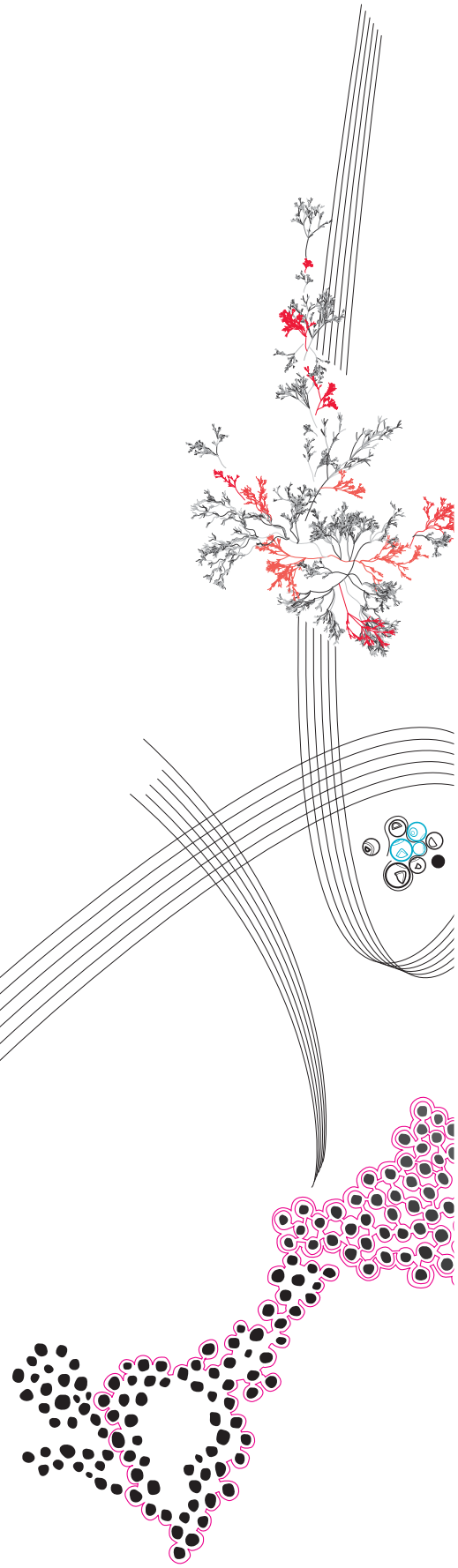


BSc Thesis Applied Mathematics

Domain adaptation under structural causal models

Thomas Kanger

Supervisors: A. Betken, H. Wen

September, 2023

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

1 Abstract

The statistical machine learning problem of domain adaptation (DA) is inspired by the human ability to transfer knowledge from one task to a different but similar task. DA arises when the training and test data are different. By exploiting the relationship between the training and test data, DA methods aim to predict the labels of test data. The performance of DA methods depends on the causal character between the covariates and the labels of training and test data. This paper quantified the expected error of the CIP estimator in the case of a standard linear causal model. Next, some experiments were done to quantify the model's accuracy in labelling test data, while making some adjustments to keep the computation time low.

Contents

2 Introduction

In today's day and age, where applications of Artificial Intelligence and machine learning procedures are increasing one often forgets the lack of transparency and accountability. Since these attributes can be attributed to the inherent "black box" nature of these procedures, steps should be made to help mitigate this "black box" effect. To address this concern and gain a deeper understanding of the operations of models, the utilisation of structural causal models or other methods becomes imperative. This Bachelor's thesis will apply structural causal models in the realm of domain adaptation to effectively mitigate the "black box" effect. The approach is the following
First, the mathematical terminology behind Structural Causal Models and Domain Adaptation will be given. Then, the general model will be introduced. Some theoretical analysis will be done to quantify the accuracy of a DA estimator. Experimental results will be done to get practical insights into the model. Finally, the results will be discussed.

This research covers topics in statistical machine learning. It includes methods for image analysis and works on using structural causal models for domain adaptation problems.

3 Terminology

3.1 Structural Causal Models

Structural Causal Models (SCMs) describe the cause-and-effect relationship between the variables. SCMs are based on variables being causally influenced by other variables. An SCM includes multiple components.

