training data points.

(b) Estimates for the coefficients $\hat{r}_Z$ as Z being the only predictor

Figure 2.9: Linear model parameter estimates in the case of a collider

such a collider structure Z and Y are dependent given X. Figure 2.9b shows that when trying to predict $\hat{Y}_{\{Z\}}$ then the coefficient $\hat{s}_Z$ for Z is approaching zero. this means that if X is not available in the prediction then actually Z becomes independent of Y and now Z is not giving any predictive power to the model. The MSE value (in this case 1)is then the same as Mean Y MSE.

2.5 Markov blankets

A *Minimal Markov blanket* for a variable in a Bayesian network is a minimal set of variables that provides all necessary information available to make inferences about the variable in case. This is the main concept for this work to set which relations we are interested in causally in an attempt to predict a target variable in changing settings. We show in this section how the Minimal Markov blanket is defined within Bayesian networks and how it can be observed from a data set. For this work we state Minimal Markov blanket just *Markov blanket* and as such we define the Markov blanket to $Y \in \mathcal{V}$:

Definition 9. *[29, 30] The Markov blanket (MB) for a variable Y is a minimal set of variables MB such that $Y \notin MB$: $Y \perp\!\!\!\perp (\mathcal{V} \setminus MB) \setminus \{Y\} | MB$.*

Alone this definition does not bring any information about causal relations between the variables within the Markov blanket. However, in the context of a Bayesian network assuming a given graph for a data set holds the Markov condition, MB d-separate a variable Y from all of the other variables $(\mathcal{X} \setminus B) \setminus \{X\}$. Then in the graph, all the parents, children, and spouses are the variables belonging to MB. The realization of a Markov blanket is that if all variables in the Markov blanket for Y are observed then all other variables do not offer any better information about Y (i.e. in an

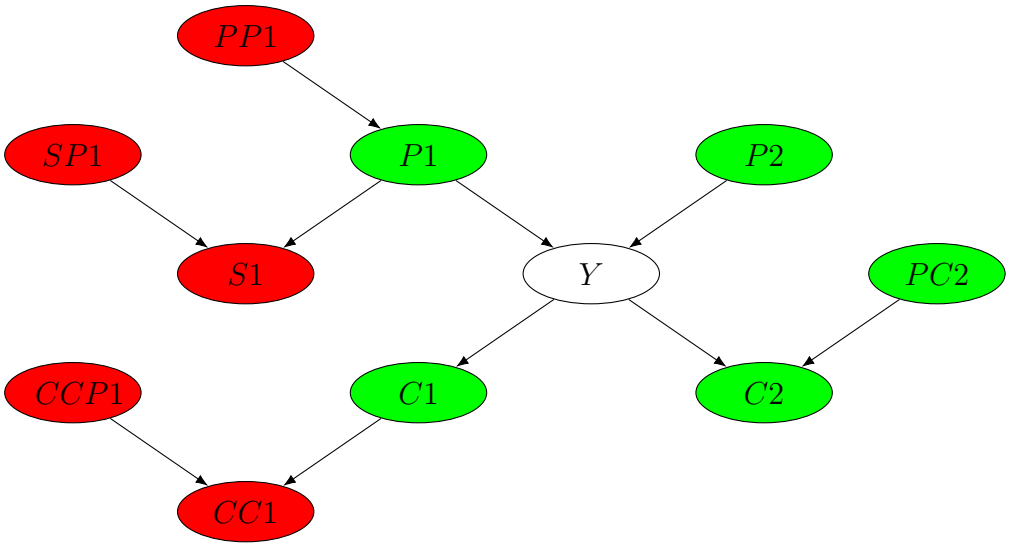


Figure 2.10: A causal graph with the Markov blanket for Y represented in green nodes. All other variables (red nodes) are irrelevant.

attempt to predict Y) and must be considered as redundant ones to those observed in the blanket. The purpose of the Markov blanket is to distinguish all relevant variables for Y. As all the relevant variables are observed, then other variables are irrelevant and should not be used in attempts to predict Y.

2.5.1 Example of a Markov blanket

We look at an example of a Markov blanket in a graph and reason with rules of d-separation how the elements (parents, children, and spouses) of the Markov blanket can be detected from this graph.

The Markov blanket to Y is shown in green nodes in Figure 2.10. We reason next with the rules of d-separation, why each variable is either relevant or not to be used to predict Y. We assume that all variables are observed:

Y is d-connected to variables $P1$ (a parent), $P2$, $C1$ (a child) and $C2$ straight due to chain rule. Variable $PP1$ (a parent's parent) is d-separated from Y due to chain rule ($PP1 \rightarrow P1 \rightarrow Y$) since the middle node $P1$ is observed. $S1$ (a sibling) is d-separated from Y due to fork rule ($S1 \leftarrow P1 \rightarrow Y$) since a middle node $P1$ is observed. $CC1$ (a child's child) is d-separated on Y due to chain rule ($Y \rightarrow C1 \rightarrow CC1$) since the middle node $C1$ is observed. $PC2$ (a spouse to Y) is d-connected to Y due to the rule of collider ($Y \rightarrow C2 \leftarrow PC2$) since the colliding node $C2$ is observed.

As the result, all observed d-connected variables to Y are included in the Markov blanket to Y as being the potential to omit information about Y. All d-separated variables are not relevant to be included due to not offering anything more about Y than the variables already in the blanket).

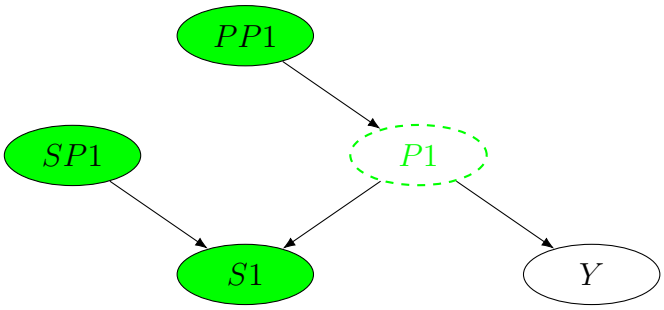


Figure 2.11: If the data from parent $P1$ is not observed then parents, children, and children's other parents of $P1$ become relevant and part of the Markov blanket to Y.

2.5.2 Markov blankets with latent variables

In a casual structure, also other variables than in MB_{entire} = {parents, children, spouses} to Y can be part of the Markov blanket to Y, if some of the variables in MB_{entire} are not observed a.k.a. latent variables [33]. This kind of setting may relax the causal sufficiency assumption (Def. 4) as now an unobserved variable can be a common cause for two observed variables, for example, a latent parent P to Y is a common cause to Y and to another observed child of P called S (a sibling to Y). Now consider the structure shown in Fig 2.11 in which the parent $P1$ is not observed in the data set in hand, then reasoning with d-separation gives us different results than the case in Figure 2.10. Now since $P1$ is not observed the rule of forks does not hold anymore and the sibling $S1$ and Y become d-connected since the middle node is not present. Now because $S1$ is an observed collider for unobserved $P1$ and the step-parent $SP1$, this makes $SP1$ and $S1$ d-connected to $P1$. Meanwhile, due to the chain rule, the parent's parent $PP1$ and Y become d-connected in the absence of the parent $P1$. As a result, in absence of $P1$ the Markov blanket to Y is extended with observed variables $PP1$, $S1$ and $SP1$.

Similarly, shown in Figure 2.12, we have a case in the left child branch of Y, in which child node $C1$ is a latent variable. This unblocks the path between the child's child $CC1$ and Y due to the chain rule making them d-connected. Again since $CC1$ is now an observed collider for the child's spouse $CCP1$ and the latent child $C1$ this makes them d-connected. As a result, in absence of $C1$ the Markov blanket to Y is extended with the child's child $CC1$ and the child's spouse $CCP1$.

2.5.3 Examples of Markov blanket discovery

Next, we set up an experiment to see how Markov blanket can be revealed from a data set of variables using linear regression model, similarly as in Figures 2.7, 2.8 and 2.9. The assumption by the rules of d-separation and conditional independence in Bayesian

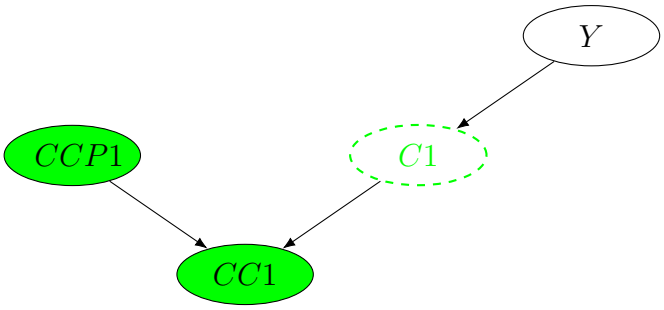


Figure 2.12: If the data from child $C1$ is not observed then children and children's other parents of $P1$ become relevant and part of the Markov blanket to Y

.

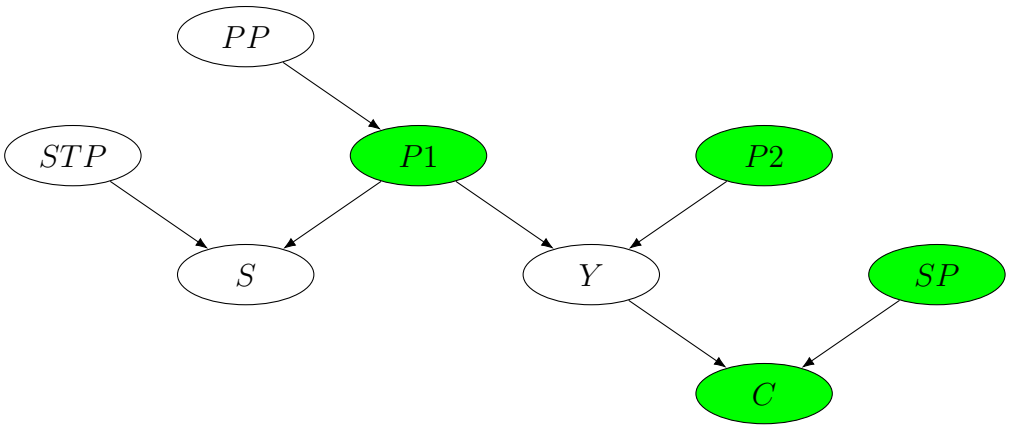


Figure 2.13: Markov blanket to Y, without any latent variables, shown green.

networks (Definition 8) is that the variables not belonging to MB will be weighted close to zero value coefficient in the outcome linear model, and only those belonging to MB will get some meaningful coefficient values in the model [33, 9]. We also test how the size of the training data affects distinguishing those variables belonging and not to the actual MB in question. We examine two scenarios: the first one is without latent variables and the second one is with one variable being a latent parent.

We use simulated data for the experiment by creating an SCM with the following setting: Consider the set of variables in the causal graph in Figure 2.13. Each variable V has an exogenous variable U_V drawn from normal distribution $\mathcal{N}(1,\, 1)$ and the value of each variable U_V is then summarized with it's parents multiplied with coefficients drawn from $\mathcal{U}([-2, -0.5]) \cup \mathcal{U}([0.5, 2])$. The idea is to avoid coefficients too close to zero value to make them distinguishable from variables getting actually weighted close to zero in results. The data set is then re-created with an increasing number of training data points. For each data set, we fit a linear regression to find coefficient estimates for the variables to predict the node Y. The resulting coefficient values for each variable are plotted in Figure 2.13 against the number of training data points used.

From the results for the first scenario in Figure 2.15 we see the variables not

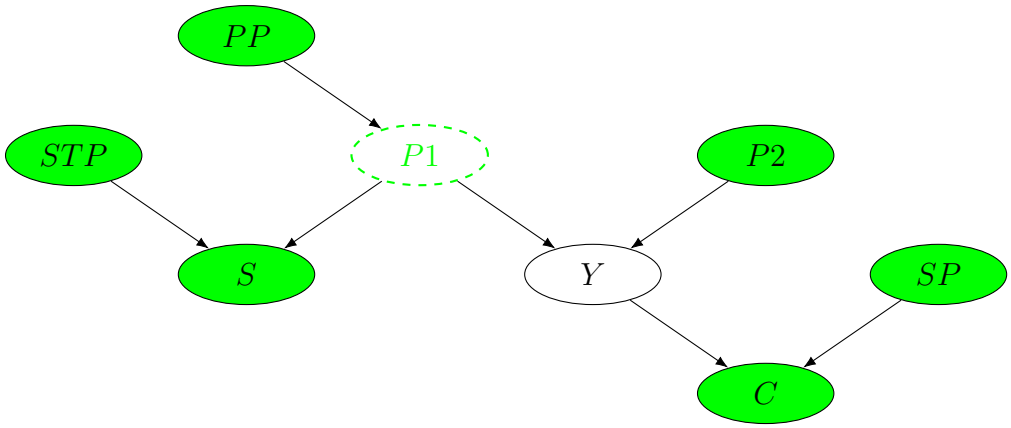


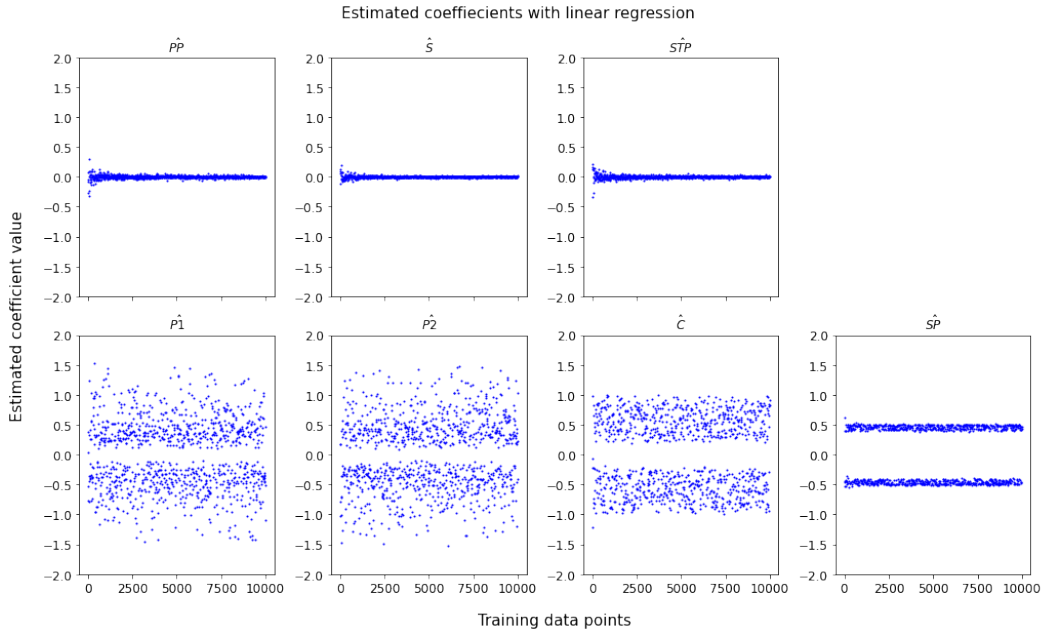
Figure 2.14: Markov blanket to Y with the latent parent $P1$ shown in green.

belonging to the Markov blanket in the graph in Figure 2.13 (Parent's parent (PP), Sibling (S) and Step-parent (STP)) get the values for their parameter estimates $\hat{PP}$, $\hat{S}$ and $\hat{STP}$ from the linear regression models soon close to zero as the number of training data point increases (the value converges to zero). Meanwhile the estimates for the variables belonging to MB ($\hat{P1}$, $\hat{P2}$, $\hat{C}$, and $\hat{SP}$) are getting estimates clearly off from the zero value. In Figure 2.14, we have the results for the second scenario with a latent parent. Now we see that actually estimates $\hat{PP}$, $\hat{S}$ and $\hat{STP}$ are getting values off from the zero, meaning they become meaningful for the fitted linear models. This is exactly due to that in absence of parent $P1$, it's parent PP, sibling S, and step-parent STP nodes that become part of the Markov blanket to the target Y.

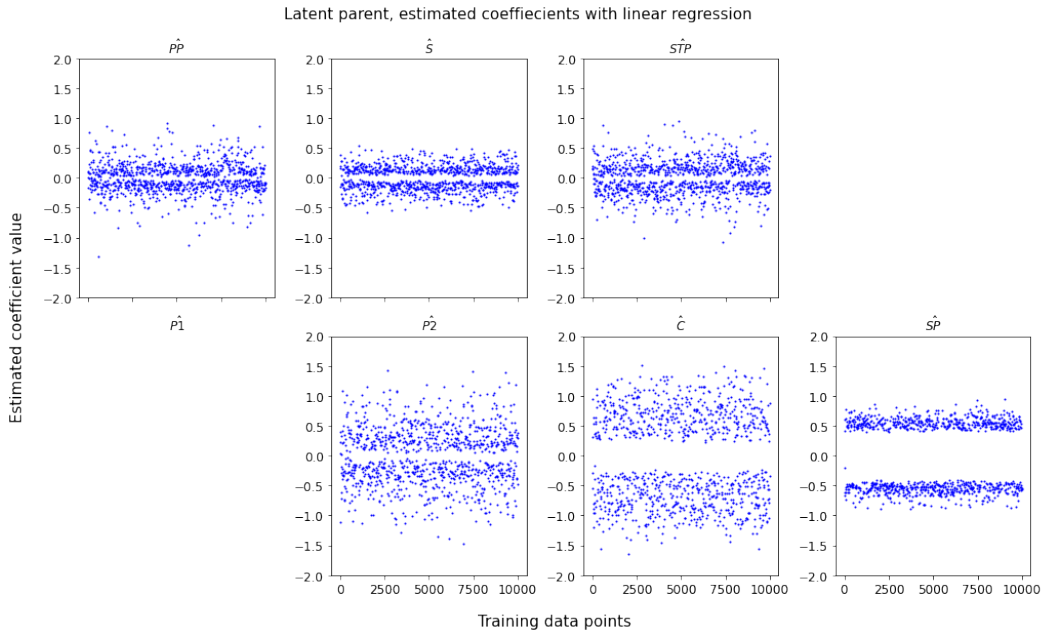
Through these experiments we have shown that in the case we have a linear ground-truth between variables and linear regression models are used to estimate the parameters for the variables, then actually the Markov blanket to the target variable (whether a full blanket or with some latent variables) can be revealed from the data. This can be seen as a simple way of finding causal structures from data sets.

