Chapter 4
Discriminant Learning Machines

Diviyan Kalainathan, Olivier Goudet, Michèle Sebag, Isabelle Guyon

Abstract The cause-effect pair challenge has, for the first time, formulated the cause-effect problem as a learning problem in which a causation coefficient is trained from data. This can be thought of as a kind of meta learning. This chapter will present an overview of the contributions in this domain and state the advantages and limitations of the method as well as recent theoretical results (learning theory/mother distribution). This chapter will point to code from the winners of the cause-effect pair challenge.

Key words: Cause-effect pairs, causal discovery, discriminant methods, mother distribution

4.1 Introduction

Distinguish causes from effects is of utmost importance in order to understand mechanisms and provide unbiased predictions, or to be able to make recommendations. In order to ascertain causal relationships, randomized controlled experiments

Diviyan Kalainathan
Team TAU - CNRS, INRIA, Université Paris Sud, Université Paris Saclay - France e-mail: Diviyan.kalainathan@lri.fr

Olivier Goudet
Team TAU - CNRS, INRIA, Université Paris Sud, Université Paris Saclay - France e-mail: olivier.goudet@inria.fr

Michèle Sebag
Team TAU - CNRS, INRIA, Université Paris Sud, Université Paris Saclay - France e-mail: sebag@lri.fr

Isabelle Guyon
UPSud/INRIA Université Paris-Saclay, Orsay, France, and ChaLearn, Berkeley, California e-mail: guyon@chalearn.org

represent the gold standard. However those experiments are often costly, unethical, or even unfeasible, leaving only available observational data. Causal discovery out of observational data has been throughoutly studied in the graph setting[1] [Pearl, 2009, Spirtes et al., 2000, Chickering, 2002], but we will focus in this chapter on the particular case where we have only access to two variables without time information to determine their causal relationship. This setting is relevant when only two variables are available, or when only two variables are of interest and are already conditioned on the covariates.

To tackle this problem, the literature proposed generative models for causal discovery which aim to find models matching the empirical distribution of the data (c.f. Chapter IV). These models are sought in a model class, that needs to be restrictive after Zhang and Hyvärinen [2009]: actually, too general a class might allow to learn an accurate generative model whatever the hypothesized causal dependencies, hindering the identification of the true causal mechanisms. Therefore, generative models explicitly assume the simplicity of the sought causal mechanism. For instance, the Additive Noise Model (ANM) [Hoyer et al., 2009] identifies causal relationships when the total of external contributions influence linearly the mechanism:

$$Y = f(X) + E \tag{4.1}$$

where Y represents the effect, X the cause and E a random noise variable accounting for the unobserved variables. The ANM explicitly models the direct effect between the variables through the possibly non-linear causal mechanism f.

Another issue related to simple model classes is the testability of the underlying assumptions[2] which proved itself to be difficult, even though pioneering has been done by Scheines [1997], Zhang and Spirtes [2002], Uhler et al. [2013].

As said, generative models strongly rely on the simplicity assumption, stating that the causal mechanism is the simplest model that generates one variable from the other(s). Here "simplicity" could be formalized in terms of Kolmogorov Complexity (K), stating that the causal direction is the direction holds the lowest K. For instance, Janzing and Schölkopf [2010][3] states that:

$$K(P_{\text{cause}}) + K(P_{\text{effect|cause}}) \leq K(P_{\text{effect}}) + K(P_{\text{cause|effect}}) \tag{4.2}$$

This strong assumption does not always hold true in real-world settings, due to e.g. missing intermediate variables or complex causal mechanisms.

These limitations have been addressed through a new learning approach to pairwise causal discovery, formalized through the Cause Effect Pairs (CEP) Challenge [Guyon, 2013, 2014] (c.f. Appendix 4.A). Considering two variables X and Y and (a sample of) their joint distribution, the CEP goal is to determine the category of their causal relationship (whether X causes Y, or Y causes X, or neither causes the other

[1] where more than two variables are available and conditional independencies can be exploited to recover the causal structure of the graph.

[2] c.f. Section 4.2.2

[3] Refer to Chapter IV for details

one). Thereby the causal discovery problem is shifted from modeling the causal mechanism relating given variables, to a classification problem where any joint distribution is associated a causal class. Accordingly, by leveraging Machine Learning algorithms, a classifier is trained to leverage causally relevant features from joint distributions of pairs of variables sampled from a Mother Distribution (Section 4.2.5) for classification. Such classification approaches come in two modes: i) ensemble learning methods build upon statistical features and pre-existing generative models; ii) discriminant learning methods build on top of representation learning and distribution embeddings.

This chapter first formalizes the pairwise cause-effect inference problem as a classification task (Section 4.2), and thoroughly presents the various approaches for feature construction in Section 4.3. The different approaches and algorithms developed to address these challenges are presented in Section 4.4. The limitations of these approaches are discussed and some perspectives for further research are presented in Section 4.5.

The appendices consist in: Appendix 4.A describes the Cause Effect Pairs Challenges organized by Guyon [2013, 2014], Appendix 4.B refers to the traditional learning bounds [Vapnik, 1998] and Appendix 4.C extends these bounds for our problem of learning out of distributions, with a kernel based feature construction step (Section 4.4.2).

4.2 Problem Setting

This section formalizes pairwise causal discovery as a learning task. You are given a dataset $((d_1,g_1),(d_2,g_2),\ldots,(d_n,g_n))$; each d_j is itself a dataset of pairs $(x_{1j},y_{1j}),...,(x_{pj},y_{pj})$ and the label g_i represents the causal mechanism at play in the dataset d_i, defined after Reichenbach's Principle of Common Cause [Arntzenius, 2010] : i) causal class ($X \to Y$); ii) anti-causal class ($X \leftarrow Y$); iii) there exists a confounding variable Z such that $X \leftarrow Z \to Y$; iv) X and Y are independent ($X \perp\!\!\!\perp Y$). Classes iii) and iv) are merged in the following; we shall return to this point in Section 4.5.

The examples are exploited using mainstream classification algorithms; eventually the trained classifier is used to predict the causal class associated with a new joint distribution $P_{X',Y'}$.

4.2.1 Notations

We will briefly introduce the various notations that we will use throughout this chapter.

- X and Y denote random variables with values in $\mathbb{R}$. Unless specified otherwise, X is considered as the cause and Y as the effect of X.

- P_X represents the probability distribution of X.
- $S_j = \{x_{ij}, y_{ij}\}_{i=1}^{n_j}$ depicts a empirical distribution, based on which the algorithms infer the causal direction of the pair.
- (S_j, g_j) depicts a empirical distribution along with its label g_j, based on which the algorithms learn the causal direction of the pair. g_j represents the ground truth of the causal relation between the two variables (encoded as described in Sec. 4.2.3). This set of data points and label is also called **causal pair**.
- $S = \{S_j, g_j\}_{j=1}^{n}$ denotes the dataset of causal pairs.
- $\mu(P_{S_j})$ represents a single vector of features (of potentially infinite dimension) encoding the embedding of the empirical distribution $\{x_{ij}, y_{ij}\}_{i=1}^{n_j}$.
- $C(X,Y)$ represents the causal coefficient (c.f. Sec. 4.2.4) for the (X,Y) pair of variables.

4.2.2 Causal assumptions

We will define here the various assumptions made in this chapter, some of which traditionally made are not explicitly made by the presented framework. However, some of the assumptions are made by the presented algorithms in Section 4.4.

Reichenbach's principle

states that if two variables X and Y are dependent, then either: i) $X \to Y$, ii)$Y \to X$, iii) $\exists Z, X \leftarrow Z \to Y$, Z being a confounding variable. The presented framework does not make this assumption[4], therefore including the case in which there can be dependency without any causal relationship, e.g. constraint or equilibrium (c.f. Chapter VII).

Causal sufficiency

assumes that the direct dependency between two variables is the result of a direct causal influence between the two variables, and not the result of a confounding effect from a hidden variable (case iii) in Reichenbach's principle). We will not make this assumption in this chapter, as we will consider this case during classification.