1. Variables. In an SCM, some variables are relevant to the system. Each variable has a range of values or states it can take on.
2. Structural Equations. The structural equations define how each variable in the system is related to the set of variables. The structural equations specify the type of causality.
3. Causal graph. The causal graph is a graph that describes the relationship between variables.

In a domain adaptation setting, SCMs can be used to describe the data generation processes of source and target domains.

3.2 Domain Adaptation

The following definition for transductive transfer learning is given by [2].
Given a source domain D_S and a corresponding learning task T_T, transductive transfer learning aims to improve the learning of the target predictive function $f_T()$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, and $T_S = T_T$. In addition, some unlabelled target-domain data must be available at training time.

For the source domain, we will be considering labelled data pairs $(x_i, y_i)_{i=1}^n$. Here $x_i \in X \in \mathbb{R}^{128\times 84}$ and $y_i \in Y := \{-1, 1\}$. For our source environment, we observe n independent and identically distributed (i.i.d.) samples $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ drawn from

the source data distribution P.
For the target domain we will be considering $\widetilde{n}$ i.i.d. samples
$\widetilde{S} = \{(\widetilde{x}_1, \widetilde{y}_1), (\widetilde{x}_2, \widetilde{y}_2), \ldots, (\widetilde{x}_{\widetilde{n}}, \widetilde{y}_{\widetilde{n}})\}$ from the target distribution $\widetilde{P}$, but we only observe the covariates $\widetilde{S}_X = \{\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_{\widetilde{n}})\}$ from $\widetilde{P}_X = Q$.

The domain and target task are assigning the label $y \in \{-1, 1\}$ to the input x, thereby forming a labelled pair (x, y), so this is a function $f : X \mapsto Y$. The goal is to find a function f that will assign the correct labels to the input. The input of this function will be the covariates. The covariates refer to the input variables shared between the source and target domain. The covariates capture the common characteristics or properties of the data instances from both domains. The function will be parameterised by $\beta \in \Theta$. Now a task is to find β such that the expected error on new, unseen instances from the target population is small. Here Θ is the parameter space, a subset of a finite-dimensional space.

To get theoretical guarantees, we want to quantify the expected error. Given an estimator function f, the performance metric of the target population risk is defined as:

$$\widetilde{R}(f) = \mathbb{E}_{(X,Y)\sim\widetilde{P}}[l(f(X), Y)]$$

Here, l is a loss function. Since different loss functions define different objectives choosing an appropriate loss function is important. The loss function is $x \mapsto x^2$ unless specified otherwise.

4 Models

4.1 Structural Causal Model

To improve the label prediction, SCMs will be considered. An implementation of SCMs into DA tasks can help gain more information about the data generation process.
When the source and target domain contain specific images of donuts, sketches in the source domain and paintings in the target domain, there will be some differences in the distributions for the data generation. There will be a difference due to the colours since sketches use white, black and tints of grey, while paintings generally use different colours. Next to this, the style of the images can be different, while this might be a minor difference. When the data is adjusted for the colours, there will be more similarities between the sketches and paintings. This makes it possible to assume both source and target domain images are generated through similar linear SCMs with additional noise. Using SCMs will be useful for label predictions since SCMs model the causal relationship that influences the data distribution. This can result in higher accuracy predictions than by using general statistical similarities.

To get to data generation equations, the image set has to be adjusted, $X \in \mathbb{R}^{128\times84}$ will become $X_a \in \mathbb{R}^{10752}$ so that we are ready for the column vector (X_a, Y), where X_a stands for the adjusted image set of X and the same holds for $\widetilde{X}_a$. From [1], we find that the data distribution P of our source environment is specified by the following data generation

equations on (X_a, Y) from P,

$$\begin{bmatrix} X_a \\ Y \end{bmatrix} = \begin{bmatrix} \mathbf{B} & b \\ \omega^T & 0 \end{bmatrix} \begin{bmatrix} X_a \\ Y \end{bmatrix} + g(a, \epsilon),$$

the target data distribution $\widetilde{P}$ is specified via the same equation,

$$\begin{bmatrix} \widetilde{X}_a \\ \widetilde{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{B} & b \\ \omega^T & 0 \end{bmatrix} \begin{bmatrix} \widetilde{X}_a \\ \widetilde{Y} \end{bmatrix} + g(\widetilde{a}, \widetilde{\epsilon}).$$