2.5.4 Causal discovery

The examples in Figure 2.15 revealed the variables belonging to the Markov blanket among all available variables. However, the end results did not provide any information about in which roles the variables are in the causal graph, for example, which one is a parent, a child, a spouse, etc. Also if the ground-truth was beyond linear and the amount of variables increases the task is not anymore as straightforward. The field of causal discovery focus on finding causal structures from data sets algorithmically [28]. In these techniques, conditional independence tests between variables are at the core of the mechanism. The purpose can be either to find just the variables belonging to the Markov blanket or to aim to reveal the structure more precisely. The end result from



(a) The results from the experiment of Markov blanket without any latent variables.



(b) The results from the experiment of Markov blanket with P_1 as a latent parent variable to Y. In this case, P_1 has no data points observed.

Figure 2.15: The coefficients from the linear modeling against two linearly defined causal data sets. Those variables whose coefficient values are estimated as seemingly non-zero values are considered to belong to the Markov blanket.

such a discovery algorithm to reveal the structure is not necessarily a complete DAG, but a type of acyclic graph holding some uncertainty i.e. about the directions of the arrows and possible hidden common causes. Especially possible hidden common causes can lead to situations that the learned structure that can consist of several possible

choices for proposed structures. For example, algorithms like Inductive Causation* (IC*) [32, 29] and Fast Causal Inference (FCI) [47] return s.c. partial ancestral graph (PAG), which indicates for each link whether it (potentially) is the manifestation of a hidden common cause for the two linked variables [33]. Silva et al. presents [9] with linear continuous variables that hidden variables that are the cause for more than two observed variables can be recovered to infer the relationships between the hidden variables themselves.

2.5.5 Causal feature selection

Feature selection in machine learning is about to identify a subset of features from the original features for optimal usage in building predictive models or understanding feature importance [15, 23]. In the era of huge data sets the number of features in data sets has expanded, for example, in cancer genomics, a gene expression data set can contain tens of thousands of features (genes) [55]. That is why effective ways to select optimal features for prediction purposes have become an emerging interest [54]. Yu et al. [54] discuss feature selection within s.c. filter-based feature selection, and from recent literature, they distinguish two kinds of approaches to achieve the goal algorithmically. The first one is so-called *classical feature selection* based on creating a ranking for the features by their relevance to the target variable [15]. The second one is s.c. causal feature selection based on detecting the Markov blanket to the target variable. The latter one has been an emerging approach in the field [22, 14, 1, 2].

Some analysis of the algorithm mechanism in both approaches reveals that both actually share the common goal [54] to focus on the highly relevant features. With a higher number of features involved in the actual Markov blanket (in the ground-truth) it makes the discovering task of the Markov blanket harder and the classical approach can yield better performance in speed to achieve a comparable feature set to the actual ground-truth set.

In that sense, for a setting where a change is not supposed, we can conclude that both feature selection strategies are valid. But as a domain change rolls in, then acknowledging the Markov blanket has more benefits to cope with change [19, 33]. For that reason, in this work, we focus on the feature selection done in a causal way by detecting the Markov blanket and doing the selection within the blankets.

3. Prediction across changes between the domains

As described in the previous chapter, in an attempt to predict a target variable Y with several features $\mathcal{V}$ in the data set, then Markov blanket $MB(\mathcal{V})$ to Y is generally the most optimal subset of features, meaning any other subset $\mathcal{V}$ usually does not yield better predictive performance. However, what if something changes in the formation of the data sets after the model is trained? Can we suppose that a model trained with $MB(\mathcal{V})$ to Y remains a good predictor in the new data or could a model trained with some subset of features $MB(\mathcal{V})$ perform actually better under a change in a domain?

In this chapter, we look into how a change may happen in the system under a study. Then we review from recent literature, what we know about certain types of changes, and how they can affect to a prediction task. We conduct some experiments to show how these phenomena can be seen in practice. Finally, to conduct an experiment later in the next chapter, we define the change we are focusing on in this thesis. We pay attention during this chapter to seek feature sets to possibly yield models which are capable to give stable predictive performance across a change in the domains, despite the magnitude of the change [34, 35, 24].

As a side note, because the focus area is in the mechanism to predict under changing conditions beforehand, we exclude the approaches that use any new data from the new domain to deal with changes, for example, the area of *change detection* investigates methods to detect a change from data and then re-model or update the predictive models to keep on with changing conditions [6]. In our cases, we consider that we do not have any new data points observed.

3.1 Hard and soft intervention

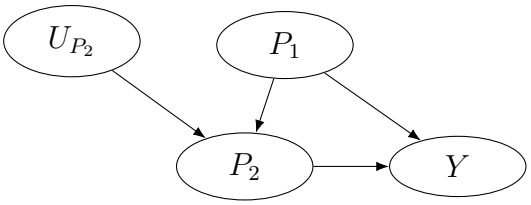
Interventions to a system under study are at the core of the research on causal inference [30, 18]. For example, a common task in causal inference is to measure the effect of an intervention on the target variable. This can be done by using assumptions about the causal graph to calculate the effect from observed data in a way that it mimics the

conditions achieved with randomized controlled trials (RCT). The RCT is seen as a gold standard to measure causal effects from empirical studies. In this work, instead of measuring the causal effect, we are rather interested how well predictive power holds after changes in the source of data generation in feature selection wise. However, these changes can be seen as interventions as well. Next, we look at how these interventions look technically if presented in a structural causal model definition.

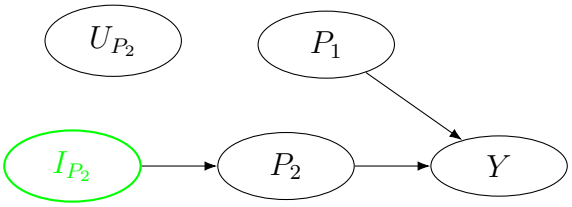
Eberhardt et al. [10] consider two types of interventions to causal graphs: structural and parametric. Structural intervention a.k.a. hard intervention is like forcing the variable under intervention to be a fixed value. This kind of assignment mechanism enables us to measure the causal effect [18]. Using a causal graph, a representation of a hard intervention in Figure 3.1b is done by cutting off all arrows pointing to the variable under intervention including also exogenous variables, hence leaving the intervention itself fully determine the value for this variable [29]. Parametric intervention a.k.a soft intervention is done by adding an additional factor to affect the variable under intervention. All other connections stay the same. With a causal graph, a representation (Figure 3.1c) is done by adding to the graph an extra variable with an arrow pointing to the variable under intervention.

A hard intervention corresponds to a similar type of setting as we looked at a data set after randomly controlled trials [13]. For example, the basic idea in a drug test setting first the test group and focus group have divided randomly thus ensuring that groups are constituted without bias in the selection. The real drugs are given to the test group members, while placebo drugs to the focus group members. The effect is calculated from the difference in the outcome (for example, a defined positive effect) between the groups. In this way, it is ensured that drugs are taken or not taken in full control, without any other factors in real life affecting the intake of the drug. The basic idea is that the balance between the groups is ensured by randomness a.k.a the assignment process to treatment and focus groups is done completely at random [18]. In this scenario, a researcher would be actually testing if the feature (i.e. a drug) under intervention is on the causal path to the target outcome (i.e. positive effect on health).

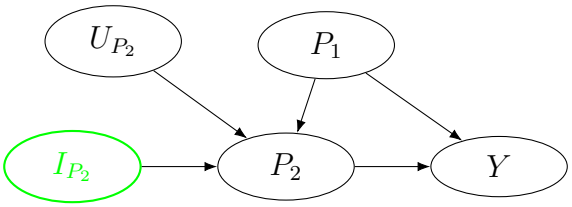
On the other hand, in the case of purely observational data, a data set is coming from a field without any controlled experiment or intended interventions behind it. For example, a drug is taken by own choice. Then the assignment process to the treatment and focus groups is not guaranteed to be done completely at random since other factors (possibly observed in the same data set) may affect this assignment process [18]. In these circumstances, a researcher needs to find ways to balance the groups with other covariates in the data set in order to make a comparison between treated and untreated to drive causal claims about the treatment. Many approaches have been studied to achieve this goal. Rubin et al. [18] propose to use the so-called



(a) No intervention. U_{P_2} is an exogenous variable affecting P_2.



(b) Hard intervention to P_2. All arrows pointing to P_2 removed . The intervention I_{P_2} sets fixed values to $P2$.



(c) Soft intervention to P_2. All arrows pointing to P_2 holds. The intervention I_{P_2} is an additional exogenous variable pointing to P_2.

Figure 3.1: No interventions, a hard intervention, and a soft intervention presented in DAGs.

propensity score $Pr(T|X)$, the probability of being treated T based on observed covariates X (defined prior to the treatment), to match the units in a data set between treated and untreated. Pearl et al. [29] suggest figuring out causal relationships using causal graphs and based on this structure, adjusting the calculations for the effect with certain weighting rules. All of these approaches involve some domain knowledge about covariates and their potential relations with the treatment and the outcome. For the purposes of this work, the closest approach is the Pearlian way to use causal graphs to capture causal awareness for the purpose of inference.

By contrast, a soft intervention corresponds situation, in which the behavior of a system can not be fully controlled, but some of the behavior can be affected from outside of the system. For example, rolling out a marketing campaign to raise attention to some product in the hope it will turn out positively in the sales outcomes. The campaign does not cut any connections between factors but might give an extra boost to some of the factors i.e. arose attention levels to a product, although in the meanwhile also other factors still normally affect the attention levels i.e. common talk

about the product experiences. In this setting, someone might be interested to ask what is the predicted outcome if the planned campaign can help to arise attention to certain levels. This scenario could be modeled from observed data to show how the measured attention levels and other factors may help to predict the outcome in sales.

The latter mechanism with soft intervention enables us to describe some real-world cases where a change can happen surprisingly, unintended, or just by making an attempt to change something without any other control over the subject. This fits the motivation of this work, which is to see how the causal qualities of a model enable the model to hold in prediction after a real-world scenario of a change has happened. In the campaign example, by boosting attention levels the campaign causes a change (a soft intervention) in the underlying real data model. The interest here is to see how well the predictive model holds after the change. Also, the knowledge (whether purely intuitive or otherwise shown) about the causal direction from peopleâs attention to their intentions to purchase the product suggests boosting a change in attention with a campaign. However, in some scenarios, the attention could be achieved from positive comments but being rather a sister to the target outcome in sales, or it could be a side effect from the usage of the product then being rather a child feature to the target. In these scenarios, if and when the sibling and the child have no arrow pointing to the target, choosing to run a campaign for attention would not contribute to the sales at all, although the child and the sibling (in case of the parent is latent) may have a significant predictive role in the model trained before the campaign. This example illustrates the importance of understanding the causal graph to maintain the predictive performance over soft interventions.

3.2 Data set shift

This section looks at some of the well-known types of changes in data sets that correspond with the situation of a soft intervention. A data set shift by Quinero et al. [38] is a situation when the training and test data distributions are different. Here the training distribution means the data used to obtain a model, and here the test distribution means the data to be used against the obtained model to cast predictions for some real-life purposes. We take a look here at a few types of such data set shifts.

3.2.1 Covariate shift

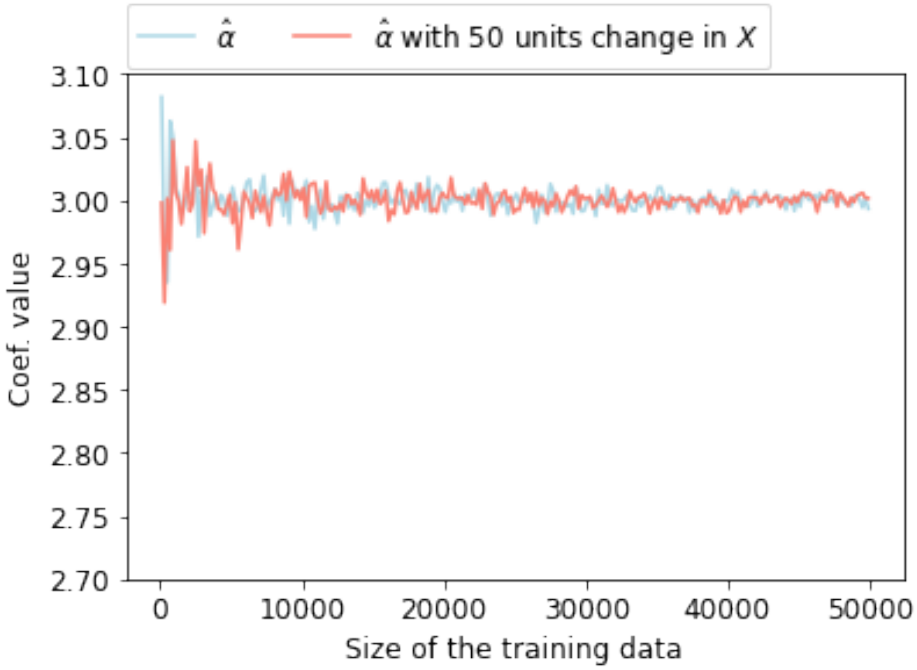
A covariate shift is a case when something changes in distributions of the input data X to predict Y [38]. We assume the input data here is in a causal pathway $X \rightarrow Y$. Training and test data are generated by a model $Pr(Y|X)Pr(X)$. Then a covariate

shift in the test data occurs if $Pr(X_{train}) \neq Pr(X_{test})$.

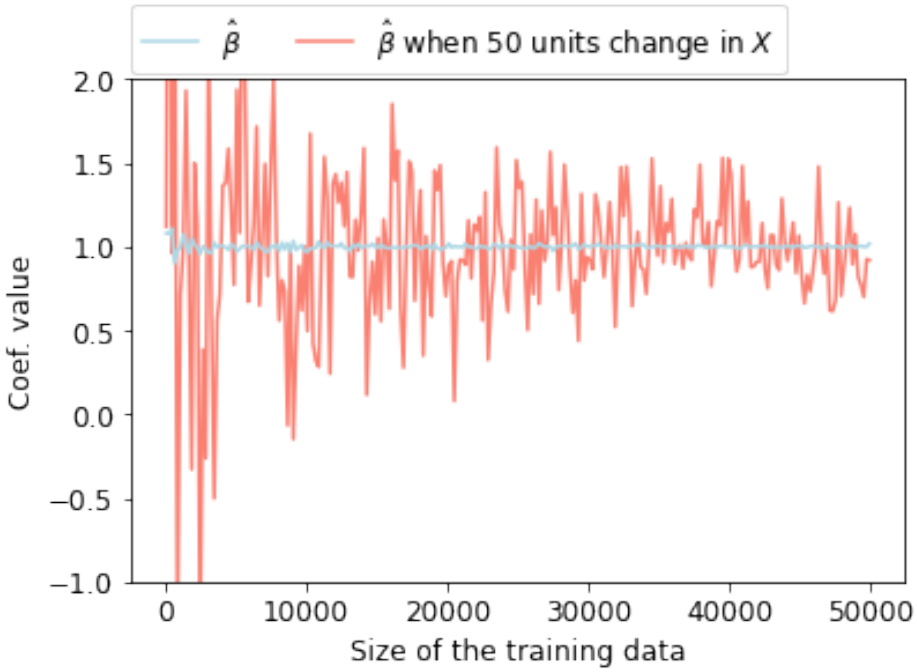
If the underlying generative model $Pr(Y|X)Pr(X)$ is estimated perfectly with the data in a training phase, then we can assume that the outcome model gives as good predictions Y_{test} in the testing phase than predictions Y_{train}, even if the data in test phase comes from a different distribution. For example, assume a linear ground-truth as generative model is $Y = \alpha X + \beta$, where $\beta \sim \mathcal{N}(\mu, \sigma^2)$. Now if the linear estimation is done with enough training data having X_{train} from a range $[a, b]$, and then obtained estimates $\hat{\alpha}$ converges to α and $\hat{\beta}$ converges to μ (the mean of β values), which would be the best possible estimates corresponding the underlying generative model. The resulting predictive model gives as accurate results to predict $\hat{Y}$ regardless if the distribution for X_{test} comes from the same range $[a, b]$, or from some other range $[a + c_a, b + c_b]$, because perfect parameter estimates would be exactly same if trained in the other range (Figure 3.2).

In practice, it might be unrealistic to perfectly estimate model parameters that enable one to catch the true nature of the underlying generative model. The more inaccurate estimates are, the more error is expected and the error is multiplied by the level of the change in X. For example, even in this same scenario, with a linear ground-truth modeled out with the linear regression technique, the more inaccurate the coefficient parameter $\hat{\alpha}$ estimated, the error in predictions will be multiplied by the distance between the training range for X_{train} and new data points X_{test} (Figure 3.3). In this scenario, the more data points are available in training data the more accurately the parameters can be estimated and hence the resulting error canceled out in this way [12].

If we consider a scenario where the ground-truth is non-linear between X and Y, but is modeled with a linear assumption, then the prediction error can increase exponentially as the change grows Consider again a causal graph now $P \rightarrow Y \rightarrow C$ and consider the relationship between the parent P and the target Y is non-linear, but the relationship between Y and the child C is a linear. Y is modeled out with linear terms. Now if there is not any covariate shift in P between training and testing data, then we can assume (by the properties of the Markov blanket) that the feature set $\{P, C\}$ gives better predictions than the $\{C\}$. However, if some covariate shift occurs in P, the resulting scenario can be so that $\{P, C\}$ is not any more functional as a feature set since the error can grow uncontrolled with non-linear ground truth between P and Y. The growing error here can not be canceled out by increasing data points in the training phase since the nature of the non-linear ground-truth can not be caught up with a linear model globally. In this case, the feature set $\{C\}$ can be a better predictor if for some reason such changes in $Pr(X_{test})$ are expected in the new data.



(a) Parameter estimation for $\hat{\alpha}$.



(b) Parameter estimation for $\hat{\beta}$.

Figure 3.2: Parameter estimations in the function of training data size with and without the covariate shift of 50 units. Despite of the shift, the parameter estimates converges close to the real values $\alpha = 3$ and the mean of $\beta = 1$.