Causal faithfulness

states that if two variables X, Y are causally related, then they are dependent. A typical case where this hypothesis does not hold true is if X influences both Y and

[4] even though many algorithms make this assumption

an auxiliary variable Z, and Z influences Y in such a way that the direct effect of X is counteracted by the influence of Z.

Causal Markov

assumes that if two variables X and Y are dependent, then they are d-connected. Under the abovementioned additional assumptions, it comes down to four cases:

1. $X \to Y$
2. $Y \to X$
3. $\exists Z, X \leftarrow Z \to Y$, Z being a confounding variable
4. $X \leftrightarrow Y$, denoting a feedback loop or a 2-cycle.

In this study, we will exclude the case of cycles between the variables of interest; i.e. there exists no paths between X and Y such as $X \to ... \to Y$ and $Y \to ... \to X$, therefore excluding the 4$^{\text{th}}$ case.

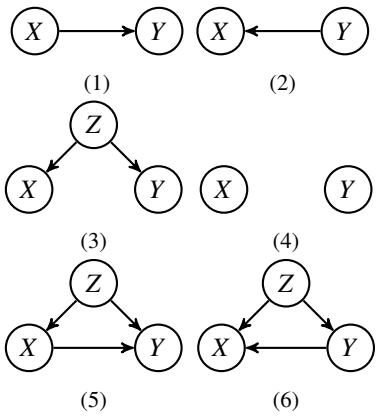
4.2.3 Causal discovery as a classification task

Let $\mathscr{S}$ denote an example associated with a pair of variables X, Y. Its description $\left(\{(x_i, y_i)_{i=1}^m\}, g_i\right)$ is an iid sample drawn after joint distribution $P_{X,Y}$ along with its associated label g_i. g_i is 1 i.f.f. X causes Y ($X \to Y$), -1 if Y causes X ($X \leftarrow Y$) and 0 otherwise (if X and Y are independent, $X \perp\!\!\!\perp Y$; or there exists a third variable Z causing both X and Y, $X \leftarrow Z \to Y$) (Fig. 4.1).
Note that the label g_i primarily depends on the relationship between both variables X and Y: if $X \to Y$ and there exists a third variable Z such that $X \leftarrow Z \to Y$ (Fig. 4.15 and Fig. 4.16), distribution $P_{X,Y}$ is labelled as 1 (c.f. Chapter I).

From a training set made of examples $\mathscr{S}_1, \ldots, \mathscr{S}_n$, mainstream classification algorithms are leveraged to train a classifier, used to associate a causal scenario with any sample coming from a new joint distribution $P_{X',Y'}$.

This problem setting casts causal discovery as a regular supervised learning task. After the usual methodology, the **training set** is used to train a classifier with given hyper-parameters; a **validation set** is used to optimize the hyper-parameters of the learning algorithm; and the performance of the trained classifier is assessed using a **test set**. Notably, this setting accommodates heterogeneous causal discovery problems: examples can involve distribution samples of different sizes, associated with continuous, categorical, ordinal or binary variables.

In order to compensate for this heterogeneity, some pre-processing step (feature construction) can be applied in order to map any joint distribution sample onto a k-dimensional real-valued vector. Appendix 4.B highlights the bounds obtained for a learning problem, for an optimal feature construction step out of distributions.



(7) Table of correspondence between causal scenarios and classes

Configuration	Class
(a)	1
(b)	−1
(c)	0
(d)	0
(e)	1
(f)	−1

Fig. 4.1: A pair of variable X and Y is associated with one out of 6 causal scenarios, falling in 3 classes.

4.2.4 Causation coefficient

While each example falls into one out of 3 causal classes (Fig. 4.1), for convenience one most often associates to each variable pair X,Y a continuous *causation coefficient* $C(X,Y) \in \mathbb{R}$, such that:

- $C(X,Y) > 0$ corresponds to $X \to Y$
- $C(X,Y) < 0$ corresponds to $X \leftarrow Y$
- $C(X,Y) \approx 0$ corresponds to $X \perp\!\!\!\perp Y$ or $\exists Z, X \leftarrow Z \to Y$

The advantage of using a continuous causation coefficient is twofold. On one hand, the absolute value $|C(X,Y)|$ is interpreted as the confidence of the prediction. When $|C(X,Y)|$ goes to 0, the causal direction is unclear; variables could be considered as either independent, or dependent because of a confounding effect; we shall return to this in Section 4.5.

On the other hand, $C(X,Y)$ is used to rank pairs of variables, supporting the definition of confidence based scores such as the precision-recall score or the area under the ROC curve score. From a practitioner's viewpoint, $C(X,Y)$ can be used to prioritize experiments in order to assess causal predictions. Additionally, $C(X,Y)$ allows practitioners to orient edges in partially oriented causal graphs.

4.2.5 Mother Distributions

After Lopez-Paz et al. [2015], the proposed causal discovery setting is amenable to a theoretical analysis rooted in statistical learning theory and risk minimization [Vapnik, 1998]. The analysis relies on the notion of Mother Distribution. Let

$\mathscr{M}$ be a distribution defined on $\mathcal{P} \times \mathcal{G}$, where $\mathcal{P}$ depicts the set of joint distributions of causally related pairs of variables, and $\mathcal{G}$ denotes the set of causal labels (Fig. 4.1). For simplicity, only the case $\mathcal{G} = \{1, -1\}$ is considered in the following; the extension to multi-classes follows from [Lopez-Paz et al., 2015]. All n examples $\{(\mathscr{S}_1, g_1), \ldots (\mathscr{S}_n, g_n)\}$ are independently sampled from $\mathscr{M}$, called the **Mother distribution** of the causal discovery problem.

As said, feature construction[5] is commonly used to map each $\mathscr{S}_j$ onto a k-dimensional real-valued vector in $\mathbb{R}^k$. The problem of learning from empirical joint distributions is thus shifted to a standard supervised learning problem of classification $\mathbb{R}^k \mapsto \mathcal{G}$.

4.2.6 Learning algorithms for this classification problem

A distinctive characteristic of the causal pairwise classification problem compared to the traditional classification problem is the nature of the samples points. In a regular classification problem a sample is a vector representing the position of the example in the feature space $\mathbb{R}^d$, where d represents the number of features. In our pairwise classification problem, a sample is an empirical distribution, a set of points $\{x_i, y_i\}_{i=1}^{n_j}$.

Therefore, a feature construction step is added between the data and the learning algorithm, making the structure of algorithms as shown in Figure 4.2

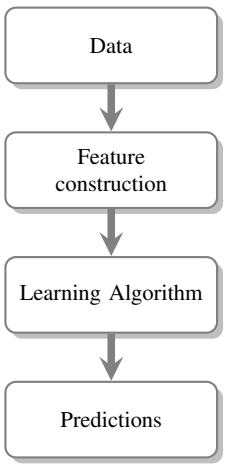


Fig. 4.2: General structure of discriminant learning algorithms for pairwise causal discovery

[5] Representation learning, mapping each distribution sample onto a latent space, will be also considered in Section 4.4.2.

4.3 Feature construction out of distributions for pairwise classification

In order to apply regular learning tools for classification, features have to be extracted out of the data distribution samples. This step is a feature construction step, and the literature has taken three different approaches to extract such features: firstly, the manual construction of causally relevant features to classify the pairs. Secondly, the embedding of the sample distributions into a fixed size feature vector: the resulting manifold will be mapped to the target classes using the training set, allowing to classify unseen examples using the same embedding. Finally, the third approach is to not only use embeddings of distributions, but to also automatically learn and identify classification patterns using the training set.