We will have $\mathbf{B} \in \mathbb{R}^{(d \times d)}$, an unknown matrix with real values and a zero diagonal such that the model will learn domain-invariant representations due to the shared factors. Vectors b and $\omega \in \mathbb{R}^d$ are unknown constant vectors; ϵ and $\widetilde{\epsilon}$ are $d+1$-dimensional random vectors drawn from the same noise distribution ε; g is a fixed function to model the change or intervention across source and target environments; $a \in \mathbb{R}^{d+1}$ is an unknown intervention.

4.2 Domain adaptation estimator

Before applying the theory, the mathematical performance of a DA estimator will be investigated. This will allow a better understanding of the results since it will give the expected error. It also results in a theoretical understanding of the estimator and can reveal challenges.

The population conditional invariance penalty (CIP) estimator is a DA method that uses label information in multiple source environments to look for the conditionally invariant components. The conditional mean is matched across multiple source environments. We will refer to it as the CIP-mean. Here we define

$$f_{CIP}(x) := x^T \beta_{CIP} + \beta_{CIP,0}$$

$$\beta_{CIP}, \beta_{CIP,0} := \arg\min_{\beta, \beta_0} \frac{1}{M} \sum_{m=1}^{M} \mathbb{E}_{(X,Y)\sim P^{(m)}} (Y - X^T\beta - \beta_0)^2$$

$$\text{s.t. } \mathbb{E}_{(X,Y)\sim P^{(m)}}[X^T\beta|Y] = \mathbb{E}_{(X,Y)\sim P^{(1)}}[X^T\beta|Y] \quad \text{a.s., } \forall m \in \{2, \ldots, M\}, \tag{1}$$

In equation 1, there is equality between the conditional expectation in the sense of almost sure equality of random variables. This equality can hold when the CIP estimator puts more regression weights on the conditionally invariant components. This means that β depends on the relevance of the attributes. There will be a higher value for the attributes that are invariant across the different source domains. By focusing on these relevant attributes, the CIP-mean emphasises the relevant attributes, specifically those generalisable across domains.

5 Theoretical results

The source and target domain contain specific images of donuts, sketches in the source domain and paintings in the target domain. The labels represent the top view of a donut and the side view of a donut. This means images with a top view generally look a certain

way, and images with a side view look different. Since we know the shape is similar in both sketches and paintings, there will mostly be a causal relationship between the source and target domain.

For a standard linear causal model, the risk of the CIP-mean estimator will be determined. Each data point in the m-th source environment is generated i.i.d. from the following equation

$$X^{(m)} = \mathbf{B}X^{(m)} + a_X^{(m)} + \epsilon_X^{(m)}, \tag{2}$$

$$Y^{(m)} = X^{(m)} + a_Y^{(m)} + \epsilon_Y^{(m)}. \tag{3}$$

This is the equation for a standard linear SCM with noise intervention. The matrix $\mathbf{B} \in \mathbf{R}^{d\times d}$ is an unknown constant matrix with zero diagonal such that $(\mathbf{I}_d - \mathbf{B})$ is invertible, $b = \mathbf{0}_d$ and $\omega = \mathbf{1}_d$.
Using $H = (\mathbf{I}_d - \mathbf{B})^{-1}$, this means the following for (3):
$X^{(m)} = \mathbf{B}X^{(m)} + a_X^{(m)} + \epsilon_X^{(m)}$, $(\mathbf{I}_d - \mathbf{B})X^{(m)} = a_X^{(m)} + \epsilon_X^{(m)}$, $X^{(m)} = H(a_X^{(m)} + \epsilon_X^{(m)})$. A condition for the CIP-mean is that

$$\mathbb{E}_{(X,Y)\sim P^{(m)}}[X^\top\beta \mid Y] = \mathbb{E}_{(X,Y)\sim P^{(1)}}[X^\top\beta \mid Y], \quad \forall m \in \{2,\dots,M\}.$$