We illustrate these phenomena with a data set generated from the following SCM:

$$\begin{aligned} P &= U_P + \delta, \\ Y &= P^2 + U_Y, \\ C &= Y + U_C, \end{aligned} \tag{3.1}$$

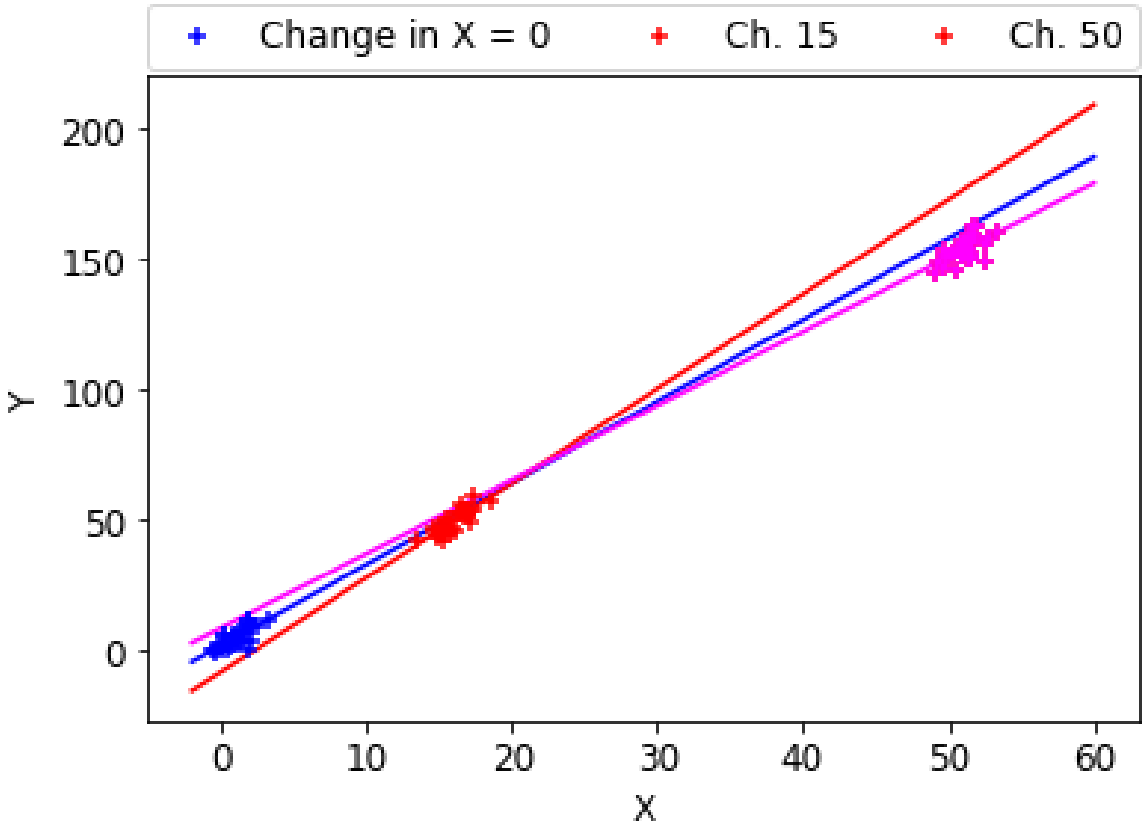


Figure 3.3: Linear regressions with different ranges in the training data (the size of the training data is now the same for every three regressions). The estimation error gets multiplied by the size of the change in ranges.

where $U_P \sim \mathcal{U}(3,4)$, $U_Y \sim \mathcal{N}(2,1)$ and $U_C \sim \mathcal{N}(3,1)$. The following two linear regression models are applied here:

$$\begin{aligned} \hat{Y}_{\{P,C\}} &= \hat{c}_P P + \hat{c}_C C + \hat{\beta}, \\ \hat{Y}_{\{C\}} &= \hat{c}_C C + \hat{\beta}, \end{aligned} \tag{3.2}$$

and they are trained with the change $\delta = 0$. Figure 3.4 shows the results as the function of δ. Even though with a mild change, the set $\{P, C\}$, which is the MB to Y, performs better, but as the change δ grows, it starts performing poorer in predictions than the set $\{C\}$. This is exactly due to the fact that the parent is actually square powered by the ground-truth, but modeled out with power one. By contrast, the causal relationship between Y and C is linear, thus using only the child as a predictor enables it to hold on better with the change. However, in this example, it is still quite a poor predictor as the change grows. This is due to another source of potential bias called *target shift* which we take a closer look at in the next section.

As a summary, in order to cope with the covariate shift, it is important to find a model that extrapolates beyond the covariate data range in the training phase [5].

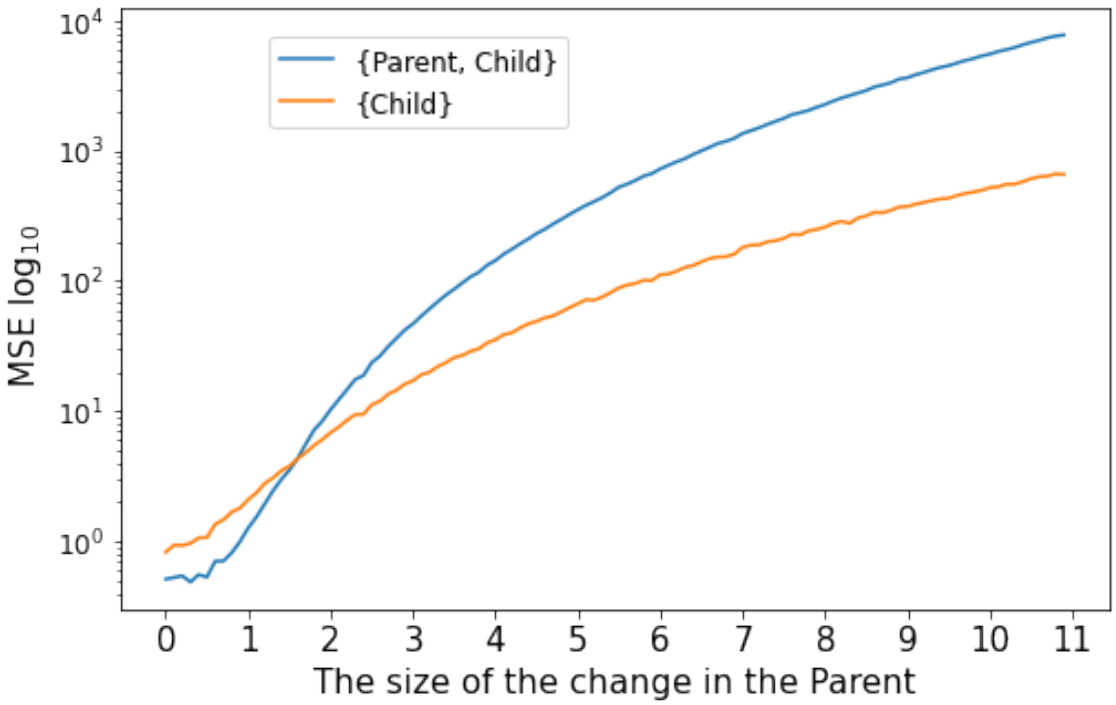


Figure 3.4: The error (MSE) in function of the covariate shift between two feature sets: {Parent, Child} and {Child}. As the change grows, the set {Child} starts to perform better. This is because the true nature between P and Y (Equation 3.1) is powered to 2, but the linear model applied here seeks only from powered to 1.

Many regression-based methods in machine learning (including linear regression methods) have the capability to extrapolate [5, 7]. Meanwhile, many other methods, albeit widely used, modern, and proven efficient in many cases, do not function well if trying to extrapolate to different ranges. For example, modeling techniques like random forest regressor (RFR) [16] and k-nearest neighbors (kNN) [3] do not have an ability to perform well outside of the training range. In this work, we focus on investigating the performance of linear regression methods on linear ground-truth data.

3.2.2 Target shift

Quinero et al. [38] discuss that the prior probability shift (a.k.a. a target shift) is a situation where the distribution of the target variable changes between the training and test phase, and hence, affects causally the descendants while everything else stays the same. The causal path here is from the target Y to a child X ($Y \rightarrow X$). The causal model $Pr(X|Y)Pr(Y)$ is valid after the change but $Pr_{test}(Y) \neq Pr_{train}(Y)$, meaning that distribution of $Pr(Y)$ is changing (for example, in a case of an outer soft intervention to Y.) Predicting Y from X is done by applying the Bayes rule [4] as

follows:

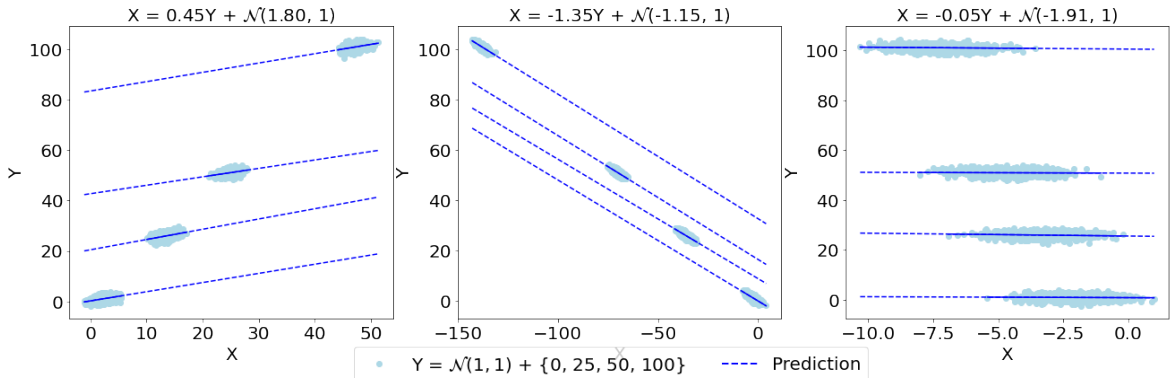
$$Pr(Y_{test}|X_{test}) = \frac{Pr(X_{train}|Y_{train})Pr(Y_{test})}{Pr(X_{test})}. \tag{3.3}$$

Now as $Pr_{test}(Y) \neq Pr_{train}(Y)$, there will be bias in prediction. As a prior probability shift occurs the incurring error in prediction is a systematic error. If there is available data points about Y_{test} the systematic error can be corrected by adjusting with the ratio $Pr_{test}(Y)/Pr_{train}(Y)$ [17] and other applied methods, for example, the prior probability adjuster (PPA) [45]. If $Pr_{test}(Y)$ is not available, it is not possible to predict $Pr(Y|X)$ anymore as accurately as before the change in Y. However, by following the logic of Storkey et al. [48], given the model $Pr_{train}(X|Y)$ and some data about X_{test} then certain distributions Y_{test} can be thought as more or less likely, Zhang et al. [56] propose a method to reconstruct $Pr(X|Y)$ by choosing a sample from training data to correspond the situation $Pr_{train}(X) = Pr_{test}(X)$. A restrictive assumption here is that data points in X_{test} are in the range of data points in X_{train}. Also, this approach requires many data points about X_{test} to get the idea of the new distribution of Y_{test}.

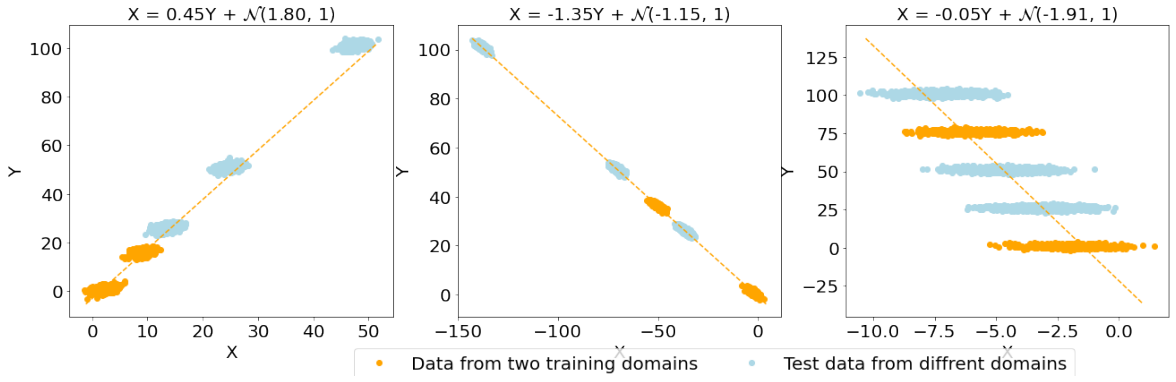
Therefore, in a situation with only one data point from X_{test}, and only having $P(X|Y)$ from one training domain, it is hard to give any predictions about Y_{test}, if a target shift has occurred. It is even harder if the new distribution of Y_{test} is not in the range of P_{train}). The error in a prediction that a target shift causes is systematic. In Figure 3.5a we see three different visualizations about target shift in Y. We can see that predictions made locally in one domain are no longer valid in other domains. The error is the bigger the change in Y has been, and it stays the same level (the distance between the prediction lines stays about the same).

However, as the error is systematic it raises the thought that could the error be corrected by having some knowledge about now how much the error rises as a function of the change level. Using only one domain for the training there is no way to infer the rate of how fast the error grows [38]. But in case there is some evidence in the training data about changes, for example, the training data consisting of two domains, then this information can be used to reveal the error rate by the change level and apply it to make predictions in new domains with different change levels. This is illustrated in Figure 3.5b with the same data as in Figure 3.5a, but now the prediction $\hat{Y} \sim X$ is made with two domains instead of one (additional domain). We can see that prediction hits much closer to domains that have encountered way higher change levels in Y. By operating in this way, it enables us to give meaningful predictions in other domains. However, it is important to note that now the prediction error in the local domain can be bigger than in the case it was modeled only with the local data points. For example, Figure 3.5a shows three different linear relationships from target Y to child X with four levels of changes to the Y. As the prediction with linear regression is done in one

domain (a change level in this case), it follows the data points well only locally, but does not scale to other domains. Figure 3.5b show the same relationships between Y and X but now the prediction is done using an extra training domain (a change level). Now the prediction scales to other test domains as well, but the accuracy is affected locally in the source domains, especially in the right-hand side example. Even though the accuracy is affected locally it is an important property that the error level is more or less invariant despite of the domains, making this kind of modeling strategy valuable if operating under a suspicion for a target shift.



(a) Model trained using only one domain.



(b) Model trained using two domains.

Figure 3.5: The tests with target shift levels. Three different linear ground-truths for the child X to Y. Predictions $P(Y|X)$ (dotted lines) with four levels of change (data point clouds). If two domains (b) are used in the training phase instead of one domain (a), then the prediction scales to all the levels of change. If trained with only one domain (a) the predictions hit systematically wrong with the other level of changes

3.2.3 Concept drift

Concept drift refers to non-stationarity in relations between X and Y [20] so that the conditional distribution $Pr(Y|X)$ changes between domains resulting:

$$\begin{aligned} Pr(Y_{train}|X_{train}) &\neq Pr(Y_{test}|X_{test}), \\ Pr(X_{train}) &= Pr(X_{test}). \end{aligned} \tag{3.4}$$

Even though X gets similarly distributed values between domains but the relationship between Y and X changes so that models estimated with the training data do not hold anymore in the new data. Thinking in a causal way, this corresponds to a situation where the structural causal model changes. For example, in the source domain SCM_S is defined as

$$\begin{aligned} P_1 &= \mathcal{N}(\mu_{P_1},\, \sigma^2_{P_1}), \\ P_2 &= \mathcal{N}(\mu_{P_2},\, \sigma^2_{P_2}), \\ Y &= r_1 P_1 + r_2 P_2 + \mathcal{N}(\mu_Y,\, \sigma^2_Y). \end{aligned} \tag{3.5}$$

But SCM_N in the new domain is different by the function that defines Y, for example:

$$Y = r_{1N} P_1 + r_{P_2} P_2 + \mathcal{N}(\mu_Y,\, \sigma^2_Y). \tag{3.6}$$

Now if $r_1 \neq r_{1N}$ then Y is affected in a different way by the parents $P1$ and $P2$ in the new data and estimated coefficients in the model do not work well anymore. Other examples are the cases when something changes in the constitution of a variable V that is not an ancestor for the Y. For example, in structures

$$\begin{aligned} I \rightarrow S \leftarrow P \rightarrow Y, \\ P \rightarrow Y \rightarrow C \leftarrow I, \end{aligned} \tag{3.7}$$

an unknown factor I that is present only in the new domain affects the sibling S (the first row) and the child C (the second row).

A concept drift can be seen as a fundamental change in the data formation process [20]. When dealing with models based on the physical laws of nature, then concept drift is not expected to happen as the laws remain the same. However, when operating with data and models in areas where the underlying real system and phenomena are unknown, or only harshly known, then the expectation for a changing concept arises. For example, in econometrics or social behavior studies the mechanism can be seen in a constant change.

In general, the concept drift is seen to ruin existing models, and the main strategy against it is to detect the change from new data as soon as possible and then update the models. In that sense, in the search for invariant prediction, the concept drift is

challenged and raises the question of can an invariant prediction be achieved at all. For this work, in the experiments in the Chapter 4, we include the concept drift cases discussed in Equation 3.7 to observe the behavior.

3.3 Invariant prediction in domain adaptation

Some research areas in *domain adaptation* focus on finding ways to adapt models trained in one or multiple source domains to make the functional in other different but related domains [40]. In domain adaptation, the change between the source and the target domains is expected to happen in distributions of features rather than in structural relationships between features or structural changes (for example, directional changes in a causal graph or changes in the real coefficients between variables). For example, a covariate shift 3.2.1 or target shift 3.2.2 can be seen as domain adaptation problems, whereas concept drift can be seen as a structural change and hence a different challenge. In some approaches in domain adaptation, the interest is to find out feature sets that would yield similar levels of prediction performance despite such a change between domains [39, 26, 41]. Peters et al. [34, 35] call this property invariant causal prediction, we call it shortly *invariant prediction*.

In an attempt to map out a feature sets between domains that allows invariant prediction performance it has shown promising results by taking into account the underlying causal structure [24, 43, 49, 50]. Peters et al. [34] showed only the direct causes remain as a set giving invariant prediction in liner ground truth setting over different types of interventions to variables and showed their outstanding performance over non-causally selected feature sets. Javidian et al. [19] focus on finding the Markov blanket for the target variables and then testing best-performing subsets of features with the blanket between domains over distributional changes. Pfister et al. [36] approach the invariant properties by defining so-called *stable blanket* within the Markov blanket that points out a stable set of features depending on which feature has been under affection between the domains.

3.4 Discussion and aligning the scope for the experiments

Direct causes as the predictor set [34] can be seen as a stable predictor set in any situation where we can assume a distributional change in model variables, except the target shift. Hence it can be seen as an overall working solution if (in one way or another) the direct causes (i.e. the parents to Y) are known and observed. However,

as a predictor set, it is not optimal in all cases. It is limited in performance due to not necessarily covering all available information about Y since it contain information only from the parental paths to Y in the causal graph. As we have seen from the Markov blanket, also children and spouses (with children) reveal information about Y. Hence, in the search for an invariant prediction, for example, using only parents, can lead to a trade-off situation about how much recognized bias is accepted due to incomplete information about Y [24].