Sine-based pairwise example dataset In this section, we will illustrate the inner workings of the various features using a simple dataset, as all the mechanisms are sine functions. The causes follow either a Gaussian distribution or a Gaussian mixture distribution; and the noise is additive and is sampled from a Gaussian or a uniform distribution. The causal mechanisms sums up to:

$$Y = \sin(\omega X + \varphi) + E \tag{4.3}$$

with E the noise variable, ω and φ the frequency and phase parameters of the sine function, sampled in $\mathcal{N}_{0,1} \times \mathcal{U}[-\pi, \pi]$. Examples of generated pairs is given in Fig. 4.3. One of the perks of this dataset is the varying complexity of the generated pairs: as ω tends to 0, some of the generated pairs tend to the unidentifiable case of Gaussian input, Gaussian noise and linear mechanism. On the opposite end, when ω takes high values the pairs come down to high frequency sinuses, in which the noise might confuse the pair.

All code of experiments performed in this section is available at: https://github.com/Diviyan-Kalainathan/ChapterV-Causal-Pairs-Book

4.3.1 Handcrafted causal features

An intuitive way to obtain features out of empirical distributions for this new learning problem is to use the output of preexisting causal discovery algorithms, but also feature characterizing the joint and marginal distributions of the samples. In this section we will discuss the different types of features that can be employed as features for the classifier, while giving some examples of those features.

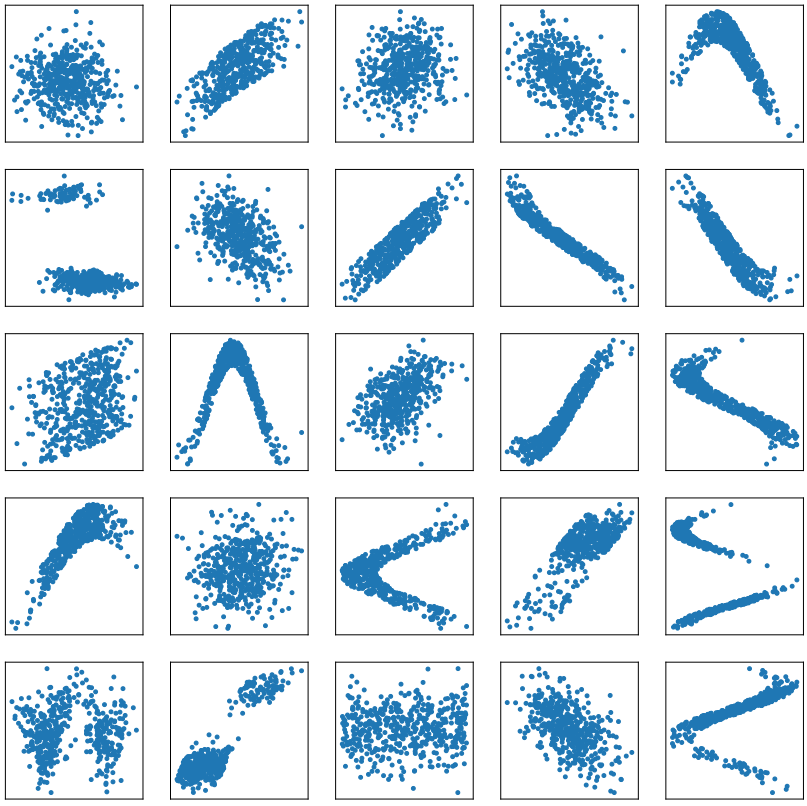


Fig. 4.3: Examples of causal pairs generated with Eq. 4.3

.

4.3.1.1 Statistical features of the distributions

In the Cause-Effect pair challenge (Appendix 4.A), all participants included independence tests in their algorithms: either to avoid testing for causal relationships if the variables are independent[6], or to maximize the accuracy of the predictions as a class in the challenge was dedicated to independent pairs.

The independence test statistics used consist in mainly two types: the correlation-based and the kernel-based tests. Firstly, the correlation-based tests consist in the well-known statistic tests such as the Pearson's correlation and the Spearman's correlation, but also tests based on mutual information. The challenges contained all types of data, including continuous data. In order to compute mutual information out

[6] Therefore assuming Causal Markov

of the empirical distributions, algorithms binned their continuous variables prior to computing mutual information based features. In this section we will consider U, V, obtained by binning X and Y. Examples of these features are mutual information, normalized mutual information [Kvalseth, 1987] and adjusted mutual information [Vinh et al., 2010].

Mutual information is a quantity measuring dependence in information theory, basing itself on how the knowledge of one variable reduces the uncertainty on the other variable. In our case, this quantity can be expressed as:

$$I(U,V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} \frac{|U_i \cap V_j|}{N} \log \frac{N|U_i \cap V_j|}{|U_i||V_j|} \tag{4.4}$$

where U, V represent the input variables and U_i, V_j represent the categories of the variables.

Normalized mutual information [Kvalseth, 1987] is a variation of the mutual information score, normalized to range from 0, representing no mutual information, to 1, representing perfect correlation.

$$NMI(U,V) = \frac{2I(U,V)}{H(U)+H(V)} \tag{4.5}$$

where H represents the entropy. However, this score does not account for chance.

Adjusted mutual information has been proposed by Vinh et al. [2010] to solve this issue, that takes into account the number of samples in each category:

$$AMI(U,V) = \frac{I(U,V) - E(I(U,V))}{\frac{1}{2}(H(U)+H(V)) - E(I(U,V))} \tag{4.6}$$

4.3.1.2 Statistical asymmetries in the distribution

In the cause effect pairs challenge, many statistical quantities have been used to highlight patterns and asymmetries that might provide hints of the causal direction. These features come in different natures: information theory, regression based or statistical properties. The latter denotes features such as moments of the empirical distributions, and moments of regression residuals. These quantities are computed for the learning machine following in the pipeline (Fig. 4.2) to lever these features in order to detect causal patterns in the distributions.

Regression based features represent the majority of features in algorithms using predefined features. They come in various forms, such as the errors of polynomial regressions of various degrees, independence of the residuals with the cause of the polynomial regression. Features of conditional distribution variability have been

introduced by Fonollosa [2016]. One of those, called **standard deviation of the conditional distributions** (CDS) manages to achieve good performance even when used alone. The CDS score measures the spread of the conditional distributions after normalization of the bins:

$$CDS(X,Y) = \sqrt{\frac{1}{M}\sum_{y=0}^{M-1} var_x(p_n(Y=y|X=x))} \tag{4.7}$$

where $p_n(Y=y|X=x)$ represents the normalized conditional probability and var_x the sample variance over x. This feature proved itself very useful for causality detection, as it captures the distribution asymmetry; typically, it standalone yields a score of .69 on the Tübingen dataset [Mooij et al., 2016].

4.3.1.3 Preexisting pairwise causal discovery algorithms

Many approaches basing their inference on predefined features [Fonollosa, 2016, Samothrakis et al., 2013] use as input of the classifier already known models for pairwise causal discovery in a stacked classfier fashion. Examples of used algorithms are the Additive Noise Model (ANM) model [Hoyer et al., 2009] or the Information Geometric Causal Inference (IGCI) model [Daniusis et al., 2012]. These two models are the most employed, as they represent a decent tradeoff between performance and computational cost.

We will briefly present these two algorithms, as a more detailed description is made in Chapter IV.

Additive Noise Model [Hoyer et al., 2009] is one of most popular approaches for pairwise causal discovery. As said, it bases itself of the hypothesis that the causal mechanism is a structural equation model based on a additive noise:

$$Y = f(X) + E \tag{4.8}$$

where f is a (possibly non-linear) function and E is a noise variable independent from the cause X. If the ANM fits in one direction and not in the other, the causal direction is identifiable.

Information Geometric Causal Inference [Daniusis et al., 2012] takes on another approach to infer the causal direction in the pairwise setting. It bases itself on the independence between the cause and the causal mechanism: under the strong assumption that X and Y are related by a bijective relation[7], the cause P_X is independent from the mechanism $P_{Y|X}$ and not in the opposite direction. This approach can also be related to a complexity approach on the mechanisms [Mooij et al., 2016].