Since the expectation of the noise ϵ is zero, the constraint becomes
$\beta^T H a_X^m = \beta^T H a_X^1, \quad \forall m \in \{2,\dots,M\}$. The difference between the observed value of the outcome variable and the prediction of the outcome variable is the residual. It is the following

$$Y^{(m)} - \beta^T X^{(m)} - \beta_0 = a_Y^{(m)} + \epsilon_Y^{(m)} + (1-\beta^T)H(a_X^{(m)} + \epsilon_X^{(m)}) - \beta_0.$$

The target residual becomes

$$\widetilde{Y}^{(m)} - \beta^T \widetilde{X}^{(m)} - \beta_0 = \widetilde{a}_Y^{(m)} + \widetilde{\epsilon}_Y^{(m)} + (1-\beta^T)H(\widetilde{a}_X^{(m)} + \widetilde{\epsilon}_X^{(m)}) - \beta_0.$$

The variance of the label is denoted as σ^2 and with Σ we denote the covariance matrix, then the CIP objective 1 becomes

$$\min_{\beta,\beta_0} \sigma^2 + \frac{1}{M}\sum_{m=1}^{M}(a_Y^{(m)} + (1-\beta^T)Ha_X^{(m)} - \beta_0)^2 + (1-\beta^T)H\Sigma H^T\beta \tag{4}$$

$$\text{s.t. } \beta^T H(a_X^m - a_X^1) = 0, \quad \forall m \in \{2,\dots,M\}. \tag{5}$$

Minimising the equation for β_0 can be done easily, we get

$$\overset{*}{\beta}_0 = \frac{1}{M}\sum_{m=1}^{M}(a_Y^{(m)} + (1-\overset{*}{\beta}^T)Ha_X^{(m)})$$

Using the expression for β_0, 4 becomes

$$\min_{\beta} \sigma^2 + \Delta_Y + (1-\beta^T)H\Sigma H^T\beta \tag{6}$$

$$\text{s.t. } \beta^T H(a_X^m - a_X^1) = 0, \quad \forall m \in \{2,\dots,M\}, \tag{7}$$

where

$$\Delta_Y = \frac{1}{M}\sum_{m=1}^{M}(a_Y^{(m)} - \bar{a}_Y)^2 \text{ and } \bar{a}_Y = \frac{1}{M}\sum_{k=1}^{M} a_Y^{(k)}.$$

Since σ^2 and Δ_Y are both positive and independent of β, we can rewrite 6 as follows

$$\begin{aligned}
&\min_{\beta}(1-\beta^T)H\Sigma H^T\beta \\
&\text{s.t. } \beta^T H(a_X^m - a_X^1) = 0, \quad \forall m \in \{2,\dots,M\},
\end{aligned}$$

Since H is invertible, we can use the substitution $\gamma = H^T\beta$

$$\min_{\gamma\in\mathbf{R}^d}(H-\gamma^T)\Sigma\gamma \tag{8}$$

$$\text{s.t. } \gamma^T(a_X^m - a_X^1) = 0, \quad \forall m \in \{2,\dots,M\}. \tag{9}$$

To solve 8 we need to make sure that $\gamma^T(a_X^m - a_X^1) = 0, \quad \forall m \in \{2,\dots,M\}$. Let $P \in \mathbf{R}^{d\times p}$ be the matrix formed with an orthonormal basis of the p-dimensional subspace span $(a^{(2)} - a^{(1)}, \dots, a^{(m)} - a^{(1)})$. Let $Q_{CIP} \in \mathbf{R}^{d\times(d-p)}$ be the matrix with columns formed by completing the columns of P to a basis of $\mathbf{R}^d$ via Gram-Schmidt orthogonalisation. Then the mapping

$$\begin{aligned}
&\mathbf{R}^{d-p} \mapsto \mathbf{R}^d \\
&\zeta \mapsto Q_{CIP}\zeta
\end{aligned}$$

constitutes a bijection between $\mathbf{R}^{d-p}$ and the set $\{\gamma \in \mathbf{R}^d | P^T\gamma = 0\}$.
We can use the bijection to change the constrained optimisation of 8 to the unconstrained one

$$\min_{\zeta\in\mathbf{R}^{d-p}}(H-\zeta^T Q_{CIP}^T)\Sigma Q_{CIP}\zeta$$

We can solve the problem by setting the gradient to zero.