Secondly, under a covariate shift, the challenge is to catch the true nature of the relationship between the direct causes and the target accurately enough to be capable to extrapolate to different ranges. In the case of a non-linear relationship, a poorly caught nature between predictors and the target can lead to exponentially growing error levels (Figure 3.4). Under a target shift, in which we assume some unknown outer factors starting to affect Y, then all sets of model predictors become more or less biased, and in this case, the parents alone may not be the least biased option for the predictor set. These consequences with such uncertainty about the true nature of relationships between make it tempting to involve other predictors from Markov blanket to the predictor set.

In the recent work around invariant prediction in domain adaptation, many methods have been proposed in a setting with multiple source domains. This setting allows averaged and regression-based solutions across multiple sources to yield optimal feature sets for better performance in the new test domain with changed data set [36]. However, if source domains are only one (or there is not any data available to indicate which source domain all source data points belong), we cannot rely on these averaged solutions.

For this reason, in the experiments of this work, we focus purely on situations in which we have only one source domain to train the model, and we have a new test domain that is changed by one feature (in the MB to Y) at a time. We walk through these changes case by case and test which feature set yields the best performance in the new data and in which cases we have a feature set giving also invariant prediction despite of the change levels.

3.5 Defining the change

In this section, we define the settings and the assumptions to base the experiments. We also make a formal definition for the change and the measurements to compare the results.

3.5.1 Settings for the experiments

The purpose of the experiments is to predict the target Y in a new domain with a model trained in the source domain. Here are listed the settings and assumptions for the experiments:

1. The causal graph is known and it stays invariant after the change in the domains.
2. We test a change basically for the features belonging Markov blanket to the target or the target.
3. The change to a feature is technically conducted as a soft intervention as explained in Section 3.1.
4. Only one change to one feature at a time is examined.
5. The change is tested on different levels, from a minor to a huge change. Our primary focus is that change is a constant. However, we also include testing the change in variance with zero mean.
6. A change is seen a sudden or unexpected. No data points are available from the new domain to infer anything about the change or to detect it anyhow. Only one data point is available about features to predict the target.
7. The focus is to investigate how different feature sets perform after the change. The search for the best suitable modeling algorithms are not covered here. The experiments use always the same modeling technique (which is linear regression if not otherwise stated).
8. In order to correctly measure the bias across the changes in the domains, we consider that theoretically lowest possible bias after a change can be different than before the change since the constitution of the data set has changed so nothing guarantees that MSE levels are the same. This involves calculating the theoretical bias in changed data set with data points available.

3.5.2 Formal definition

Consider a data set D^S from the source domain S generated by causal model SCM_S. By the definition of the Markov blanket, to predict $\hat{Y}$ in a case where the new data is coming from the same domain S the model giving the smallest prediction error is generally achieved by training the model with the feature set MB to Y with data in D^S. We calculate and write the prediction error then as

$$e^S_{S,MB} = [Y^S - \hat{Y}^S_{MB}]^2. \tag{3.8}$$

Here the subscript in $e_{S,MB}$ means the model is trained with data in the domain S with all features in the MB involved in the modeling. The superscript in e^S says the model is applied in the domain S to calculate the prediction error e. The subscript in $\hat{Y}_{MB}$ says that all features in MB were involved in the model to predict $\hat{Y}$ and the superscript $\hat{Y}^S$ says the model is applied in the domain S to predict $\hat{Y}$.

Now consider new data D^N is coming now from a new domain N generated by model SCM_N, where SCM_N and SCM_S share the same causal graph $\mathcal{G}$, but something might have changed in the formation of features i.e. in the functions between variables. Then again, the model giving the smallest prediction error from data D^N is generally achieved by training with MB to Y with data in D^N and we calculate and write the prediction error then as:

$$e^N_{N,MB} = [Y^N - \hat{Y}^N_{MB}]^2. \tag{3.9}$$

By this we assume, if trained with sufficiently many data points, the smallest error in prediction in D^N can be achieved with $e^N_{N,MB}$[24]. In the other words, training with MB to Y with the new data is best possible for getting the smallest prediction error in the new data. This serves as a theoretical ground-truth for the smallest possible error with the new data, and generally, it holds [24, 19]:

$$e^N_{N,MB} \leq e^N_{S,MB}, \tag{3.10}$$

which states that model trained in the new domain yields generally smaller prediction errors in the new domain than had trained in any other domain. This theoretical ground-truth $e^N_{N,MB}$ is not available with a real-world data, but it is available with a simulated data set in the experiments. This allows us to compare achieved prediction errors to this ground truth to retrieve the real result for each testable predictor set in the experiments. Next, we look at how the analysis of prediction error is done with help of this ground-truth available.

3.5.3 Transfer bias and incomplete information bias

Magliance et al.[24] use the ground-truth $e^N_{N,MB}$ (available with simulated data) to define two kinds of bias that can be used to analyze the prediction performance of different sets of predictors in case of a domain change. The biases are calculated for a set of features A. In our experiments, we have always $A \subseteq MB$. The first type of bias is called ***transfer bias*** (TB), which occurs when a model's parameter values obtained by training in D^S are no longer as valid to give the best prediction for $\hat{Y}^N$. Transfer bias TB for a set of features A is defined as:

$$TB_A = e^N_{S,A} - e^N_{N,A}. \tag{3.11}$$

The second type of bias is called ***incomplete information bias*** (IIB), which occurs when a selection of features is not holding all features that can provide information about Y. In the context of domain change from S to N the incomplete information bias IIB for a set of features A is defined as:

$$IIB_A = e^N_{N,A} - e^N_{N,MB}. \tag{3.12}$$

If $A \subset MB$, and thus not holding all relevant features to predict Y in one domain, then some IIB is expected to occur even if the domain was not changed. If $A = MB$ to Y, then IIB is not expected to exist at all.

The ***total bias*** for a feature selection A is then $B_A = TB_A + IIB_A$. The selection of A for predictions then affects the size of total bias in a case of domain change. In practice, the selection of A is often a trade-off between TB and IIB. For example, by choosing to use all observed features that belong to the MB for Y we ensure zero IIB, but in the meanwhile, TB can be huge. If only a selection of the parent variables is chosen, as suggested by Peters et al.[34] for A then we can minimize the TB but more uncertainty remains in Y, i.e. due to a latent parent, the larger will be the IIB. In other words, available parents may not be enough powerful to define Y properly.

The nature of IIB is that if they occur between domains, the bias is systematic, meaning that the bias cannot be canceled out by increasing the sample size in the training phase, but as a turn, the size of the change does not generally affect IIB, making it often the invariant part of the total bias. However, the size of a change can affect the proportions of TB and IIB in the total bias. In the next chapter, TB and IIB are used in action to analyze the predictive performance of each subset $A \subseteq MB$ to Y under different types of changes between the domains. To be better suitable for comparisons in the experiment, we calculate IIB and TB by using *mean squared error* MSE instead of the squared error SE (used in Equations 3.8 and 3.8).

4. Experiments

In this chapter, we describe how the simulations are conducted, and what is included in the experiments. After that results are presented and discussed for each case tested. At the end of this chapter, we summarize the individual results and discuss about the common points found.

4.1 Experiment setup

In the experiments, we create a causally defined randomized data set with a help of the structural causal model definitions with corresponding DAGs 2.1. The change in a data set is then simulated by doing a soft intervention to the structural causal model and then generating the new test data set. Basically, the test cases are different change levels in the nodes in the Markov blanket, and in each case, we test how different feature sets, which are subsets of the Markov blanket to Y, perform to predict target Y when trained in before any change. The results then include the comparisons of the subsets of features measured in the biases introduced in Section 3.5.3. The results in these measures are represented with some box plot data visualizations. In the next sections, we explain these procedures in detail.

4.1.1 Data simulation procedures

We use causally formed simulated data to represent the data both in a source domain and in a new domain after a change has occurred. In the formation of a causal data set D we apply a structural causal model [30] to generate data sets having real causal relationships between variables. D consists of two subsets: D^S is the data in the source domain, D^N is the data set in the new domain.

We first look at how a source domain data set D^S is constructed. Consider SCM_S as a structural causal model that constructs data set D^S. Each SCM_S has a causal graph G (which remains the same throughout an experiment) and a set of functions F_i to define each feature V_i, where i indexes the feature in question. If not otherwise stated, a linear continuous value for each feature is applied, and hence the function F_i

for the value of each feature in G is defined as:

$$F_i : V_i = \sum_{V_j \in Pa(V_i)} [c_{ij} V_j] + \epsilon_i, \tag{4.1}$$

where V_i states the feature in question, $V_j \in Pa(V_i)$ are parent nodes for V_i, term c_{ij} is a coefficient for each parent of V_i and ϵ_i is an additional random variable to add noise to each feature. Term ϵ_i represents an exogenous parent variable for V_i and it is unobserved (not available for training and prediction purposes).

In data formation for each i and j, the value for c_{ij} is drawn from an uniform distribution $\mathcal{U}$:

$$c_{ij} \sim \mathcal{U}(a, b), \tag{4.2}$$

where a and b define a continuous range for the value. By default the range is from $a = -2$ to $b = 2$. ϵ_i is drawn from a normal distribution $\mathcal{N}$:

$$\epsilon_i \sim \mathcal{N}(\mu_{e_i}, \sigma^2_{e_i}), \tag{4.3}$$

where the mean parameter is drawn from $\mu_{e_i} \sim \mathcal{U}(-1, 1)$ and the variance $\sigma^2_{e_i} \sim \mathcal{U}(0.1, 2)$.

A data set D^N is created to represent corresponding data in D^S after a change to a feature has occurred. In the formation of D^N the SCM_N is otherwise the same as SCM_S (including the same parameter values for σ^2, μ_{e_i} and $\sigma^2_{e_i}$) in the formation of D^S, but with additional term added into the function definition to the feature that is affected by the change. So in the new domain, the feature that is affected V_d has the following function F^N_d:

$$F^N_d : V_d = \sum_{V_j \in Pa(V_d)} [c_{dj} V_j] + \epsilon_d + \delta, \tag{4.4}$$

where δ represents the additional change in variable V_d. δ takes it value from normal distribution $\delta \sim \mathcal{N}(\mu_\delta, \sigma^2_\delta)$, where μ_δ and σ^2_δ are fixed constants for each experiment. Thus a constant change δ_{Ch} is done with $\sigma^2_\delta = 0$ and $\mu_\delta = Ch$ and a change δ_{Ch} in variance is done with $\sigma^2_\delta = Ch$ and $\mu_\delta = 0$.

The functions of other features in D^N remain the same as in D^S. However, the values of other features are affected if the feature V^N_d is their ancestor in the causal graph. In this way, a change in some features is causally inherited throughout the data set. This corresponds exactly to the setting with a soft intervention as explained in Section 3.1.

4.1.2 Modelling and feature selection

In the experimental part, linear regression methods are applied (if not otherwise stated). The objective is to find the best-performing set of features to predict Y^N

with a linear model trained in D^S. As reasoned in Section 2.5, as each feature belonging to the Markov blanket to Y is observed (available for modeling and prediction in a data set) then only subsets $A \subseteq MB$ to Y are relevant in purpose to find best performing predictor set [36, 19]. The performance of each model trained in D^S with a feature selection A is then measured by mean squared error (MSE) in the new domain as well as with transfer bias (Section 3.11), incomplete information bias (Section 3.12) and total bias, which is the sum of TB and IIB. Where the total bias can be seen as the ultimate measurement that defines the best performance, there MSE, TB, and IIB can be seen as the measurements to characterize and explain the observed total bias.

In the changing setting we encounter, on some occasions, some feature selection A can actually yield so badly performing predictions of $\hat{Y}^N$ that the error is higher than the predictions made with bald guesses. For that reason, a bottom line, a naive prediction is added to cut off all the feature sets performing worse than the naive one, and thus find the focus to those sets performing better than that. The naive prediction model is the mean of the Y^S to predict $\hat{Y}^N$. This is also called the prediction with an empty set of A.

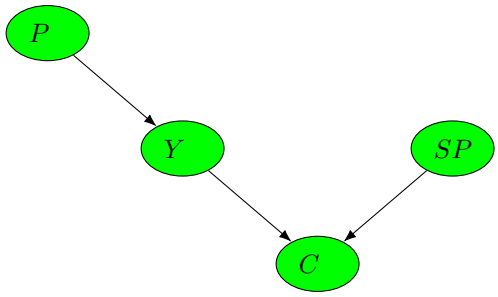
4.1.3 Conduction of experiments

As explained in 4.1.1, each data set D in the experiments is generated with SCM with drawn values from probabilistic distributions, we can expect each data set to be unique and hence the best fitting model for each selection of A is a unique as well. These circumstances involve that the chance may have a role that can affect results so that not every time the results are not the same and then comparison between different set of features can yield varying suggestions for best performing selection of A.

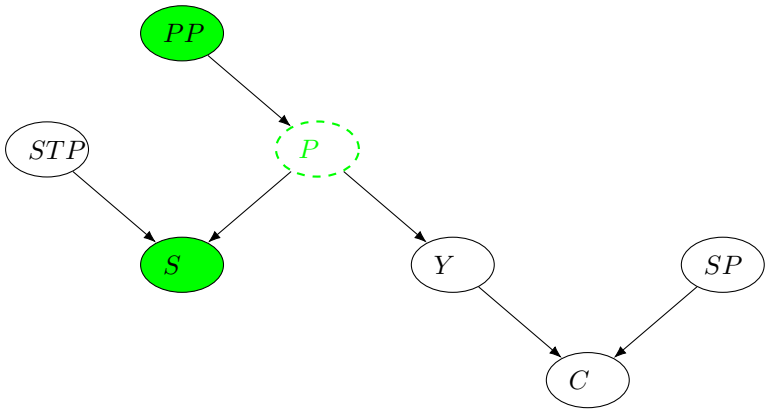
For the above reason, the conduction of the experiments applies a statistical approach. This means repeating an experiment in hand multiple times with randomly varying parameter values (4.1.1 for SCM in each trial (one repeat of an experiment) and then examining results about the performance of predictor sets statistically. This approach allows to cancel out the effect of chance (i.e. resulting data set is a special case) and may reveal also how wide variance different sets of A may have, which might be notable when evaluating different feature selection strategies. Varying over parameters allows seeing which feature sets give steadier results despite parameter combinations differing over trials. Trials are repeated 100 times for each experiment.

4.1.4 Testable scenarios

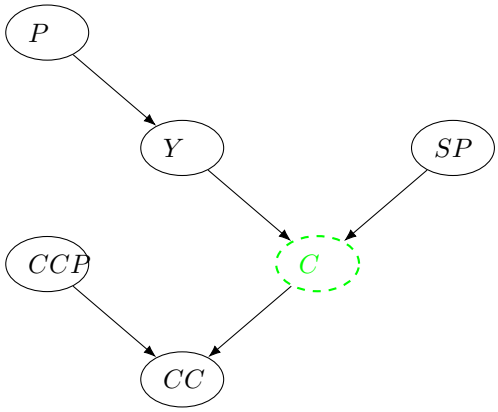
We will run experiments in two types of scenarios just like we had once introduced the Markov blanket: we have an entire Markov blanket and then we have latent cases. The



(a) Entire Markov blanket. Changes in all nodes (in green) tested.



(b) Markov blanket with a latent parent. Changes in nodes P, S and PP tested.



(c) Markov blanket with a latent child. Changes in the latent child tested.

Figure 4.1: Three scenarios of Markov blanket with changes tested (nodes in green).

testable scenarios are as follows and presented in Fig 4.1:

1. The first scenario is an entire Markov blanket that has a parent, child, and spouse to the target. We test the change separately in each of these nodes (including the target) Figure 4.1a.

2. In the second scenario we have a latent unobserved parent, a sibling (a child of the latent parent), a step-parent (another parent of the sibling), and the latent

parent's parent. We test the change separately in the latent parent, the sibling, and the parent's parent Figure 4.1b.

3. In the third scenario we have a latent child to the target, a spouse, a child's child, and a child's child's other parent. We test the change in a latent child Figure 4.1c.

4.2 Walking through the results

In the following sections, we walk through each case of a change to variables in the three scenarios of the Markov blanket by running experiments described in the previous chapter. We review results to point out the best set A to predict Y as well as make a notice about invariant prediction. We discuss the results by inferring with the rules of d-separation and make some additional tests about how much bias can be canceled out by increasing data points in the training phase. We show that in some cases bigger training data can help to reduce bias, but in some cases, the bias stays, no matter how many data points are used in the training phase. As a result, we will see which sets give invariant prediction performance despite the magnitudes of the changes. As described in Section 3.5.3, in the experiments, the comparison between sets is done by looking at the total bias. Total bias being the sum of transfer bias and incomplete information bias is then characterized by these two measurements. The base result set (for example 4.3) is a box plot visualization comparing the feature sets against different change levels (for both constant changes and changes in variance). The units for the bias are presented in log_{10} -scale in order to allow results from different change magnitudes to fit the plot.

4.3 Results from an entire Markov blanket

We first look at changes in each node of the Markov blanket to the target Y and a change in Y itself as shown in Figure 4.1a. The Markov blanket (MB) in these cases consists of a parent P, a child C, and a spouse SP to Y. The compared feature sets are the entire MB, an empty reference set (target mean) and subsets of MB: $\{P, C, \{P, C\}, \{C, SP\}\}$. For representational purposes the sets with a spouse but without a child are not included, since a spouse without a child to the target is actually d-separated from the target, and hence, it can not transmit any information about the target to improve the prediction performance.

For each node in Markov blanket to Y the results are gathered in a visualization, for example, the case Change in a Parent in Figure 4.3. It presents box plots across

the compared feature sets the total bias in the top row, the transfer bias in the center row, and the incomplete information bias in the bottom row. The measurement of a bias in the y-axis is MSE represented in $\log_{10}$-scale. The compared feature sets are on the x-axis. For each feature set are shown results in four levels of changes (the four box plots for each feature. Left-hand side graphs present the results at constant change levels. The right-hand side graphs present the results of changes in variance levels.

The estimated linear model for each testable set $\hat{Y_A}$ is

$$\hat{Y_A} = \hat{c}_P P + \hat{c}_C C + \hat{c}_{SP} SP + \hat{\beta}, \tag{4.5}$$

where $\hat{c}_P$, $\hat{c}_C$ and $\hat{c}_{SP}$ are estimated coefficient parameters and $\hat{\beta}$ is the estimated intersection parameter for the model. If some feature is not included in the feature set A then the coefficient parameter for the feature is fixed to zero.