[7] therefore assuming minimal noise

4.3.1.4 Applying transformations to variables

Beyond computing all the above-mentioned features on the empirical distributions given as input, Almeida [2018] and Fonollosa [2016] also compute additional sets of the same features, but by changing the input variables by applying transformations. These transformations come in various forms, such as conversions from one type of variable to another, aggregating the distributions, or computing regressions and using the residuals as the new variables.

This kind of transformations allow for computing higher order statistics and to grow the number of features considerably, as such transformations can be stacked multiple times before computing the various features.

4.3.2 Building distribution embeddings

Another approach to feature construction for pairwise causal discovery is to use distribution embeddings to represent the distribution samples in a latent space as a vector with a fixed number of features. Unlike computing a custom set of variables (Section 4.3.1), this approach represents each distribution in a latent space and the learning algorithm learns to split this latent space into the different classes. Inference of unseen pairs consist in applying the embedding to the distribution and reporting the label assigned to the region in the latent space corresponding to the image of the sample. One could see this operation as to look for the closest distribution in the training set to the sample and assign its label. Appendix 4.C develops the bounds given in Appendix 4.B for a kernel-based preprocessing instead of assuming the optimality of the feature construction.

In this section, we will focus on two types of embeddings: kernel-based embeddings of the joint distribution (Section 4.3.2.1) and embeddings of the conditional distributions (Section 4.3.2.2).

4.3.2.1 Kernel based embeddings

Kernel embeddings for learning machines have proven themselves to achieve great performance through strong representational power [Boser et al., 1997, Schölkopf et al., 1997]. To leverage this performance, Lopez-Paz et al. [2015] introduced kernel-based embedding for feature construction in pairwise causal discovery. Starting from the dataset of empirical distributions $S = \{(x_{ij}, y_{ij})_{j=1}^{n_i}\}_{i=1}^{n}$, a kernel mean embedding allows to project all those empirical distributions into the same Reproducing Kernel Hilbert Space (RKHS) $\mathscr{H}_k$. To obtain a homogeneous and low dimension embedding, Lopez-Paz et al. [2015] uses random cosine based embeddings that approximate empirical kernel mean embeddings in low dimension:

$$\mu_{k,m}(P_{S_j}) = \frac{2C_k}{|S|} \sum_{x_{ij},y_{ij} \in S_j} (\cos(w_j^x * x_{ij} + w_j^y * y_{ij} + b_j))_{j=1}^m \in \mathbb{R}^m \tag{4.9}$$

where $\{w_j, b_j\}_{j=1}^m$ are the kernel parameters sampled i.i.d. in $\mathbb{N}_{0,2} \times [0, 2\pi]$, as well as their number m defining the number of dimensions of the output space, P_S is the empirical distribution, and $C_k = \int_{\mathscr{X}} p_k(w)dw$, with $p_k : \mathbb{R}^d \mapsto \mathbb{R}$ the positive and integrable Fourier transform of the chosen kernel k, equal to 1 in this case.

Illustration using the sine dataset (Sec. 4.3) We will now highlight the performance of the kernel mean embeddings using the dataset introduced at the beginning of this section. By applying the embedding and by reducing the dimension of the output feature space using T-SNE [Maaten and Hinton, 2008], we obtain the Figure 4.4. T-SNE is a projection technique that allows for visualization of high dimension spaces, compressing information into local information: close points in the original space are close in the projected space. One can notice on Fig. 4.4 that multiple small homogeneous clusters (from the same class) emerge (such as (1) and (2)), along with a large central heterogeneous cluster ((3)). The small clusters highlight the efficiency of the embedding approach to distinguish classes: Figure 4.5 shows examples of pairs from these distinct clusters, which causal direction is easily identifiable. The pairs composing the same cluster present also the same characteristics of distributions. However, the embedding shows the large cluster (3) is composed by samples from both classes, making these hard to distinguish. Indeed, as shown by scatter plots of some of those pairs in Fig. 4.6, those pairs are hardly identifiable, even though they were labelled and generated by Eq 4.3. These pairs represent distributions sampled from Eq. 4.3, using a small value of ω and a small signal/noise ratio.

4.3.2.2 Embeddings of the conditional distributions

In order to highlight the asymmetries in the distributions, Mitrovic et al. [2018] has introduced embeddings on the conditional distributions $P_{Y|X}$ and $P_{X|Y}$ instead of the joint distribution. This allows for distinguishing asymmetries along with building an embedding of the distribution. In Mitrovic et al. [2018], the proposed conditional embedding is based on the Gaussian kernel along with an α quantity that performs as conditioning:

$$\mu_{k,M}(P_{S_j}) = \left\{ \sum_{j=1}^{n_i} \alpha_j(y) k_m(\cdot, x_j), \sum_{j=1}^{n_i} \alpha_j(x) k_m(\cdot, y_j) \right\}_{m \in M} \tag{4.10}$$

with $\alpha(y) = (\mathbf{L} + n\lambda \mathbf{I})^{-1} \mathbf{l}_y$, $\mathbf{L} = [l(y_i, y_j)]_{i,j=1}^n$, $\mathbf{l}_y = [l(y_1, y), \ldots, l(y_n, y)]^T$, $\alpha(\cdot) = [\alpha_1(\cdot), \ldots, \alpha_n(\cdot)]^T$, regularization parameter λ, identity matrix $\mathbf{I}$, and M the set of parameters for the kernel k.

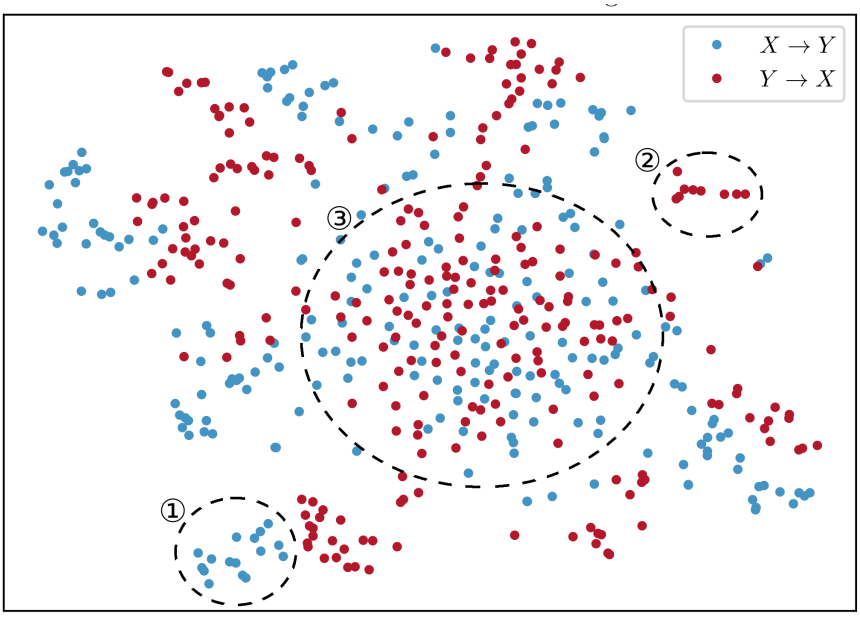
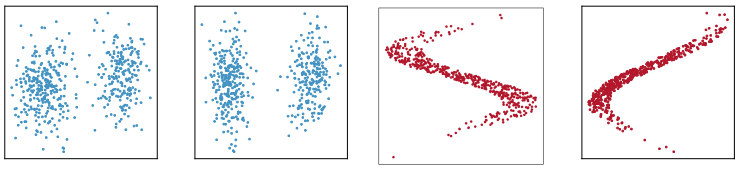


Fig. 4.4: T-SNE of the sine dataset with random mean kernel embeddings. Each point represents a causal pair $\{x_i, y_i\}_{i=1}^{n_j}$ sampled following Eq. 4.3 and a unique set of parameters ω and ϕ. The label of the respective pair is represented by its color.