$$\begin{aligned}
&\frac{\partial}{\partial\zeta}(H-\zeta^T Q_{CIP}^T)\Sigma Q_{CIP}\zeta = H\Sigma Q_{CIP} - 2Q_{CIP}^T\Sigma Q_{CIP}\zeta = 0, \\
&2Q_{CIP}^T\Sigma Q_{CIP}\zeta = H\Sigma Q_{CIP}, \\
&\zeta = 2(Q_{CIP}^T\Sigma Q_{CIP})^{-1}H\Sigma Q_{CIP}.
\end{aligned}$$

We end up with the following CIP estimator after transforming the variables back,

$$\begin{aligned}
&\beta_{CIP} = 2H(Q_{CIP}^T\Sigma Q_{CIP})^{-1}H\Sigma Q_{CIP} \\
&\beta_{CIP,0} = \frac{1}{M}\sum_{m=1}^{M}(a_Y^{(m)} + (1-\beta_{CIP}^T)Ha_X^{(m)})
\end{aligned}$$

The target population risk for (β, β_0) becomes the following

$$\sigma^2 + (\widetilde{a}_Y + (1-\beta^T)H\widetilde{a}_X - \beta_0)^2 + (1-\beta^T)H\Sigma H^T\beta.$$

The expected error is obtained by plugging in the CIP estimator into the equation

$$\widetilde{R}(f_{CIP}) = \sigma^2 + (\widetilde{a}_Y + (1-\beta_{CIP}^T)H\widetilde{a}_X - \beta_{CIP,0})^2 + (1-\beta_{CIP}^T)H\Sigma H^T\beta_{CIP}. \tag{10}$$

This provides us with the theoretical expected error.

6 Experimentation

6.1 Dataset

To start our process of finding a function f, we need a good data set with a source environment, at least one target task and a large enough quantity of i.i.d. samples. The DomainNet dataset [3] contains images sorted into different categories. For computational purposes, there should not be too many images. The labels need to be detectable as well. There was a collection of images of donuts, subdivided into sketches, paintings and other types of images. The images can be manually labelled depending on whether the images represent donuts from the top view or the side.

6.2 Preprocessing of data

Some of the images in our dataset were of bad quality, so deleting some images was necessary. The images differed in pixel dimensions, but this was not a problem since we could preprocess the data by resizing them to become elements of $\mathbb{R}^{128\times 84}$. The source domain will contain sketches of donuts, which I manually labelled as -1 or 1 depending on whether the images represent donuts from above or from the side, respectively. For our target domain, we will be using paintings of donuts, where the target task is to label them as -1 or 1 depending on whether the images represent donuts from above or from the side.

Our next step of preprocessing the data is to make all images grayscaled and to normalise the pixel values between 0 and 1. This step ensures that similar sketches made with a darker pencil are seen as similar sketches by the model. This also finishes the preprocessing of the data.

6.3 Background removal

The data from our source and target domain will still look differently, seeing as the white background of sketches will likely be a coloured background in the paintings.

(A) The sketch of a donut from above

(B) The sketch of a donut from the side

(C) The painting of a donut, unlabelled

FIGURE 1: Two images from the source domain and one from the target domain after being resized.

Above, we can see three images. Our goal is to remove the background of images in both the source and target domain such that the differences in pixel values due to different colouring vanishes. To remove the background, we will use an edge detection algorithm. The local binary pattern (LBP) is a feature that we will use. For all pixels, except the first and last row and column, we will compare the pixel value with the pixel values of their