4.3.1 Change in a parent to the target

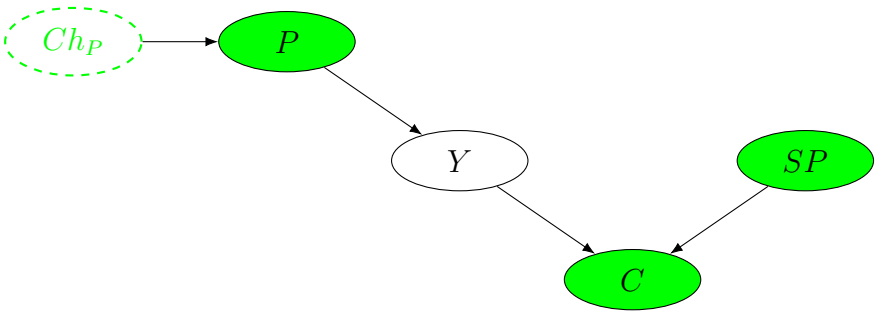


Figure 4.2: Change in the parent node P

The results of changes in the parent node, in Figure 4.3 top-line graphs, show that MB gives the lowest total biases as a predictor set in all levels of changes, and in both constant and variant types of changes. When using MB in the new domain then the IIB is zero (Figure 4.3 bottom-line graphs), but also TB (Figure 4.3 center-line graphs) is in lowest level in all instances of results. Despite that, clearly the bigger the level of change is in the parent, the bigger is TB. This result suggests that there is not any invariant set as a predictor available.

However, as the training is done with a mostly different range of values of predictors than the trained model used for predictions, then the training phase is prone not to capture the full nature of the underlying generative model and hence estimates the parameters inaccurately. This can be due to a limited amount of training data from a too-limited range of data points. In Figure 4.4, with simulated data is shown that by the increasing amount of data points in the training phase the predictions in the new domain are actually getting more accurate. We can see that the more data points

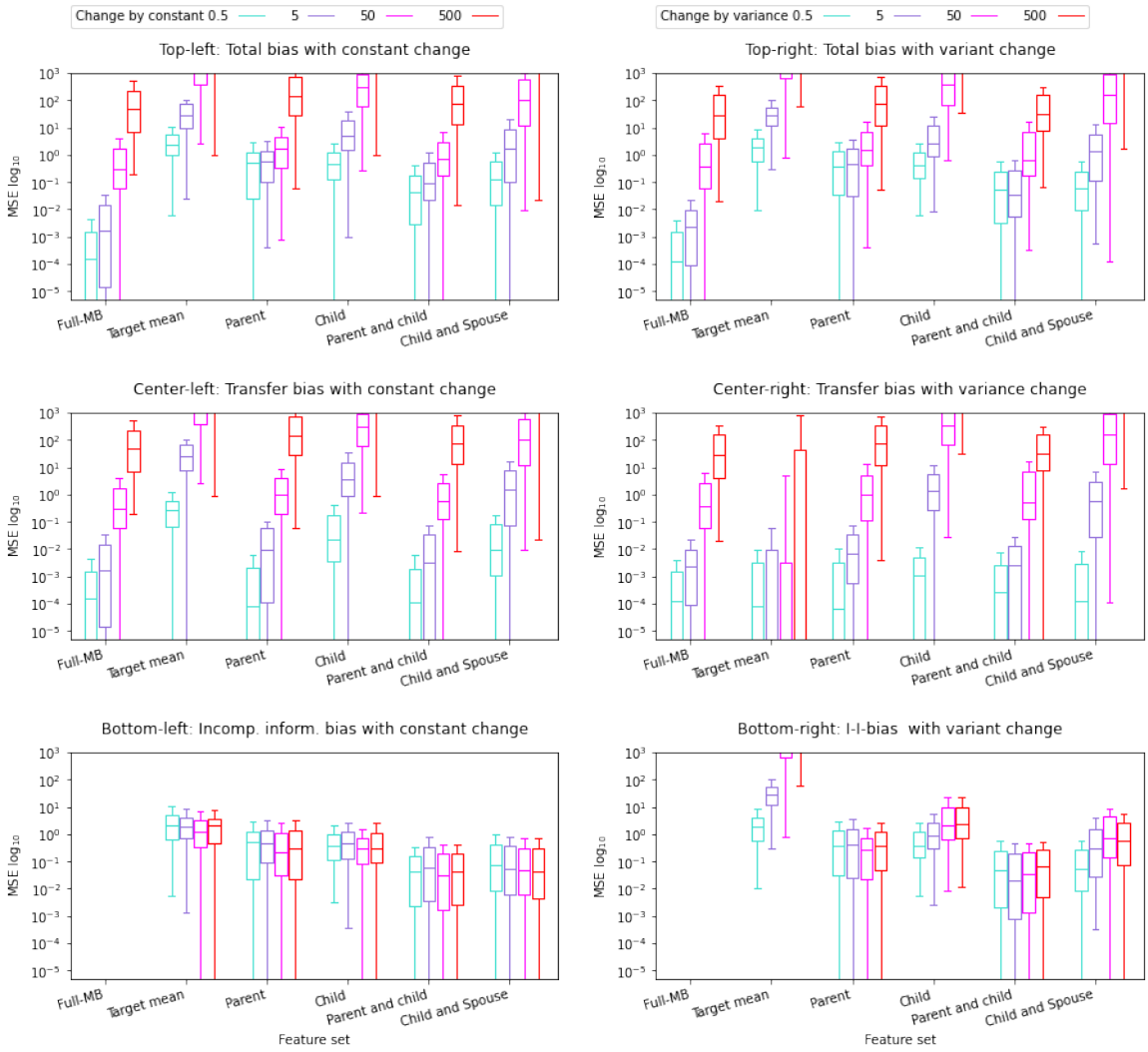


Figure 4.3: The results of a change in the parent node. MB as a predictor set gives the least amount of total bias (top-line graphs). Incomplete information bias (bottom line) is zero since MB holds all possible information about the target. With the Markov blanket, the transfer bias (center-line) seems to grow as the size of the change increases.

in the training phase, although in different ranges, the bias in a higher level of change can be reduced to as low levels as with a low-level change. As a conclusion, by the increasing amount of data points in the training phase, MB is actually an invariant predictor set in the case of the change in the parent.

Inferring with the rules d-separation, consider the change in the parent as an additional cause Ch_P to the parent P in Figure 4.2. Then we can think of this additional cause as a node with an arrow to the parent. Then the causal graph from Ch_P to the target Y would be a chain $Ch_P \to P \to Y$. By the chain rule, as P are available then Y is independent of Ch_P. Even though due to Ch_P the new domain might have values of P that are not seen in the training phase, still the function from P to Y remains the same. In the case of linear ground truth, the function is $c_P P$, where c_P is a coefficient

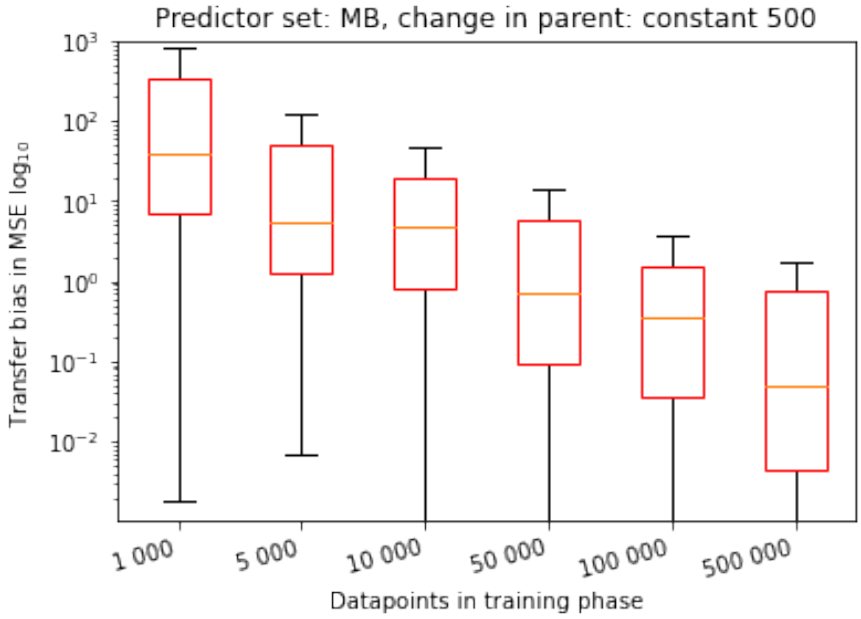


Figure 4.4: Predictor set here is the Markov blanket to target Y and data in the new domain is encountering constant 500 unit change in a parent node. As the amount of data points is increased in the training phase this will result in transfer bias dropping drastically.

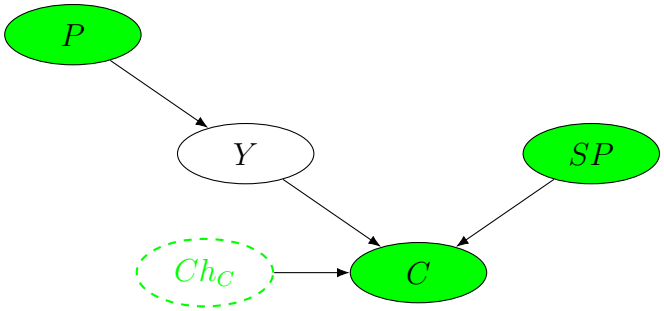


Figure 4.5: Change in a child node.

parameter. The more accurately the estimation of c_P is done in the training phase, the better predictions extrapolate to different data ranges in the new domain.

4.3.2 Change in a child

Results in Figure 4.6 show that all sets with the child node included get high transfer bias (center left and right) and it gets substantially bigger as the size of the change grows. This can be explained by the fact that as in the new domain the child will get values from different ranges then the estimation of the coefficient for the child node will make predictions about Y biased since C does not affect the Y anyhow. The other sets without C involved have invariant total bias and the best performing set is then P. In fact with set P, SP the results are almost exactly similar, but this is due to the

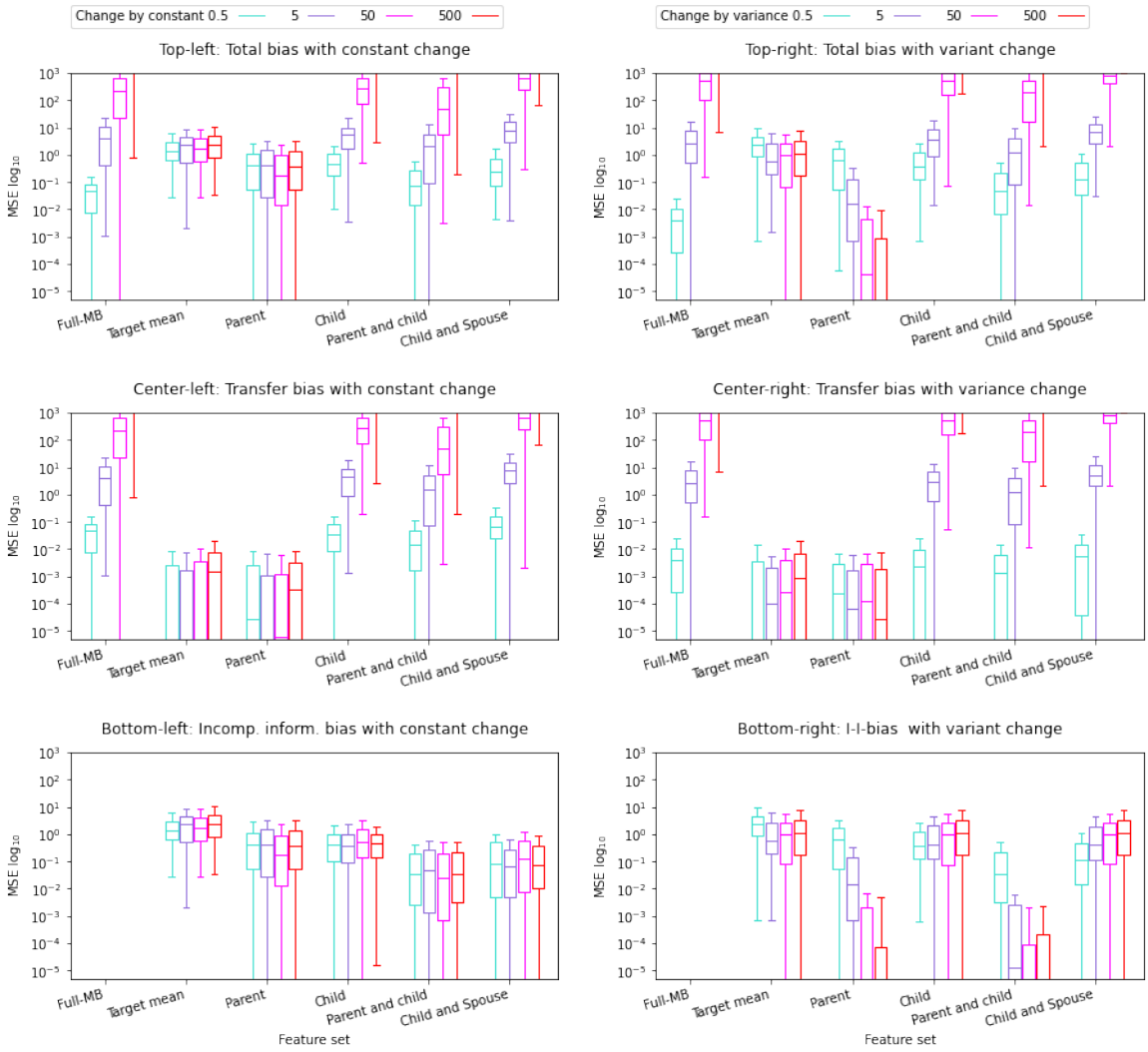


Figure 4.6: The results of a change in the child node.

fact that the spouse SP is d-separated from Y since C is not available in this set, and hence the estimated coefficient for SP tends to zero.

With the smallest level of change (0.5) we can see that Full-MB gives yet lower total bias (top-left) than set P. This is due to the fact that Full-MB is a biased model. With the child included it has zero IIB (bottom-left), and with mild change, the transfer bias is not that high yet. However, as the change crows the bias with set P stays invariant and is the best set to predict Y. With set P the TB is actually extremely low, so that amount of the total bias is almost purely due to IIB. IIB stays invariant as the change grows. In case of a constant change, IIB can not be canceled out by increasing the data points in the training phase. This is because Y is also affected by an exogenous variable U_Y and in absence of C from the predictor set, from this part, the role of U_Y to Y remains uncovered. With this case of linear ground-truth by increasing data points in the training phase with set P we can achieve an estimation of

coefficient for P in Y arbitrarily accurate. Then for the estimation of the intersection value of the linear model, we would get the value arbitrarily close to the mean of U_Y. This is the incompleteness that is captured by IIB.

4.3.3 Change in a spouse

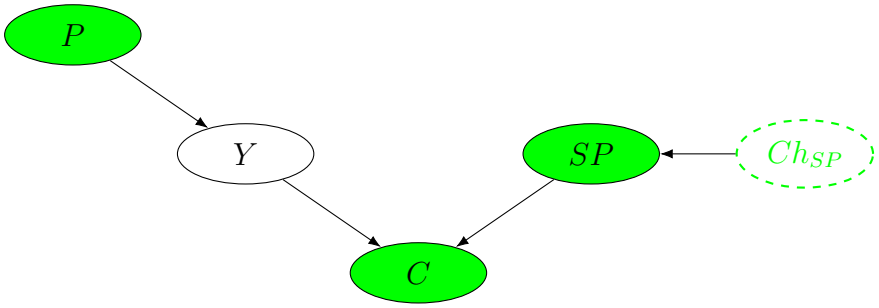


Figure 4.7: Change in a spouse node.

Results in Figure 4.8 show that only the predictor set P stays invariant as the change to a spouse grows (top-left). The set P has the same properties here as in case of change in a child, extremely low TB, encountering some IIB (bottom-left) since no information about U_Y in absence of C and SP. However, in this case, Full-MB is still the best performing set when the amount of change is at least until 5 units, with 50 units even with set P, but with 500 units of change not performing well anymore. Compared to the case of change in a child the effect of change in a spouse is drastically lower. This is due to the role of the spouse in the Markov blanket is actually to enlighten the effect of the exogenous variable U_C while the role of a child is to enlighten the effect of U_Y which is otherwise the unseen part of Y.

4.3.4 Change in the target

Results in Figure 4.10 top-left show that with a small constant change (0.5 units) the MB is yet the best predictor set. But in higher change levels of changes, the bias with MB grows fast and then the set {Child, Spouse} becomes the best predictor set. This can be reasoned so that as the direct effect of the change to Y comes into the picture, then the effect of the coefficient for the parent $\hat{P}$ is not anymore sufficient to define Y alone. The coefficient takes too much role in prediction and makes P a biased predictor in the model. However, the transfer bias (Figure 4.10 center-row) with the set {Child, Spouse} seems to grow as the change level grows and hence it can not be considered an invariant predictor set. Further examination in Figure 4.11 shows that increasing data points in the training phase does not help to reduce the bias. These results suggest it does not exist sets to yield invariant predictions in the case of a change in the target.