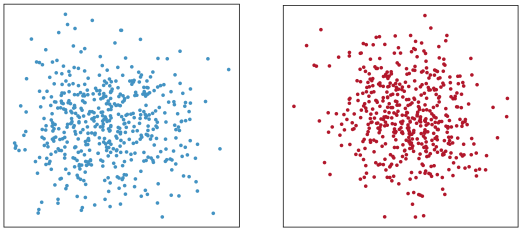


(1) $X \to Y$ pair in ① (2) $X \to Y$ pair in ① (3) $Y \to X$ pair in ② (4) $Y \to X$ pair in ②

Fig. 4.5: Scatter plots of causal pairs in distinct clusters, their number refer to those in Fig. 4.4.

4.3.3 Automatic feature construction out of distributions

Kernel embeddings allow for a general and strong representation of the distributions. However, these representation are not specific to the problem of pairwise causal discovery and therefore some patterns might be missed by those. Therefore, adapting the embeddings to the given distributions and to the task through learning allow the algorithms to automatically distinguish relevant patterns in the distributions, thus merging the last two steps of the four-step procedure described in Fig. 4.2.



(1) $X \to Y$ pair in (3) (2) $Y \to X$ pair in (3)

Fig. 4.6: Scatter plots of causal pairs in the middle cluster in Fig. 4.4. The causal direction is unclear as seen on the scatter plots.

This paradigm fits with "Deep Learning" or more generally into the "Automatic Machine Learning" concept in which only the data has to be fed to the algorithm with no further domain specific knowledge. This merges representation learning and supervised learning: the algorithms learn their own features based on the given data and task.

4.3.3.1 Learning an custom embedding from the samples

Going from empirical distributions to a fixed-size vector representing the learnt relevant features implies a dimension reduction operation. Lopez-Paz et al. [2017] leverages mean embeddings to perform this operation: after applying a transformation to each point in the sample, all outputs are averaged to produce the feature vector representing the sample. This process can be summed up by the following equation:

$$\mu(P_{S_j}) = \frac{1}{n_j} \sum_{i=1}^{n_j} f(x_i, y_j) \tag{4.11}$$

where f is a function with learnable parameters, n_j is the number of points in the sample $S_j = \{x_i, y_i\}_{i=1}^{n_j}$. In Lopez-Paz et al. [2017], the f function is represented by a neural network learnt by backpropagation.

Application on the sine dataset By applying the same methodology as in Section 4.4.2, we train neural network-based mean embeddings using NCC (c.f. Section 4.4.3.1) [Lopez-Paz et al., 2017], and then we plot the embeddings (in 20 dimensions) of the pairs using T-SNE [Maaten and Hinton, 2008]. The results shown in Fig. 4.7, denotes a much clearer separation between the two classes than in Fig. 4.4, therefore highlighting the effectiveness of such automatic feature construction.

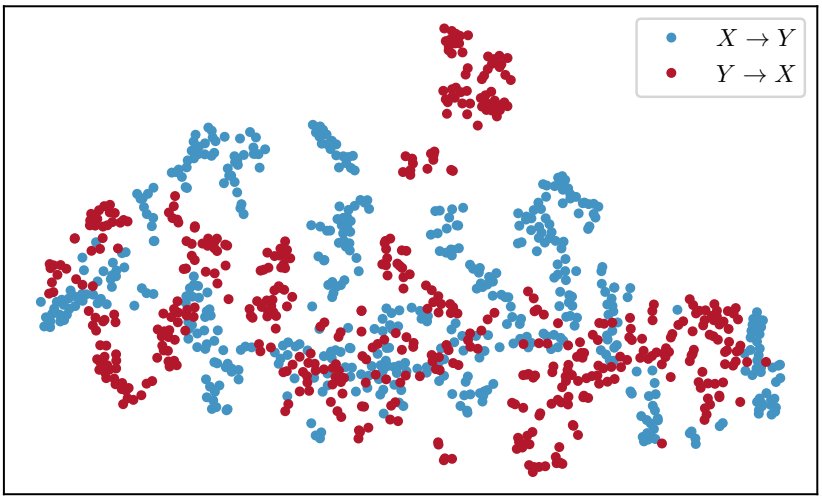


Fig. 4.7: T-SNE of the sine dataset with trained kernel embeddings after 2000 epochs. Each point represents a causal pair $\{x_i, y_i\}_{i=1}^{n_j}$ and the label of the respective pair is represented by its color.

4.3.3.2 Visual patterns on the joint distribution

Another idea to represent the empirical distribution into a fixed-size two dimensional object would be to represent the pair given as input as a scatter plot; the algorithms would then try to visually identify causal patterns in the drawn scatter plot. This approach, exploited by Singh et al. [2017] through a deep convolutional neural network, aligns itself with the examples and the idea that non-invertible causal mechanisms (a visually noticeable feature) give away the causal direction.

Many different visual representations of the distributions are available to the practitioners, and little is known on their influence. We will focus on two of them: the "raw" scatter plot of the data, where a pixel is either 1 or 0 depending on whether a data point is present in the region represented by the pixel. The second is obtained by considering a Gaussian distribution centered on each point, with a relatively low variance. The following equations sum up these two approaches:

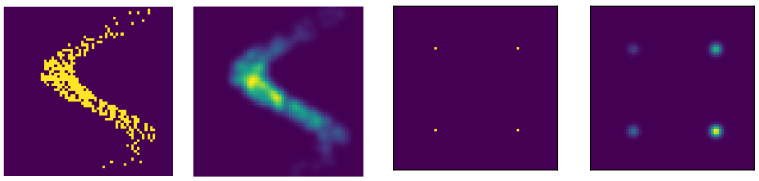
$$\mu_{\text{raw}}[i,j] = \begin{cases} 1 \text{ if } \exists (x,y) \in S, (x*r, y*r) \in [i, i+1] \times [j, j+1] \\ 0 \text{ otherwise} \end{cases} \tag{4.12}$$

where r represents the chosen resolution of the image.

$$\mu_{\text{gaussian}}[i,j] = A \sum_{x,y \in S} e^{-\left(\frac{(x-i)^2}{2\sigma_1^2} + \frac{(y-j)^2}{2\sigma_2^2}\right)} \tag{4.13}$$

where A and $\sigma_1, sigma_2$ represent respectively the amplitude and the standard deviation of the Gaussian distributions.

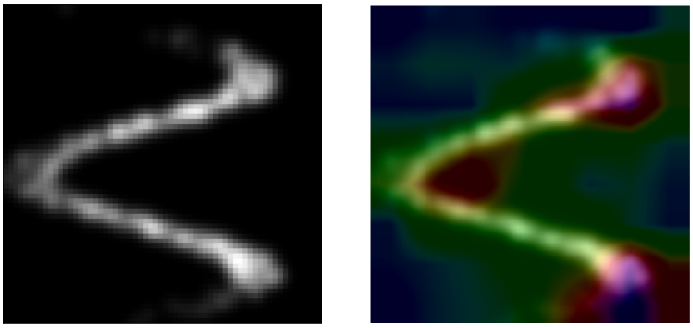
The outputs given by those two approaches is illustrated in Fig. 4.8, Singh et al. [2017] highlighted the influence of some preprocessing methods; they claim that "raw" scatter plot are better for numerical variables as it allows for detection of subtle causal patterns (Fig. 4.81, 4.82), whereas density based scatter plots are more suited to categorical variables, as "raw" scatter plot can sum up to grid-like images (Fig. 4.83, 4.84).



(1) Raw scatter plot (2) Density based plot (3) Raw scatter plot of a binary pair (4) Density based plot of a binary pair

Fig. 4.8: Scatter plots using either Eq. 4.12 for (1, 3) and Eq. 4.13 for (2, 4)

Experiment using gradient visualization We will now perform an experiment consisting in training a convolutional neural network on the above-mentioned sine dataset, and in a second step visualize which pattern triggers the prediction of a causal direction by using Grad-CAM [Selvaraju et al., 2017], a recent visualization technique. The convolutional neural network consists in 3 convolutional layers, taking as input 64×64 pixels images and producing 4096 features fed into 3 layers of dense layers. The network is trained using Adam [Kingma and Ba, 2014], and converges rather quickly using minibatches of 32 images. After applying Grad-CAM, we obtain the Figure 4.92, which highlights that the network uses the non-invertible characteristic of the given causal pair as it looks vertically for the point where a value of X has multiple images in Y, therefore highlighting the non-injectivity of the mechanism function in the $Y = F(X,E)$ hypothesis.