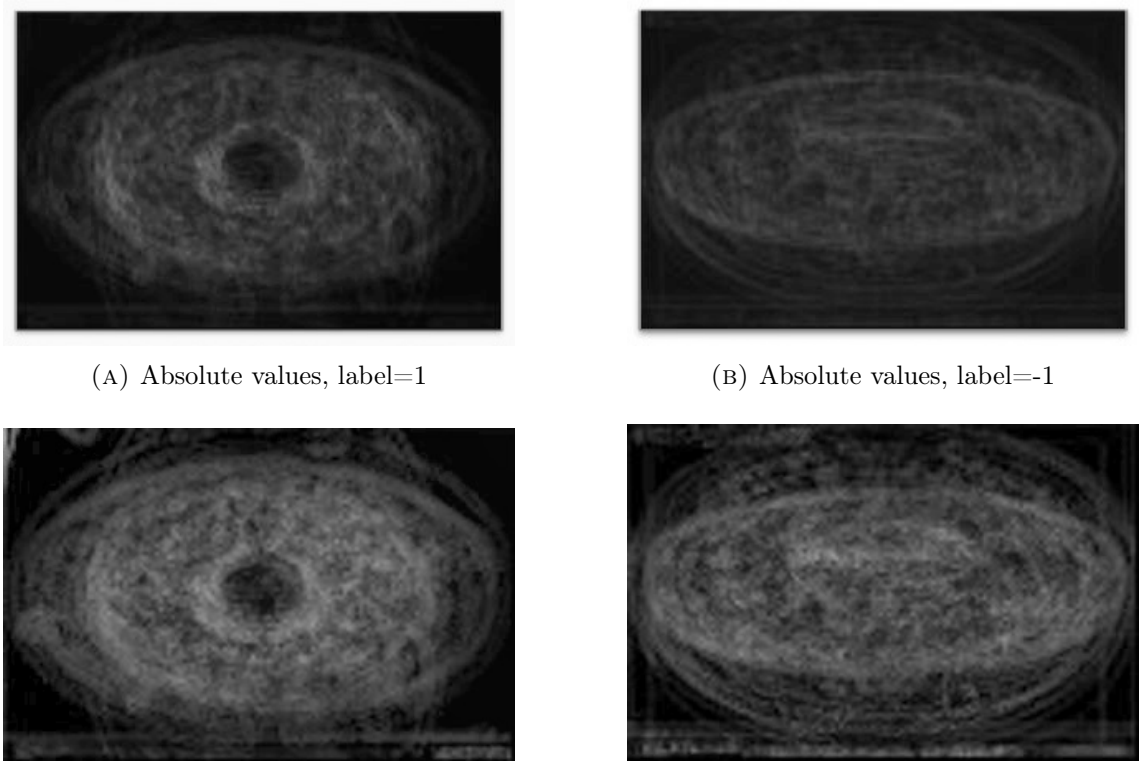
eight neighbours. Below you can find the pseudo-code of the procedure for pixel (i, j) with pixel value $a_{(i,j)}$ and threshold value b:

$values = [0] * 8$
$position = 0$
counterclockwise view at each neighbouring pixel, starting at pixel $(i-1, j-1)$
for pixel (x, y)
if $a_{(x,y)} + b > a_{(i,j)} > a_{(x,y)} - b$
$values[position] = 0$
$else$
$values[position] = 1$
$position+ = 1$
save $values$ and look at the next pixel.

This will result in an $8 * 1$ vector with binary values at each position. We will use these vectors later on. We will call this method the LBP with only big differences.
Another possibility for background removal is to add the absolute difference with the neighbouring pixels. This will also result in $8 * 1$ vectors, but now without a threshold value. We will call this method the LBP with absolute values.

To improve our label prediction, we will incorporate structural causal models. The relevant variables in our model are the pixel values after background removal. To identify the label, we need to find out whether the shape of our target image fits better with the top view or the side view of a donut. In our case, the $8 * 1$ vectors after background removal are the covariates, and the label is either -1 or 1.

Since the causal relationship between the variables in the SCM has been specified, we need to estimate the parameters. Given our total set of labelled training data, there are multiple methods to determine the variables.
We could have an individual relevance factor for each pixel in their $8 * 1$ vector. The number of pixels is huge, so there will be many options, and it would take a large computation time to determine possible individual relevance factors. Also, the size of the pixels is larger than the number of images in our source domain, so there are multiple possible sets of individual relevance factors.
Another possibility is to divide the grid into blocks and determine the relevance factor for each block. By increasing the size of the blocks, we will decrease the computation time. This will also decrease the accuracy.
We can also have one parameter depending on the label. This parameter will be the relative group intensity which is the total intensity for label 1 divided by the total intensity for label -1. To calculate the intensity, we need to do some steps. For each image in our target data of label 1, we will determine the $8 * 1$ vectors for each pixel according to one of the LBP methods. For each pixel, we will take the sum of all values in this vector. For each image, we will take the sum over all pixels and call it the intensity of label 1. To determine the total intensity for label 1, we will divide the intensity of all images in our source domain with label 1 with the total number of images in our source domain with label 1. We will do the same for each image of label -1. Now that we have the total intensity of labels 1 and -1, we can calculate the relative group intensity.
The reason to calculate the relative group intensity is to ensure that the higher average total intensity of a label does not interrupt the labelling process by choosing one label