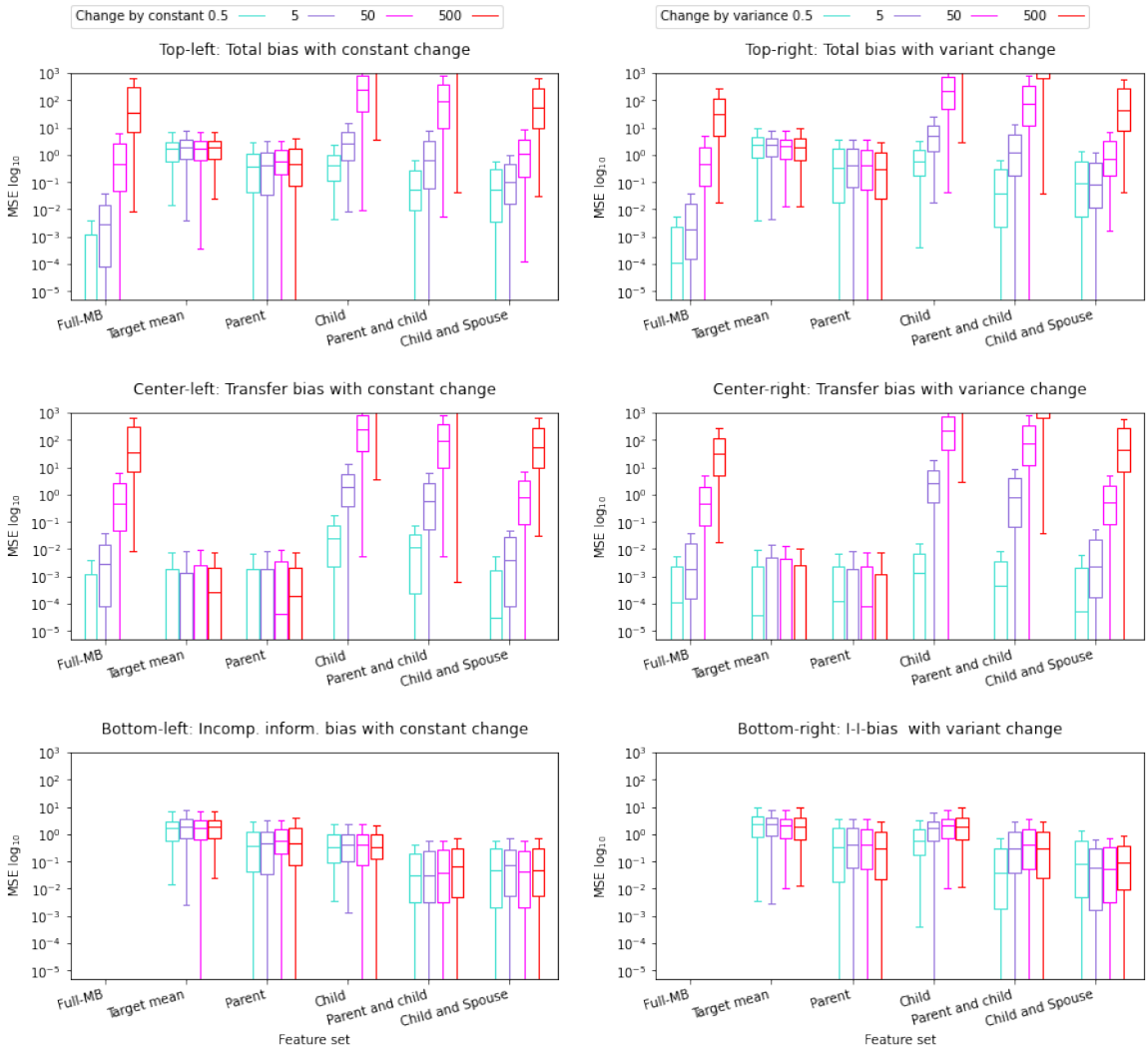


Figure 4.8: The results of a change in the spouse node.

By examining the linear model parameters estimated after different levels of changes, it was found that the coefficient values remain the same as in the source domain, but the interception value grows along with the change level. This seems to be exactly the case of the target shift discussed in Section 3.2.2 causing a systematically growing lag between the predictions and the real values in the function of the change level. As discussed in Section 3.2.2, if there were more source domains available containing information about different change levels in the target, then this lag can be fixed by modeling across all source domains, and hence a resulting capable to yield invariant predictions in the new domain with other change levels.

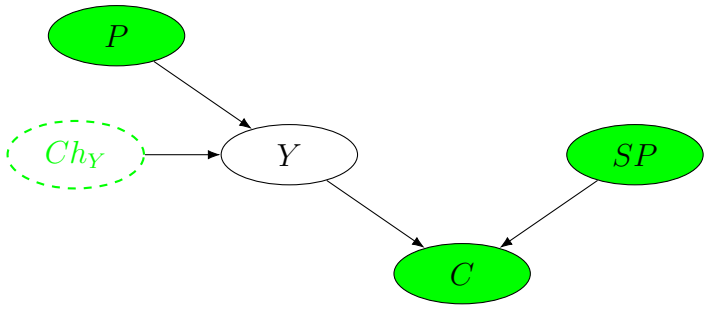


Figure 4.9: Change in the target.

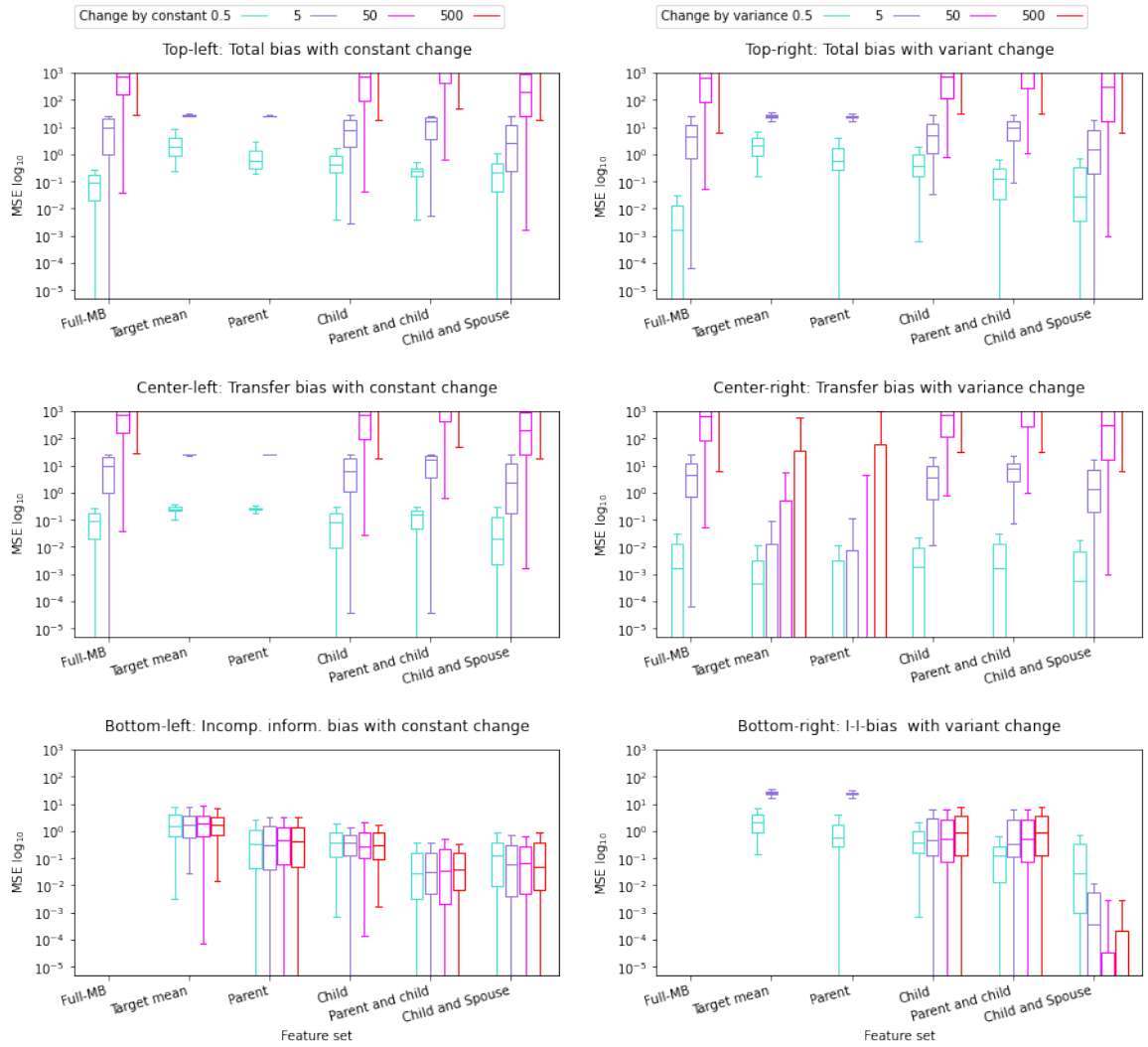


Figure 4.10: The results of a change in the target node.

4.4 Results with latent variables in Markov blankets

Next, we look at some changes in the nodes of the Markov blanket (MB) having a latent parent or a latent child as shown in Figures 4.1b and 4.1c.

Figure 4.11: A change in the target. An extreme change level of 500 units. An attempt to reduce the transfer bias for the feature set {Child, Spouse} by increasing the size of the training data. Seemingly the bias is not coming down.

In the case of a latent parent, the estimated linear model for each testable set $\hat{Y_A}$ is

$$\hat{Y_A} = \hat{c}_{PP}PP + \hat{c}_S S + \hat{c}_{STP}STP + \hat{c}_C C + \hat{c}_{SP}SP + \hat{\beta}, \tag{4.6}$$

where PP is the parent's parent, S is the sibling and STP is the step-parent, C is the child and SP is the spouse. In the case of a latent child, the estimated linear model for each testable set $\hat{Y_A}$ is

$$\hat{Y_A} = \hat{c}_P P + \hat{c}_{CC}CC + \hat{c}_{CCP}CCP + \hat{c}_{SP}SP + \hat{\beta}, \tag{4.7}$$

where P is the parent, CC is the child's child, CCP is the other parent of the child's child and SP is the spouse. Again as some feature is not included in the feature set A then its representative coefficient parameter is zero (for example, $\hat{c}_{SP} = 0$ if the spouse is not part of a set A)

4.4.1 Change in a latent parent

Results in Figure 4.18 top-left shows that the best performing set is {Step-P, Sibling, Child, Spouse} in every levels of change Ch_P. The set MB is the second best, however, sometimes the difference is not very clear, there are many trials when MB has given better estimates (see the overlapping boxes in the plot). A slightly poorer performance can be explained because the role of PP in the latent P (and then causally in Y) becomes smaller as the direct change Ch_P to P grows, and hence estimates for latent P comes biased.

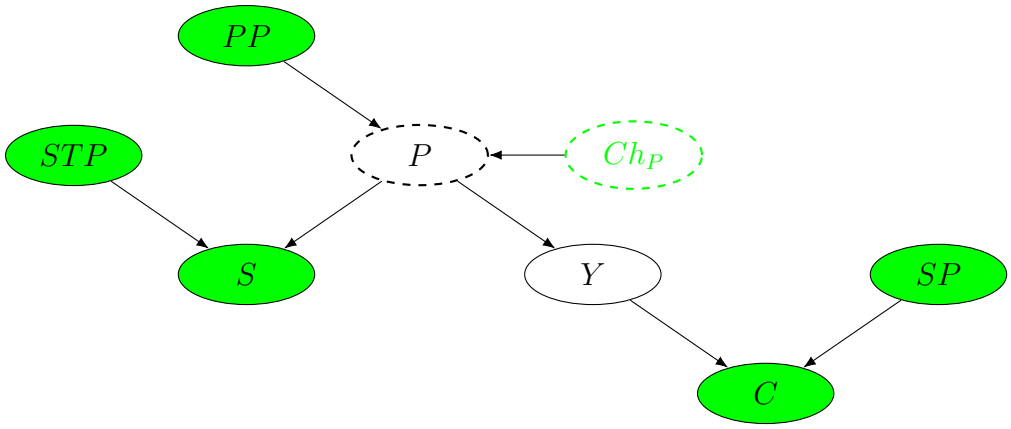


Figure 4.12: Change in a latent parent.

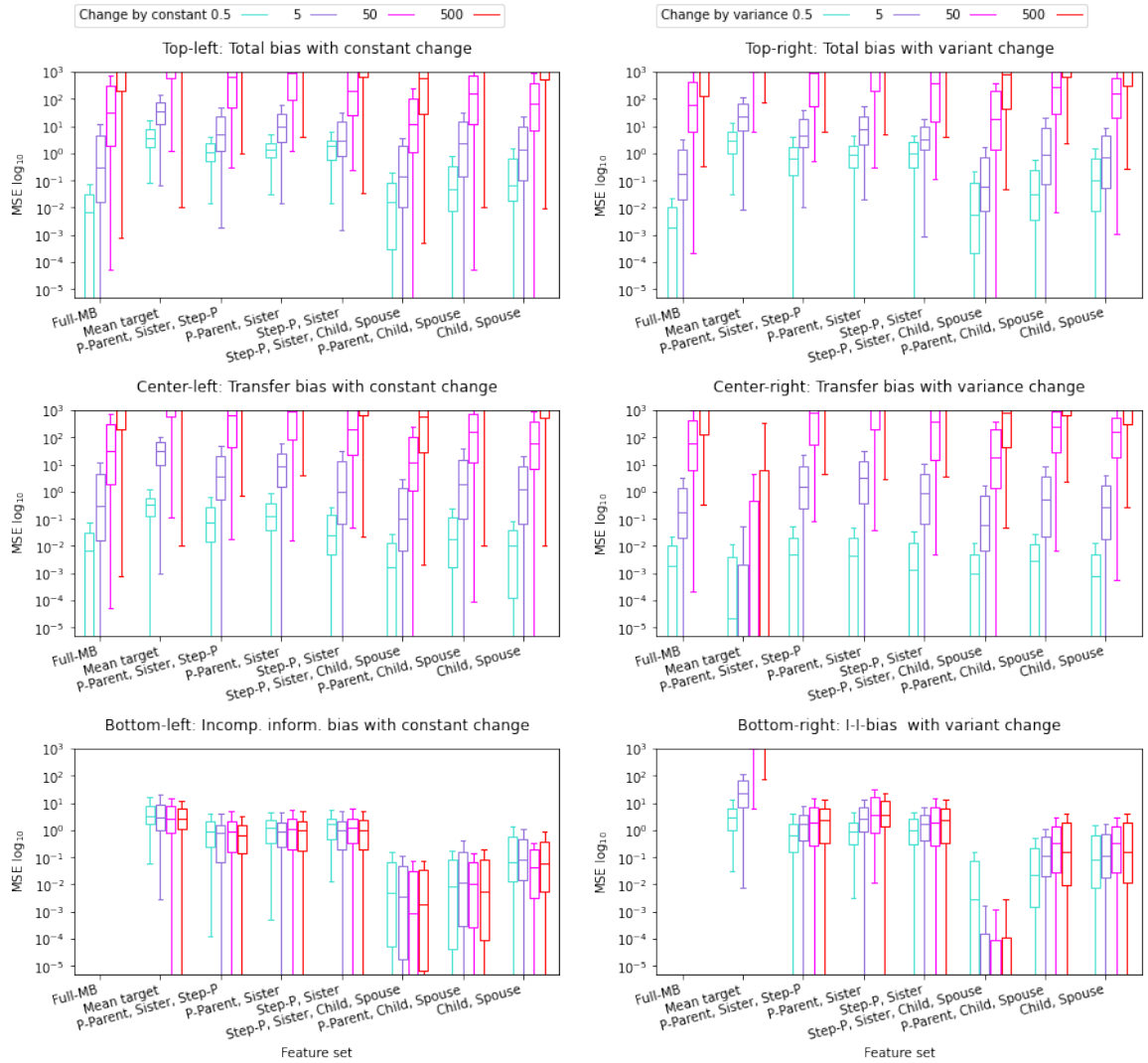
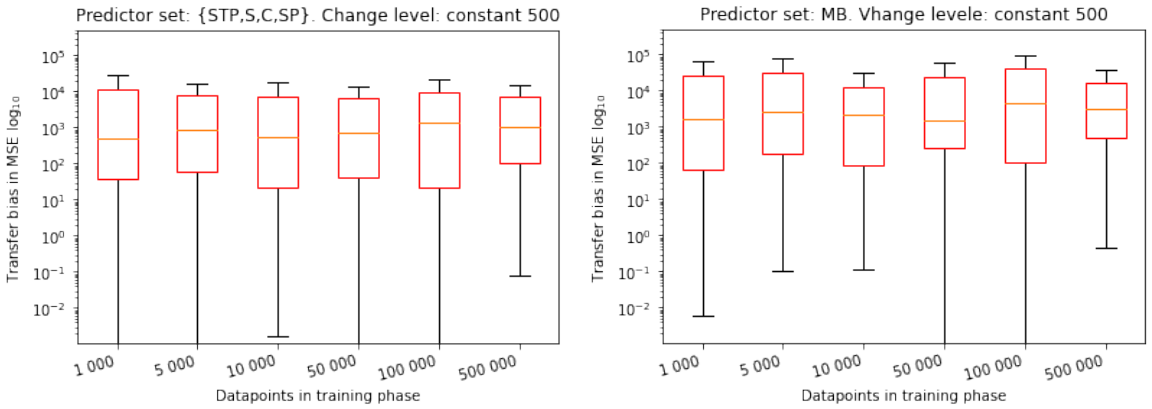


Figure 4.13: Results of a change in the latent parent.

We can see that neither the set {Step-P, Sibling, Child, Spouse} nor MB stay invariant as the change Ch_P grows, and the increasing bias can not be canceled out by



(a) An attempt with feature set: {Step-P, Sibling, Child, Spouse}.

(b) An attempt with feature set being the full Markov blanket in the scenario.

Figure 4.14: Change in a latent parent. The plots show the attempts to reduce the transfer bias by increasing data points in the training phase. The attempt fails with both feature sets (a and b).

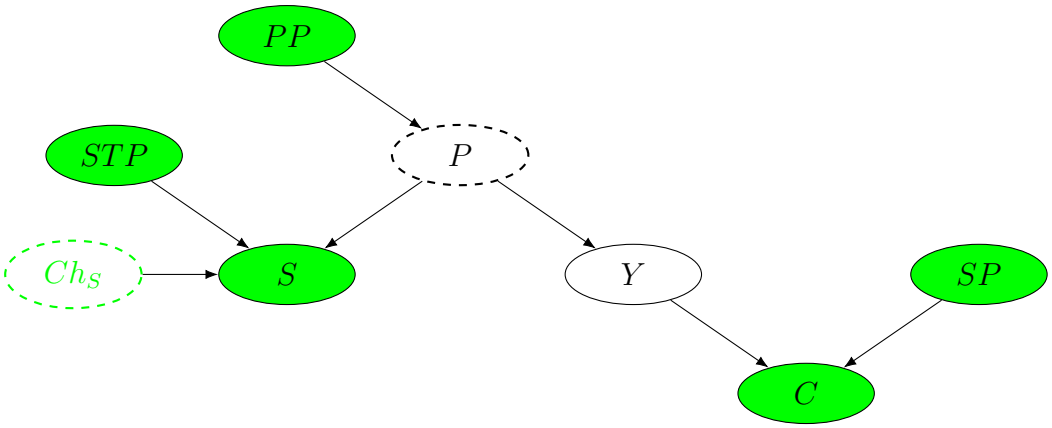


Figure 4.15: Change in a sibling with a latent parent.

using more data points in the training phases (Figure 4.14.)

4.4.2 Change in sibling with latent parent

Results in Figure 4.19 (top-left) show that the set {P-Parent, Child, Spouse} is the best performing set in from the second smallest level of changes on (5 units) and it stays also invariant over all change levels. This can be explained due to change in sibling does not affect Y and any of the predictors in the set anyhow. The only cost in bias comes with the IIB (bottom-left) since S can emit information about P but it is not involved in the set. Also sets {P-Parent} and {Child, Spouse} stay invariant, but their IIB levels are higher due to they are less complete.