(1) Original scatter plot of a $Y \to X$ pair given as input

(2) Heatmap of the gradient for the $Y \to X$ class

Fig. 4.9: Gradient sensitivity analysis of a $Y \to X$ causal pair using Grad-CAM [Selvaraju et al., 2017], X being on the X-axis and Y on the Y-axis.

4.4 Overview of algorithms using the mother distribution framework

This section reviews the main pairwise causal discovery algorithms participating in the Challenges (Appendix 4.A), distinguishing three categories of pre-processing methodologies: i) manually defined features describing the empirical distributions (Section 4.3.1); ii) features based on the kernelization of the empirical distributions (Section 4.4.2); and iii) latent features based on neural net-based change of representations (Section 4.4.3). As said, standard learning algorithms are used on the top of the pre-processing phase (Section 4.4.1) to learn classifiers and predict the causal label associated with an empirical joint distribution. To avoid redundancy with the previous section, the feature construction step of algorithms will be briefly presented.

4.4.1 Learning algorithms

For the sake of self-containedness, this section briefly presents the long known supervised learning algorithms used in causal discovery algorithms, referring the reader to Bishop [2006] for a more comprehensive introduction. Throughout this section, we will refer to the original software provided by the authors, but many of them are available at `https://github.com/Diviyan-Kalainathan/CausalDiscoveryToolbox`.

4.4.1.1 Decision trees

A decision tree is a tree-like graph model, hierarchically testing conditions on the data features until arriving in a tree leaf, here associated with a causal class (Fig. 4.10). Decision tree learning [Breiman, 1984] iteratively proceeds by determining the most informative feature depending on the current training set. In a classification context, one selects the feature maximizing an information score (e.g. information gain or Gini score; $f^* = \arg\min \sum_v p(f(x)=v) \sum_c p(y=c|f(x)=v) log(p(y=c|f(x)=v))$ and the training data is split according to the value of the selected feature; in a regression context, one selects the feature maximizing the variance of the label conditionally to the feature value ($f^* = \arg\min \sum_v Var(y|f(x)=v)$).

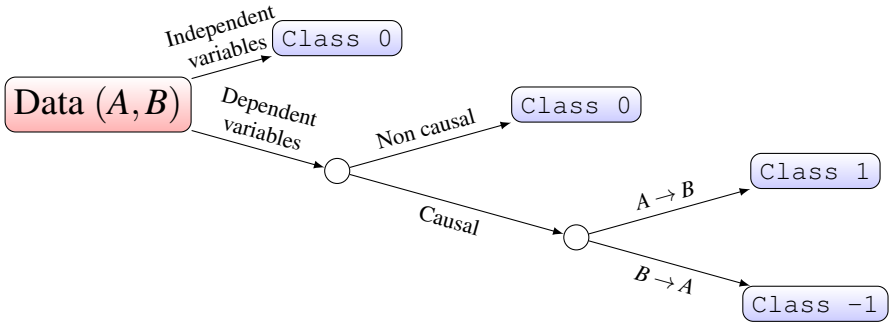


Fig. 4.10: Example of decision tree based on two features: an independence test and a confounder test; the class is the causal label.

4.4.1.2 Random forests

Random forests [Breiman, 2001] address the main limitation of decision trees, namely their potential to overfit small- or medium-sized data through hyper-parameter setting. Given d features, random forests build a large number of decision trees, independently learned using a fraction of the available features (classically, $\sqrt{d}$) and a random subset of the training samples; these trees are aggregated using a vote procedure (bagging). Random forests are celebrated for their excellent empirical performance and computational efficiency. An extension of random forests, Extra-Trees [Geurts et al., 2006] proceed by selecting the splitting condition uniformly (uniformly selecting the feature and the splitting condition in the feature range).

4.4.1.3 Neural networks

Like decision trees, neural nets (NNs) have extensively been used since the 80s for their computational efficiency, versatility and performance. A neural net is a set of interconnected computational units called neurons, delivering an output computed in a non-linear way from its weighted input. NN learning consists of adjusting the weights in order to optimize the learning criterion (e.g. cross-entropy in the classification context; mean-square error in the regression context). In the standard case of feedforward NN (acyclic computational graph), the weights are optimized using stochastic gradient descent as long as all terms involved in the learning criterion are differentiable. The computational efficiency of large neural net learning is related to the use of highly parallel computational architectures such as Graphical Processing Units (GPUs) [Raina et al., 2009]. The stacking of many neuronal layers, yielding deep NNs, supports the building of increasingly abstract features and delivers applicative breakthroughs (see Goodfellow et al. [2016] for an overview).

4.4.1.4 Boosting methods

Boosting [Schapire, 1999] is a term to qualify meta-algorithms that base themselves on many small algorithms, added sequentially as an ensemble method. To each classifier a weight is assigned to measure its relevance for the current classification task. The misclassified training examples are weighted so the following classifiers added to the ensemble improve the performance of the ensemble by focusing on those examples. Well known examples of boosting methods are Adaboost [Freund and Schapire, 1997] and Xgboost [Chen and Guestrin, 2016].

Two methods, using decision trees on the top of manually defined features, with good performance in the Cause-Effect pairs challenges, are **ProtoML** (Section 4.4.1.5) and **Jarfo** (Section 4.4.1.6).

4.4.1.5 ProtoML

Description ProtoML [Almeida, 2018] won the 2013 Cause-Effect pairs challenge on Kaggle. It is based on a pipeline, generating and selecting and generating very many features (up to 20,000+), and achieving supervised learning on the top of the selected features; overall, it aims at minimal human intervention.

Feature construction is based on multiple feature patterns, a feature pattern being a valid set of conversion functions followed by an aggregation function if the feature function outputs a multi-dimensional value. All possible feature patterns are applied to the data

Learning algorithm is a gradient boosted decision tree ensemble [Friedman, 2001] is learned on the top of the extensive set of features thus created, and the learned classifier predicts the causal direction associated with an empirical distribution.

Computational cost is the main limitation of the approach is its learning and prediction computational times (as all features involved in the learned classifier have to be computed for each sample). After the competition, another algorithm named **autocause** based on ProtoML was proposed by the same author, using much fewer features with a huge computational gain at the expense of a slight performance loss.

4.4.1.6 Jarfo

Description Jarfo [Fonollosa, 2016], one of the best performing algorithms over both challenges, operates as follows: i) a type-dependent preprocessing of the input variables is applied; ii) information theoretic measures and other causally relevant features are computed; iii) a gradient boosting classifier. It is rather popular due to the robust performance/computational cost ratio that it offers.

Feature construction The preprocessing of the initial variables goes as follows. Numerical variables are normalized and binned along 19 intervals to compute features such as discrete mutual information or discrete entropy. Categorical variables are relabelled with sorted probabilities to obtain numerical variables. Information-theoretic measures include discrete entropy, mutual information, divergence, and standard deviation on conditional distributions (CDS). Extra features, commonly used in conditional discovery, are computed: Hilbert Schmit Independence Criterion (HSIC), moments, the IGCI score [Janzing et al., 2012] for causal discovery, a Pearson correlation and a polynomial fit on the variables, and the obtained residual of the fit.

Learning algorithm is a gradient boosting classifier based on the previous features is trained using a 10-fold cross-validation.

Computational cost is average, and is dependent on the number of computed features.

4.4.2 Learning over distribution embeddings

The second category of pre-processing uses kernel-based representations of distributions. This randomized functional representation of distributions, exploited using random forest learning, yields the Random Causation Coefficient (RCC) [Lopez-

Paz et al., 2015] with good accuracy and computational efficiency. This idea was extended by Mitrovic et al. [2018] for embeddings of conditional distributions.