(A) Absolute values, label=1

(B) Absolute values, label=-1

(C) Binary values, label=1

(D) Binary values, label=-1

FIGURE 2: The average pattern per label given both LBP methods

almost always.

6.4 Labelling the data

The data is preprocessed, and the background of the images is removed. Next to this, the relative group intensity has been calculated. There should be a linear correlation between the images and the labels since there is a linear SCM. Given each image in our target domain, the difference between each pixel value with the average pixel value for each label should be calculated. The prediction will be the label with the lowest total difference. For the LBP methods with only a big difference, the noise is reduced. For the LBP method with absolute values, the noise will be reduced due to the difference in the average value. Now the noise in our target domain will have a minimal effect since the noise will have a similar effect on both labels. For a visual understanding, we can calculate the average pattern in the source domain for each label given an LBP method. This can be calculated by taking the sum of all entries inside the average vector per pixel and creating a new image with these pixel values.
An interesting difference between the LBP methods is the 'noise' at the top and bottom of the images. This 'noise' is only present when applying the LBP method with binary values. This can be explained due to the watermarks in some of the images. The watermarks in the images result in high binary values at the edges of the letters, while the individual intensities might not be that high. When applying the LBP method with absolute differences, these watermarks are not visible, which shows that the averages of these watermarks are not relevant.

6.5 Results

The target domain consists of 160 paintings of donuts that need to be labelled. Below you can find a table with the results.

	Reality	**Predicted accurately**	**Predicted wrongly**	**Accuracy (%)**
Label = 1	88	59	23	67.0
Label = -1	72	46	26	63.9

TABLE 1: Results when applying LBP with absolute difference

	Reality	**Predicted accurately**	**Predicted wrongly**	**Accuracy (%)**
Label = 1	88	79	9	89.7
Label = -1	72	14	58	19.4

TABLE 2: Results when applying LBP with binary values

7 Discussion & Conclusion

In this paper, we used structural causal models (SCMs) to analyse the prediction performance of the CIP estimator in the case of a standard linear causal model. We used the mathematical knowledge of domain adaptation (DA) to quantify the expected error of the CIP estimator. This helped mitigate the "black box" character inherent in Artificial Intelligence applications by gaining extra insight into the model's operations. By quantifying the error, future researchers can use this error to determine whether the CIP estimator would be well-suited for their problem.

A Python program was written to simulate the model, which resulted in some experimental data. A relatively small dataset was used, and new methodologies were tried-out to limit the computational time of the coding program.
The first method employed the linear binary pattern, which incorporates the absolute difference of neighbouring pixels. Using this method attained commendable accuracy.
The second method employed the linear binary pattern, emphasising only significant discrepancies among adjacent pixels. The idea behind the method was to detect the edges of the image to detect the background and remove it. Using the second method improved the accuracy in detecting images of label 1, but this was accompanied by a decrease in the accuracy of label -1. This model predicted most images to be of the label 1 while slightly more images were of label 1. More attention to the threshold value could potentially yield more precise outcomes. However, the small sample size is insufficient to make definitive claims about the functioning of the methods.

References

[1] Yuansi Chen and Peter Bühlmann. "Domain adaptation under structural causal models". In: *Journal of Machine Learning Research* 22 (2021). ISSN: 15337928.

[2] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359. ISSN: 10414347. DOI: 10.1109/TKDE.2009.191.

[3] Xingchao Peng et al. "Moment Matching for Multi-Source Domain Adaptation". In: *Proceedings of the IEEE International Conference on Computer Vision* 2019-October (Dec. 2018), pp. 1406–1415. ISSN: 15505499. DOI: 10.1109/ICCV.2019.00149. URL: https://arxiv.org/abs/1812.01754v4.