The set MB can get the better result yet with the smallest level of change (0.5 units) due to zero IIB but gets biased fast as levels of changes grows due to increasing transfer bias. All sets with S included have these same phenomena since as S gets

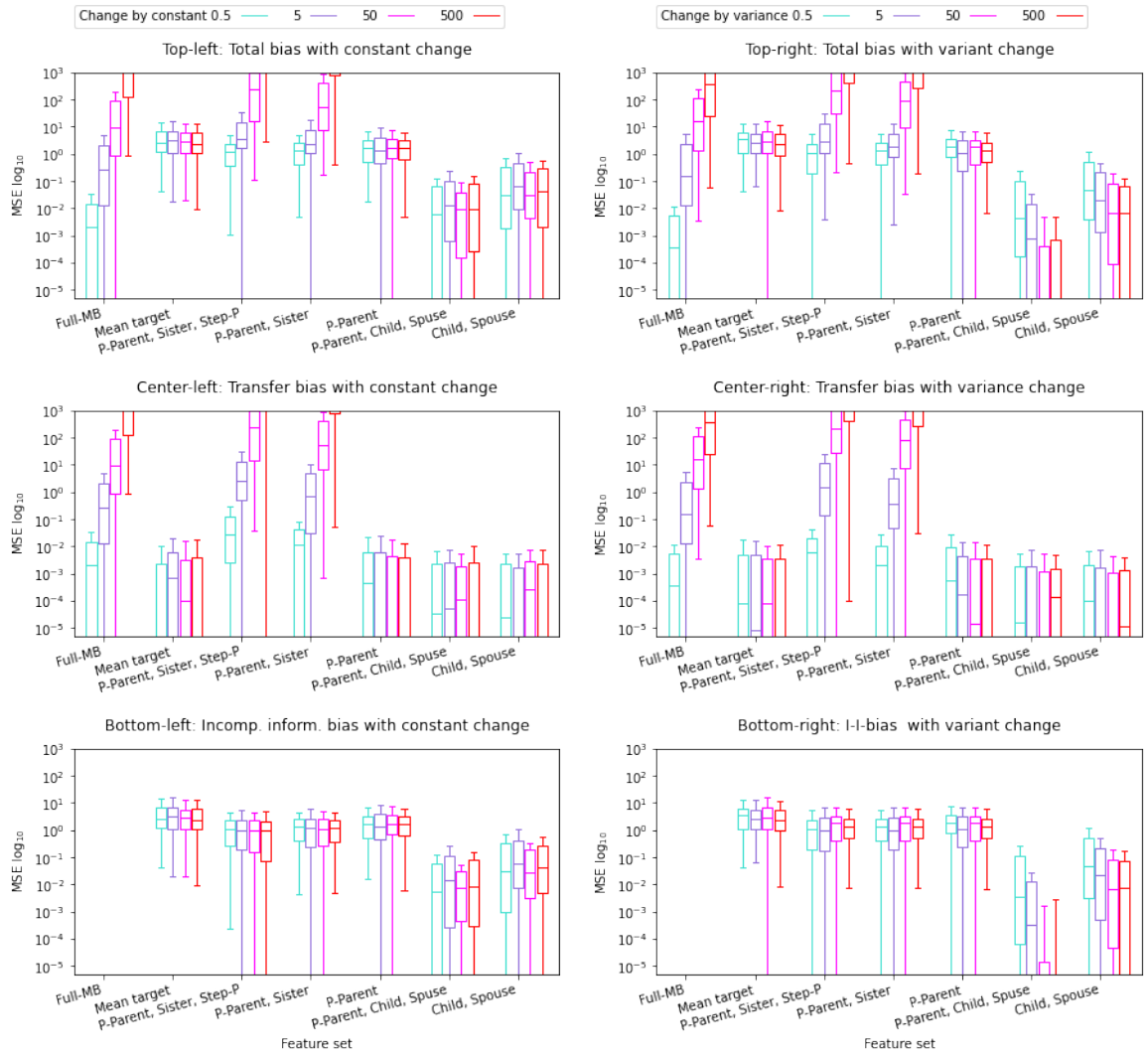


Figure 4.16: The results from the change in a sibling with a latent parent.

higher values due to Ch_S then estimated $\hat{S}$ increases the value of $\hat{Y}$ but it has not any real causal effect to Y.

4.4.3 Change in a parent's parent as the parent is latent

Results in Figure 4.19 (top-left) show that the best performing set is {P-Parent, Sibling, Step-Parent, Child, Spouse (the full MB in this case)} in all levels of change. However, it does not seems to be an invariant predictor with a constant number of training data points set as the level of transfer biases arises clearly by the change level. The results from experiments with bigger training data in Figure 4.19 show that the transfer bias is reduced to minimal, and thus, the predictor set can be considered as being capable to yield invariant predictions. The phenomena here is obviously due to the covariate shift, the same as with the change in the parent case in Section 4.3.1. Too small training

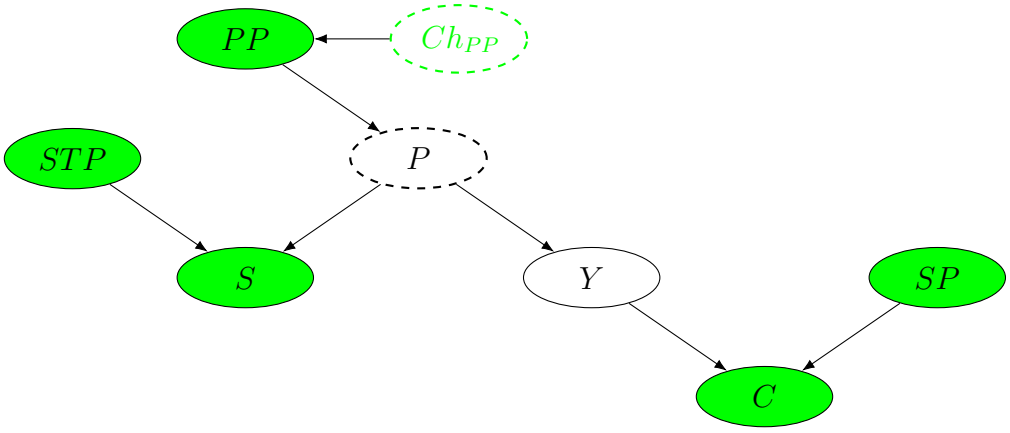


Figure 4.17: Change in a parent's parent as the parent is latent.

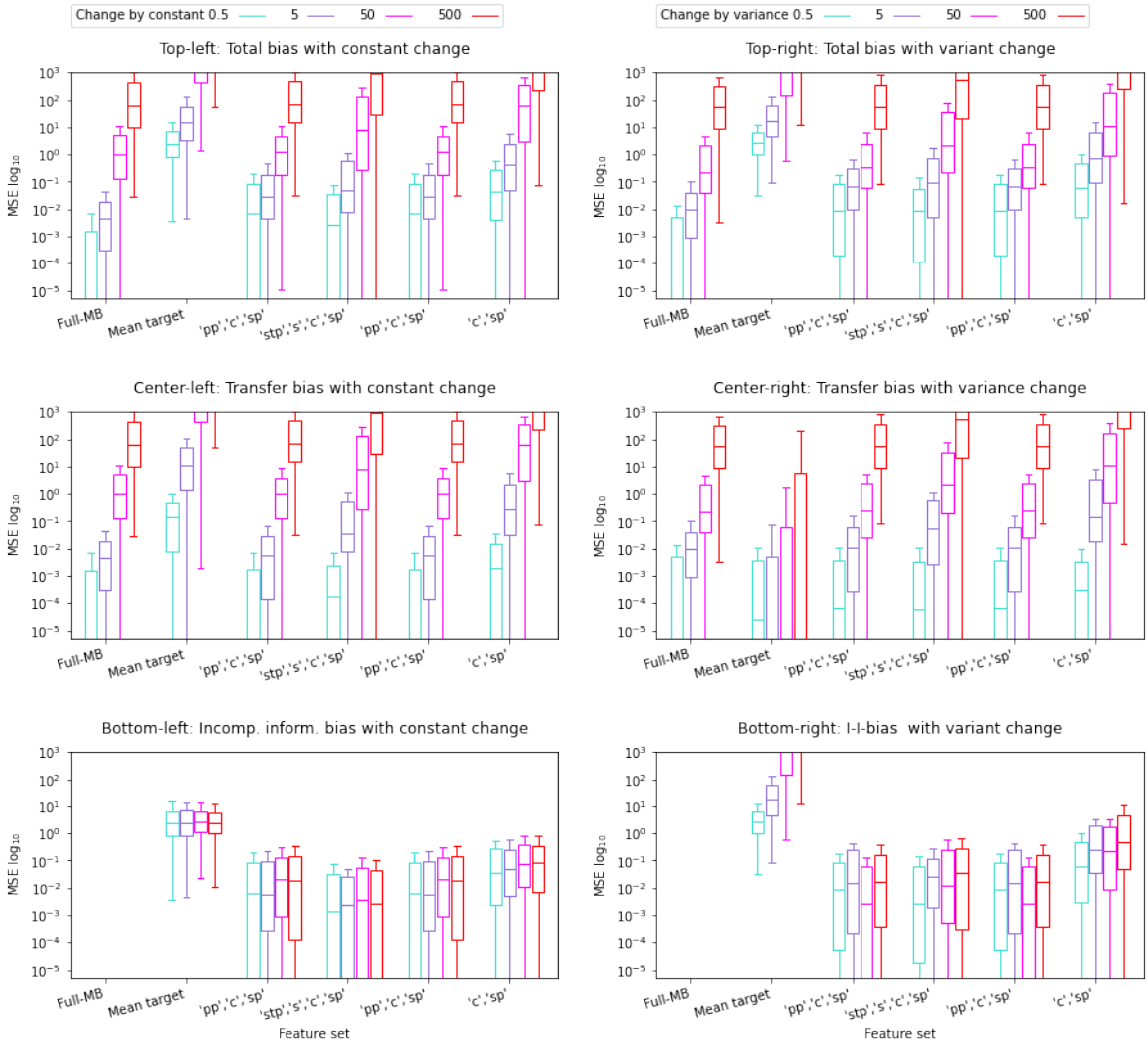


Figure 4.18: The results from the change in a parent's parent, as the parent is latent.

data can not estimate the coefficients accurately enough to predict any more accurately with test data far out of the training range.

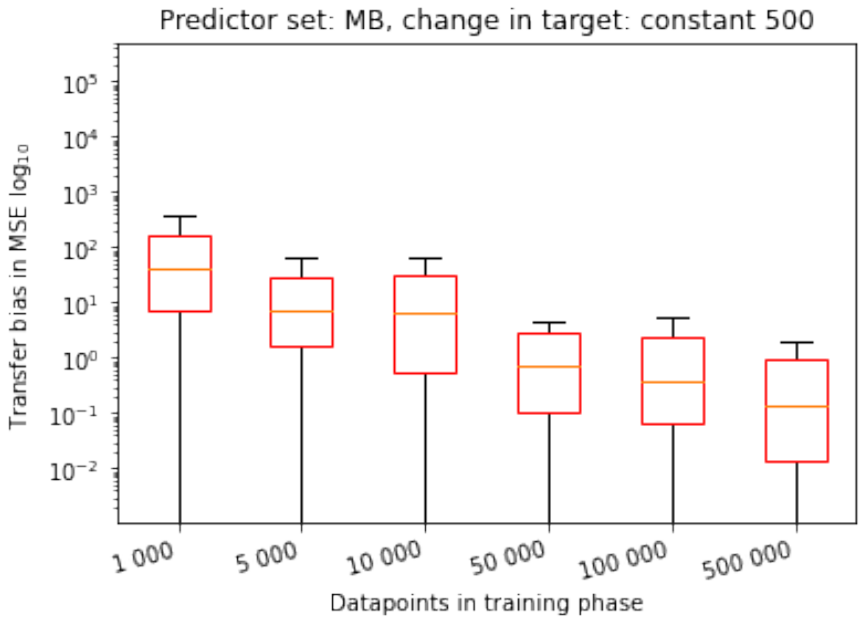


Figure 4.19: Change in a parent's parent as the parent is latent. An attempt to reduce the transfer bias success by increasing the data points in the training phase. This result shows the feature set MB capable of invariant predictions in this case.

4.4.4 Change in a latent child

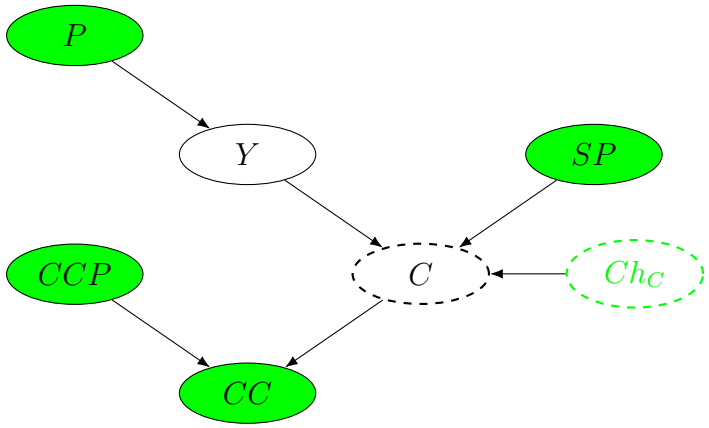


Figure 4.20: Change in a latent child.

Results in Figure 4.21 (top-left) shows that the set {Parent} is the best performing and invariant set. As the child is affected by the change, all coefficients downstream do not hold anymore and will cause the transfer bias to grow as the change grows. However, with a small change full Markov blanket can still yield better performance. As if the Child's child is not included in the set, then the spouse becomes d-separated from Y, and hence not helping with predictions. These observations correspond to the situation with a change in an observed child in Section 4.3.2.

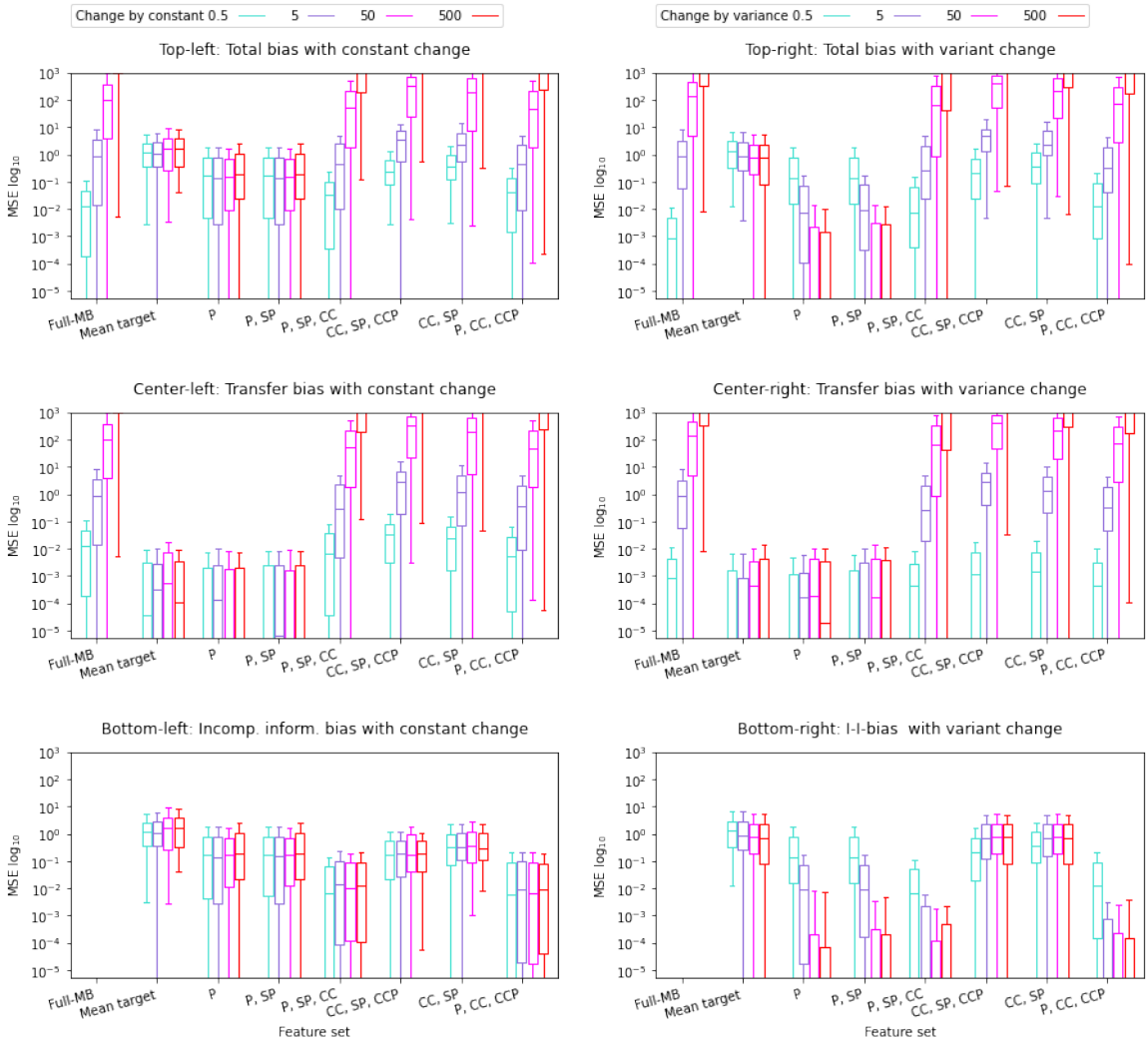


Figure 4.21: The results of the change in a latent child.

4.5 Discussion of the results

In this section, we first recap the results in summary tables. Then we discuss the overall findings.

As a summary, in Table 4.1 we see, if the subject for the change is either a Child node or Spouse node, then both of these nodes need to be stripped off from the predictor set to enable invariant prediction as a change hits these nodes. In the case of a change in a parent, the MB is the best invariant predictor set, however, it is still prone to increasing bias if the obtained model is not estimated accurately enough as the error is multiplied by the level of change. In the case of a change in the target, there is not any invariant predictor set available. The set {Child, Spouse} can yield the best possible performance. The systematic error causing the transfer bias in a target shift can be corrected if there is information available on some previous target changes

in the training data.

Change subject	Best set	Invariant prediction	Exceptions
Parent	{Parent, Child, Spouse}	Yes	Risk of bias arises as the range change due to restrictions in model accuracy
Child	{Parent}	Yes	With small changes MB still the best set
Spouse	{Parent}	Yes	With small changes MB still the best set
Target	{Child, Spouse}	No	If the training data contains additional domains about target change, then invariant prediction can be adjusted.

Table 4.1: Summary of changes without latent cases.