4.4.2.1 Randomized Causation Coefficient

Description RCC [Lopez-Paz et al., 2015] introduces kernel embeddings of distributions to pairwise causal discovery, while producing robust performance: standalone precision score above 0.80 on the Tüebingen dataset, and 3rd place on the fast causation challenge [Guyon, 2014].

Feature construction is based on projecting empirical distributions into a RKHS using random mean kernel embeddings, the causal pairs being classified in this new space. (c.f. Sec. 4.4.2)

Learning algorithm is a decision tree learning directly over the kernel space.

Computational cost for this approach is very attractive as the feature construction step is summed up as a projection into a latent space using a random feature matrix.

4.4.2.2 Kernel Conditional Deviance for Causal Inference

Description Kernel Conditional Deviance for Causal Inference (KCDC) [Mitrovic et al., 2018] is an algorithm that extends the approach of Lopez-Paz et al. [2015] regarding embeddings of distributions, by applying it to conditional distributions instead of the joint distributions. It achieves an accuracy score of 78.7% the Tüebingen dataset.

Feature construction As explained more throughoutly in Sec. 4.3.2.2, the embedding is built using the Gaussian kernel along with a conditioning quantity α.

Learning algorithm ranges from a difference between scores of different parameter sets to a random forest algorithms depending on the version of the algorithm used. Another well performing algorithm is a majority vote between the outputs of the scores.

Computational cost is rather low as the algorithm has a simple decision algorithm and the feature construction step is straightforward.

4.4.3 Mapping distributions onto latent spaces

A general trend in Machine Learning, best examplified by Deep Learning [Goodfellow et al. [2016]], aims to seamlessly and autonomously integrate representation learning and supervised learning. This subsection presents two algorithms mapping the empirical joint distributions onto a latent space.

4.4.3.1 Neural Causal Coefficient

Description As said (Eq. 4.21), the random causation coefficient approach proposed by [Lopez-Paz et al., 2015] is based on a predefined kernel matrix capturing the distribution sample. In a further work, Lopez-Paz et al. [2017] learn this feature matrix using a multilayer perceptron. The data sample is sequentially supplied to the NN, and the corresponding outputs are averaged to define a single point, submitted to the classifier NCC:

$$\mathrm{NCC}(\{(x_i,y_i)\}_{i=1}^m) = \psi \left(\frac{1}{m} \sum_{i=1}^m \phi(x_i,y_i) \right), \tag{4.14}$$

where both classifier ψ and representation ϕ are simultaneously trained as neural networks from the sample data.

Feature construction is a neural network, processing each data point before computing the average of all points of a distribution sample (c.f. Section 4.3.3.1).

Learning algorithm is also a neural network, taking as input the mean embedding of the pairs.

Computational cost On the one hand, the NCC approach linearly scales with the size of the training set, using stochastic gradient descent to train both classifier ψ and representation ϕ. On the other hand, neural training is known to require large sized datasets. Empirically, NCC standalone achieves a score of 0.79 precision on the Tüebingen dataset [Mooij et al., 2016], matching the score of RCC at a fraction of its cost.

4.4.3.2 Convolutional neural networks

Description Finally, another possibility is to view the empirical distribution as an image, and to exploit convolutional neural architectures extensively used in computer vision to learn from such images. [Singh et al., 2017] exploits scatter plots

built from the empirical distributions and uses these to train a convolutional neural network architecture (CNN).

Feature construction For numerical variables, the scatter plot is used after a standard normalization. For categorical or binary variables, scatter plots usually are uninformative (Fig. 4.11), and this issue is addressed by coloring the points with the normalized frequency of the observations.

(a) Binary-Binary: -1 (b) Binary-Binary: -1 (c) Num-Num: 1 (d) Num-Num: 1

Fig. 4.11: Scatter plots of binary-binary (a, b) and numerical-numerical (c, d) empirical distributions. Raw scatter plots (i.e. data points) are represented in a) and c); colored scatter plots in b) and d) associate to each point its frequency [Singh et al., 2017].

Learning algorithm The CNN is used together with a gradient boosted classifier inspired from **Jarfo** [Fonollosa, 2016], delivering a score of .825 on the Cause-effect pairs challenge. This score, obtained after the end of the challenge, outperforms that of the challenge winner **ProtoML** [Almeida, 2018].

Computational cost The computational cost is rather low as it leverages the computational efficiency of convolutional neural networks. However the proposed solution as a ensemble method with Jarfo (Sec. 4.4.1.6) makes it quite computationally heavy.

4.5 Discussion

Within the Mother Distribution framework (Section 4.2), the pairwise causal discovery problem is cast as a supervised learning problem. The advances made along this formalization, leveraging machine learning algorithms and due to the efforts of all participants to both causality challenges, are impressive (Table 4.1). This section analyses their current limitations and discusses some research perspectives to address them.

4.5.1 Sensitivity to Mother Distributions and Generalization

A primary limitation is due to the examples used to train the classifiers. As widely known, the accuracy of trained classifiers depends on the quality of the training examples. In quite a few causal examples however, the joint distributions present typical features giving away the causality label, a phenomenon referred to as *data leakage*. Another issue would be the presence of biases in the training set of the classifiers. For example, if the causal pairs with one categorical variable and one numerical variable are always labelled such as categorical $\rightarrow$ numerical, the learning algorithms might learn biased features on distributions due to the training set.

For instance, if we take an example with two numerical quantities but one is sampled regularly; e.g. a physical experiment evaluating the influence of voltage on the perceived luminance of a light bulb typically proceeds by setting the voltage value using a regular grid. The acquisition process thus introduces a substantial bias in the data through the marginal distribution of the cause (Fig. 4.12), with a number of unique values much lower for the cause than for the effect variable.

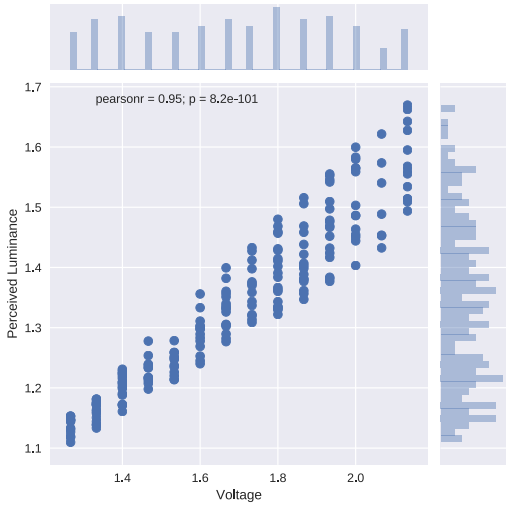


Fig. 4.12: Joint and marginal plots of the voltage/luminance of the light bulb example.

Such biases hinder the generality of the causality classifiers, as they might be exploited by learning algorithms and induce biased hypotheses.

A second limitation related with the data is their insufficient amount. As far as neural nets and deep learning are involved in the learning process, the quantity of examples also becomes essential. Given the comparatively few variable pairs for which the causality label is known from prior knowledge, many authors thus rely on

data augmentation, generating new artificial examples from scratch or by perturbing the available examples [Lopez-Paz et al., 2015].

However, theoretical results require that causal classifiers be trained and evaluated on examples following the same Mother Distribution. The empirical results (Table 7.2) also confirm that the classifier accuracy is much better when applied on data following the same Mother Distribution as the training examples. As in all machine learning problem, the simplest setting is the i.i.d. setting in which training and test data are drawn from the same distribution. The same applies to the cause-effect pair problem: higher performance is attained when the pairs are drawn from the same mother distribution. Unfortunately, in many real world applications, one does not know from which ""mother distribution" a new incoming pair to be classified is drawn and one does not have labeled examples of cause-effect pairs from the "mother distribution" of interest.

Both limitations, regarding the quality and the quantity of the training data, can be addressed using Domain Adaptation and Transfer Learning principles [Ben-David et al., 2010, Ganin et al., 2016], adapting classifiers trained from abundant artificial and diversified Mother Distributions to focused application domains.