From Table 4.2 we see the comparison of the results from the undergone cases having a latent parent or a latent child (the first row). The case having a latent parent and that parent is subject to a change can not yield invariant prediction and even all ancestral paths (for, example Parent's parent) must be excluded for the best performance in bigger change levels. However, if the subject of a change is a parent of this latent parent (the third row) then an invariant prediction is achievable by including the parent's parent with the rest of the MB to Y. The sibling being the subject for a change (the second row) holds an invariant prediction if the sibling itself is dropped from the set. As well as the child being the subject of a change then an invariant prediction can be achieved by dropping the child and the whole branch descendant to Y. With the smallest levels, the full Markov blanket can still yield the best performance.

Each of the undergone cases of a change is somehow problematic in an attempt to predict with a model trained in a single source domain. However, results show huge differences in characteristics between the cases. In a way, it is the least problematic if the change happens to be in an observed parent to the target. Then the Markov blanket is the best set, and an invariant prediction can be achieved at least by having enough data points in the training phase to estimate the coefficients accurately enough. Having MB as a predictor set a researcher can be sure that information is not missed,

Change subject	Best set	Invariant prediction	Exceptions
Latent parent	{Step-parent, Sibling, Child, Spouse}	No	Parent's parent included (full MB) can yield better performance occasionally
Sibling of a latent parent	{P-Parent, Child, Spouse}	Yes	With small changes full-MB still the best set
A parent of a latent parent	{P-Parent, Sibling, Step-Parent, Child, Spouse (Full MB)}	Yes	Risk of bias arises as the range change due to restrictions in model accuracy
A latent child	{Parent}	Yes	With small changes full-MB still the best set

Table 4.2: Summary of changes with latent cases.

for example, if not sure if a change is going to happen. MB stays a safe choice for both cases.

Using only observed parents is the most reliable and stand-alone choice for the predictor set. It is clearly capable of an invariant prediction as the change occurs in an observed child, an unobserved child with an observed child's child, an observed spouse, or an observed parent itself. Of course, this comes with a cost of higher baseline error due to incomplete information bias. Among the cases undergone, the case of a target shift is the worst scenario as there is simply no way to find a good predictor set as the systematic bias runs in by the change in the target. However, predicting with the covariates related to the descendants helps to stay on track for a while. If the setting is extended to have information about the target change in other domains, it helps drastically to cope with the systematic bias by adjusting the regression across all domains.

Having an unobserved parent sets also trickier circumstances in feature selection wise. The Markov blanket, in this case including a sibling, a step-parent, and a parent

of the latent parent, is an invariant choice only if a change hits the parent's parent. Meanwhile, if a change hits the parent itself or the sibling, then MB is no more a good choice at all. A change hitting to a sibling sets a dangerous scenario if the sibling is included in the set. A sibling can have a higher correlation with the target than with the parent's parent making the sibling a tempting choice to be included. However, choosing only the parent of the latent parent from the upstream to the set is kind of a safer choice. It handles invariant predictions in cases a change hits the sibling or the parent's parent itself.

As a change hits the latent parent itself then there is no way to achieve an invariant prediction with any possible set. This kind of change ruins the help of replacing with the family member variables like a sibling, step-parent, and parent's parent to cover for the missing parent.

5. Conclusions

In this thesis work, with the experiments, we have investigated changes to different nodes around the Markov blanket to the target variable and measured the prediction performance with models trained before the change with different subsets of the Markov blanket variables. The change tested here is technically a soft intervention type of intervention as discussed in Section 3.1. The change corresponds to a situation where a feature is affected by an intended or unintended outer factor and the resulting data set is thus an observational data set. The performance was measured with the transfer bias and incomplete information bias to be suitable for after-change settings as discussed in Section 3.5.3.

In the results, we paid special attention to pointing out, for each type of change in question, if the property of invariant prediction performance was achievable. In order to make such a notice properly, throughout Chapter 4, we made some additional tests with different sizes of training data points. The results showed us in some occasions (for example, the covariate shift) the transfer bias could be canceled out by increasing the training data, while in some occasions it did not help to reduce the transfer bias (for example, the target shift and the change in a latent parent). All the results, in the details and together, were discussed and outlined in Section 4.5.

In this chapter, we highlight a few points from the results to discuss the meaning from a wider perspective. We also conclude the limitations of this work and some topics for further studies in this subject.

5.1 Baseline instructions for the feature selection

If the level of a change is expected to be tiny, then the Markov blanket is a good choice as a feature set, if some sort of extra bias is accepted. As with MB we can guarantee zero incomplete information bias despite the change levels to come. This advantage can keep the total bias minimal compared to other feature sets as a tiny change runs in. Using MB also relaxes the need for different strategies depending on the affected node.

If the expected change level is higher then causal understanding can be leveraged.

We might either choose only the parents as predictors as suggested by Peters et al. [34] to enable an invariant prediction feasible or with the help of special expectations about which node is prone to a change, we can choose the stable blanket for each case as suggested by [36]. In the former choice, we accept a certain level of incomplete information bias, but in the turn, we release from thinking which node is affected (except the change in the target node). For the latter one, we seek less incomplete information bias. The challenge then is to avoid using nodes around affected children and spouses by cutting the affected branch off from the feature set. The same applies to using siblings in the case of latent parents. This is important because, in a way, siblings can be actually very common features to be used. We will discuss this yet in Section 5.3.

If the affected node is the target itself, then the systematic transfer bias in predictions is inevitable. The only way to cope with this situation is to have some information in training data about the changes in the target before. This would allow to make regression across past domains and thus correct the systematic lag (3.2.2). It is notable that just some information about earlier changes in the target can help to correct the lag substantially.

5.2 Dangerous latent parents

A notable finding is that as the change hits an unobserved parent, then the invariant prediction can be achieved only if there is another observed direct cause to the target. So any other variables around the latent parent can not support anymore after the change in the latent one. We suppose further that if another latent parent without any observed near covariates (parents or children of this node) is hit by a change, then the emerging situation corresponds to what is target shift, and thus, any invariant prediction is not possible either.

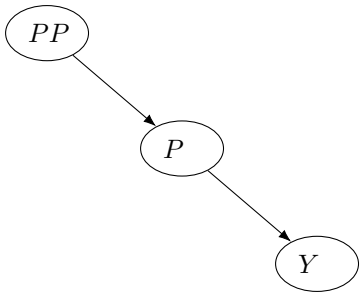
In a conclusion, if a researcher has a clue that an unobserved parent is under threat of changing, then using the covariates like the parent's parent or the parent's other children (siblings to the target) is not a good choice. However, if other parents are observed, then a better strategy would be to discard the covariates of the unobserved and rely purely on the observed parents. This is in line with what Peters et al. [34] suggested.

5.3 Real parents are essential, but theoretical

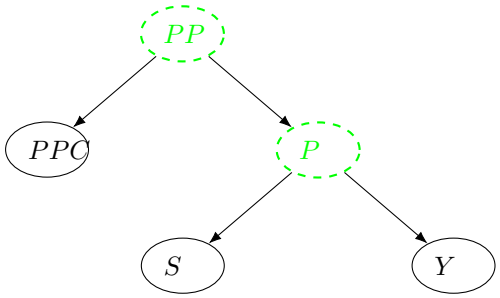
In general, reflecting on a fundamental problem occur whenever measuring out the values of interest. A measured value is merely an image of the subject in the interest

[11, 25]). On occasions, the measured value can be really close to the subject. For example, going to the example we had in Section 3.1, the price of a product can be seen to be a direct cause of some sales figures. The real price of a product is probably close to the values we would have in a data set gathering product prizes from a market. Maybe some failed records, but otherwise, everything is in line. Meanwhile, other types of features can be way more biased. For example, consider customer satisfaction is been thought to be a direct cause of new sales figures. The satisfaction is then obtained by rolling out a customer survey. The survey method itself could make the outcome value biased already, but also the satisfaction is a vague value to answer correctly for the respondents, and there might as well be outer factors affecting how the respondents give their answers. So having results from such a survey, we would rather have some variable giving us some guidance about real customer satisfaction than something in concrete.

From the point of view explained, anyone could rise a question like: Can we ever assume a direct causal, and thus, a parental relationship to a target? Aren't there always other factors affecting the measured value? In other words, the speculation is based on the thought that all observed measurements about variables in interest are more or less noisy, for one reason or another. In this case, the actually measured variable can be seen as merely a child for the subject variable (i.e. the results from the survey) it represents, and the subject variable here is actually a latent parent (the actual customer satisfaction) for that measurement. In other words, any measured parent to a target is actually not a real parent, but another child to the anticipated parent, and hence a sibling to the target. Thus the same analog goes on. A measured parent's parent to a target is not the real parent's parent, but rather a sibling to the parent. This thought is illustrated in Figure 5.1.



(a) An anticipated causal graph expecting observed P to be a parent to Y and observed PP a parent to P.



(b) In the real graph, P is actually a latent variable while the corresponding observed value is actually S, and PP is actually a latent variable while the corresponding observed value is actually PPC.

Figure 5.1: An illustration of the idea about the real parents is unobserved.

This states that we might be thinking of having the real parents observed, but what we actually have are siblings or aunts. These features can be good predictors still when used within one domain, as we discussed Markov blankets with latent parents. But in changing setting these covariates can no anymore give us invariant predictions.

In a conclusion, the parents to the target are essential for invariant prediction, but once they are measured, it is a hard assumption that these values really present the real parents. And thus, one should be critical if they are really capable to give invariant predictions anymore.

5.4 Limitations

In this work, we have basically encountered theoretical settings with simulated data. In that sense, the suggestions from this work do not correspond to any real-life situations as is. For example, in order to understand the roles of the nodes around the target variable in changing conditions, we have used a very limited set of variables included. In real-life cases, there can be observed plenty of parents and other members of the Markov blanket to the target.

Also within the cases covered, we did not include any additional complexity to the relations between features in the DAGs. For example, having extra arrows from ances-

tors to descendants, or having confounding situations between the parents. However, having such complexity may not necessarily change the conditions for invariant predictions observed in this work, but these occasions might work differently in predictive performance, and hence be relevant in feature selection wise.

5.5 Further topics

In this work, we intentionally studied simple circumstances to get a baseline understanding of the invariant prediction conditions. As the experiments were done with basically linear ground-truth data and modeled out with linear regressions, then the results about invariant prediction are supposed to hold at least in a simple layer. This serves as a starting point to widen the scope to cover cases generally. The next direction could be to investigate the non-linear ground-truth with such a modeling technique that enables catching the non-linear nature. The hypothesis is that invariant prediction conditions found in this study still hold when going beyond the linear, but achieving the invariant prediction might get trickier in a more complex setting, as we already saw with cases with covariate shift in Section 3.2.1.

Another field for further investigation is the target shift. As explained in Section 3.2.2, the bias in case of the target shift can not be fixed with data from only on training domain, but even some hints about earlier changes in the target may help to fix the bias by regressing throughout training domains. Thinking further, about several domains in a timeline, raises the thought that, the close domains are in the timeline, the more likely they must be dependent. Thus the levels of changes can be dependent between the domains, and thus allowing us to predict the change in a new domain by involving, for example, time series modeling with Gaussian processes [42].

The third area of interest is to investigate more closely the phenomena that with latent parent other covariates can not help to achieve invariant prediction conditions. This combined with the problem of having no real parents in feature sets (instead of other closely related covariates) raises questions like, how closely related covariates need to be for the real parent in order to still perform feasibly.

Bibliography

[1] C. F. Aliferis, A. R. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part I: algorithms and empirical evaluation. *J. Mach. Learn. Res.*, 11:171–234, 2010.

[2] C. F. Aliferis, A. R. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part II: analysis and extensions. *J. Mach. Learn. Res.*, 11:235–284, 2010.

[3] O. Anava and K. Y. Levy. k*-nearest neighbors: From global to local. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4916–4924, 2016.

[4] D. Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[5] M. Bartley, E. Hanks, E. Schliep, P. Soranno, and T. Wagner. Identifying and characterizing extrapolation in multivariate response data. *PLoS One*, 14(12), 2019.

[6] M. Basseville and I. Nikiforov. *Detection of Abrupt Change Theory and Application*, volume 15. PTR Prentice-Hall, 1993.

[7] S. F. Crone, J. Guajardo, and R. Weber. A study on the ability of support vector regression and neural networks to forecast basic time series patterns. In *Artificial Intelligence in Theory and Practice*, pages 149–158, Boston, MA, 2006. Springer US.

[8] A. Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

[9] R. B. de Andrade e Silva, R. Scheines, C. Glymour, and P. Spirtes. Learning the structure of linear latent variable models. *J. Mach. Learn. Res.*, 7:191–246, 2006.

[10] F. Eberhardt and R. Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007.

[11] I. Farrance and R. Frenkel. Uncertainty of measurement: A review of the rules for calculating uncertainty components through functional relationships. *The Clinical biochemist. Reviews / Australian Association of Clinical Biochemists*, 33:49–75, 05 2012.

[12] H. Fischer. *A history of the central limit theorem. From classical to modern probability theory*. Springer, 2011.

[13] R. A. Fisher. The arrangement of field experiments. In *Breakthroughs in statistics*, pages 82–91. Springer, 1992.

[14] I. Guyon, C. Aliferis, et al. Causal feature selection. In *Computational methods of feature selection*, pages 63–86. Chapman and Hall/CRC, 2007.

[15] I. Guyon and A. Elisseeff. *An Introduction to Feature Extraction*, pages 1–25. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[16] T. Hengl, M. Nussbaum, M. Wright, G. Heuvelink, and B. Graeler. Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables. *PeerJ*, 6:e5518, 08 2018.

[17] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 601–608. MIT Press, 2006.

[18] G. W. Imbens and D. B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, 2015.

[19] M. A. Javidian, O. Pandey, and P. Jamshidi. Scalable causal transfer learning. *CoRR*, abs/2103.00139, 2021.

[20] Y. Kadwe and V. Suryawanshi. A review on concept drift. *IOSR Journal of Computer Engineering*, 17:20–26, 01 2015.

[21] D. Koller and N. Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

[22] D. Koller and M. Sahami. Toward optimal feature selection. In *Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96), Bari, Italy, July 3-6, 1996*, pages 284–292. Morgan Kaufmann, 1996.

[23] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, 2005.

[24] S. Magliacane, T. van Ommen, T. Claassen, S. Bongers, P. Versteeg, and J. M. Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018*, pages 10869–10879, 2018.

[25] P. Monari. The concept of measure, from plato to modern statistics. art of living or intellectual principle? *Statistica*, 65:243–256, 01 2005.

[26] S. Park, O. Bastani, J. Weimer, and I. Lee. Calibrated prediction with covariate shift via unsupervised domain adaptation. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3219–3229. PMLR, 2020.

[27] J. Pearl. Bayesian networks: A model of self-activated memory for evidential reasoning. Technical Report CSD-850021 R-43, Cognitive Systems Laboratory, Computer Science Department, UCLA, 1985.

[28] J. Pearl. *Probabilistic reasoning in intelligent systems - networks of plausible inference*. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, 1989.

[29] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009.

[30] J. Pearl. *Causal inference in statistics : a primer*. Wiley, Chichester, West Sussex, 2016 - 2016.

[31] J. Pearl and D. Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic Books, Inc., USA, 1st edition, 2018.

[32] J. Pearl and T. Verma. A theory of inferred causation. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91). Cambridge, MA, USA, April 22-25, 1991*, pages 441–452. Morgan Kaufmann, 1991.

[33] J.-p. Pellet and A. Elisseeff. Finding latent causes in causal networks: an efficient approach based on markov blankets. In *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

[34] J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78, 2015.

[35] N. Pfister, P. Bühlmann, and J. Peters. Invariant causal prediction for sequential data. *Journal of the American Statistical Association*, 114(527):1264–1276, 2019.

[36] N. Pfister, E. G. Williams, J. Peters, R. Aebersold, and P. Bühlmann. Stabilizing variable selection and regression. volume 15, pages 1220–1246. Institute of Mathematical Statistics, 2021.

[37] R. Pugliese, S. Regondi, and R. Marini. Machine learning-based approach: global trends, research directions, and regulatory standpoints. *Data Science and Management*, 4:19–29, 2021.

[38] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[39] I. Redko, N. Courty, R. Flamary, and D. Tuia. Optimal transport for multi-source domain adaptation under target shift. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 849–858. PMLR, 16–18 Apr 2019.

[40] I. Redko, A. Habrard, E. Morvant, M. Sebban, and Y. Bennani. *Advances in Domain Adaptation Theory*. Elsevier, 2019.

[41] T. Richardson. Markov properties for acyclic directed mixed graphs. *Scandinavian Journal of Statistics*, 30(1):145–157, 2003.

[42] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110550, 2013.

[43] M. Rojas-Carulla, B. Schölkopf, R. Turner, and J. Peters. Invariant models for causal transfer learning. *The Journal of Machine Learning Research*, 19(1):1309–1342, 2018.

[44] C. Rudin and J. Radin. Why are we using black box models in ai when we do not need to? a lesson from an explainable ai competition. *Harvard Data Science Review*, (2), 2019.

[45] M. Saerens, P. Latinne, and C. Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Comput.*, 14(1):21â41, jan 2002.

[46] N. Silver. *The Signal and the Noise: Why So Many Predictions Fail-but Some Don't*. Penguin Publishing Group, 2012.

[47] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search, Second Edition*. Adaptive computation and machine learning. MIT Press, 2000.

[48] A. J. Storkey. When training and test sets are different: characterising learning transfer. In *In Dataset Shift in Machine Learning*, pages 3–28. MIT Press, 2009.

[49] A. Subbaswamy and S. Saria. Counterfactual normalization: Proactively addressing dataset shift using causal mechanisms. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018, Monterey, California, USA, August 6-10, 2018*, pages 947–957. AUAI Press, 2018.

[50] A. Subbaswamy, P. Schulam, and S. Saria. Preventing failures due to dataset shift: Learning predictive models that transport. In *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 3118–3127. PMLR, 2019.

[51] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, page 255â270, USA, 1990. Elsevier Science Inc.

[52] G. West. *Scale: The Universal Laws of Growth, Innovation, Sustainability, and the Pace of Life in Organisms, Cities, Economies, and Companies*. Penguin Press, 2017.

[53] S. Wright. Correlation and causation. *Journal of agricultural research*, 20(7):557–585, 1921.

[54] K. Yu, L. Liu, and J. Li. A unified view of causal and non-causal feature selection. *ACM Trans. Knowl. Discov. Data*, 15(4):63:1–63:46, 2021.

[55] Y. Zhai, Y. Ong, and I. W. Tsang. The emerging big dimensionality. *IEEE Comput. Intell. Mag.*, 9(3):14–26, 2014.

[56] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain adaptation under target and conditional shift. In *Proceedings of the 30th International Conference on Machine Learning, ICML*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 819–827. JMLR.org, 2013.