4.5.2 Refining the supervised learning problem

Variable pairs (X,Y) actually fall in one out of four cases: the causal case $X \to Y$, the anti-causal case $X \leftarrow Y$, the independent case $X \perp\!\!\!\perp Y$, and the confounder case $X \leftarrow Z \to Y$. For convenience, the four cases are handled using a single continuous causation coefficient $L(X,Y)$, positive in the causal case and negative in the anti-causal case, and both the independence and the confounder case are associated with a low absolute value of $L(X,Y)$. In all generality however, a low value of $L(X,Y)$ might reflect either the independence of both variables, or the uncertainty regarding the causal direction.

A perspective for further research thus consists in extending the proposed framework, and associate with each variable pair (X,Y) two continuous scores, noted $(\lambda_{X,Y},\lambda_{Y,X}) \in \mathbb{R}^2_+$, respectively characterizing the causal and anti-causal strength of the link between both variables. This pair of scores lends itself to a clear interpretation (Figure 4.13), enabling to distinguish the independence region where both scores are low, from the region of 2-cycles where both scores are high, from the confounding case where both scores are neither low nor high but similar, from the causal and anti-causal region.

4.5.3 Explaining the causal mechanism

Another perspective for further research concerns the explanation of the causal mechanism. Quite a few causal algorithms proceed by identifying the potential

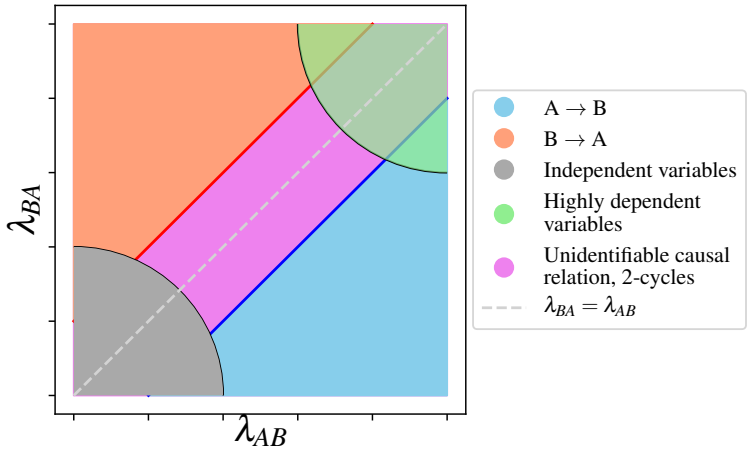


Fig. 4.13: Representation of a 2 dimension causal coefficient $(\lambda_{AB}, \lambda_{BA})$ and the associated causal interpretations depending on the values of each dimension. Taken from Kalainathan et al. [2018].

causal mechanism leading from X to Y and vice-versa, and selecting the causal label depending on the causal mechanism best fitting the data (subject to some limitations on its complexity, as noted in the introduction). It is of utmost interest to the practitioner to "open the black-box" and understand the nature of the underlying causal mechanism, typically distinguishing four cases depending on the influence of the noise variable E:

- $Y = f(X) + E$ (Post-additive)
- $Y = f(X) \times E$ (Post-multiplicative)
- $Y = f(X + E)$ (Pre-additive)
- $Y = f(X \times E)$ (Pre-multiplicative)

A potential approach would be to extend and apply the Automated Statistician [Lloyd et al., 2014, Kim and Teh, 2018] to uncover the nature of the causal mechanism, making a leap towards explainable causal learning.

4.6 Conclusion

Pairwise causal discovery has shown itself as a slightly particular machine learning problem: in fact, the samples are not represented by single vectors of features, but by empirical distributions which number of samples is not fixed. However, literature

has quickly adapted itself through the cause-effect pair challenges [Guyon, 2013, 2014] by adding a feature construction step before its traditional learning algorithm. Algorithms have taken different paths to build their features: off the shelf features of distributions, embeddings of distributions, or even learning those embeddings. Finally, discriminative learning machines have proven themselves to be quite useful for pairwise causal discovery as their accuracy exceeded 80 percent.

Acknowledgments

The authors want to thank David Lopez-Paz for many discussions and insights, and Corentin Tallec for his insightful feedback. The second author was funded on a grant from *La Fabrique de l'Industrie*.

References

Diogo Montinho Almeida. Pattern-based causal feature extraction. *Cause-effect Pairs*, (Chapter 10), 2018.

Frank Arntzenius. Reichenbach's common cause principle. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2010 edition, 2010.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, 2010.

Chris Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

Bernard Boser, Isabelle Guyon, and Vladimir Vapnik. Pattern recognition system using support vectors, July 15 1997. US Patent 5,649,068.

Stéphane Boucheron, Olivier Bousquet, and Gábor Lugosi. Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics*, 9:323–375, 2005.

Leo Breiman. Classification and regression trees. 1984.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

David Maxwell Chickering. Optimal structure identification with greedy search. *JMLR*, 2002.

Povilas Daniusis, Dominik Janzing, Joris Mooij, Jakob Zscheischler, Bastian Steudel, Kun Zhang, and Bernhard Schölkopf. Inferring deterministic causal relations. *arXiv*, 2012.

José Fonollosa. Conditional distribution variability measures for causality detection. *arXiv*, 2016.

Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Isabelle Guyon. Chalearn cause effect pairs challenge, 2013. URL http://www.causality.inf.ethz.ch/cause-effect.php.

Isabelle Guyon. Chalearn fast causation coefficient challenge. *Codalab platform, ChaLearn*, 2014.

Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. *NIPS*, 2009.

Dominik Janzing and Bernhard Schölkopf. Causal inference using the algorithmic markov condition. *IEEE Transactions on Information Theory*, 56(10):5168–5194, 2010.

Dominik Janzing, Joris Mooij, Kun Zhang, Jan Lemeire, Jakob Zscheischler, Povilas Daniušis, Bastian Steudel, and Bernhard Schölkopf. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 182:1–31, 2012.

D. Kalainathan, O. Goudet, I. Guyon, D. Lopez-Paz, and M. Sebag. SAM: Structural Agnostic Model, Causal Discovery and Penalized Adversarial Learning. *ArXiv e-prints*, March 2018.

Hyunjik Kim and Yee Whye Teh. Scaling up the Automatic Statistician: Scalable structure discovery using Gaussian processes. In *AISTATS*, volume 84 of *Proceedings of Machine Learning Research*, pages 575–584, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Tarald O Kvalseth. Entropy and correlation: Some comments. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(3):517–519, 1987.

James Robert Lloyd, David Duvenaud, Roger Grosse, Joshua B. Tenenbaum, and Zoubin Ghahramani. Automatic construction and natural-language description of nonparametric regression models. In *AAAI*, 2014.

David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Ilya O Tolstikhin. Towards a learning theory of cause-effect inference. *ICML*, 2015.

David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Schölkopf, and Léon Bottou. Discovering causal signals in images. *CVPR*, 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Jovana Mitrovic, Dino Sejdinovic, and Yee Whye Teh. Causal inference via kernel deviance measures. *arXiv preprint arXiv:1804.04622*, 2018.

Joris M Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *JMLR*, 2016.

Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, Bernhard Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

Judea Pearl. *Causality*. 2009.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pages 1313–1320, 2009.

Rajat Raina, Anand Madhavan, and Andrew Y Ng. Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th annual international conference on machine learning*, pages 873–880. ACM, 2009.

Walter Rudin. *Fourier Analysis on Groups*. Wiley, 1962.

Spyridon Samothrakis, Diego Perez, and Simon Lucas. Training gradient boosting machines using curve-fitting and information-theoretic features for causal direction detection. 2013.

Thomas Schaffter, Daniel Marbach, and Dario Floreano. Genenetweaver: In silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16): 2263–2270, 2011. wingx.

Robert E Schapire. A brief introduction to boosting. 1999.
Richard Scheines. An introduction to causal inference. 1997.
Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997.
Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. 2017.
Karamjit Singh, Garima Gupta, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Deep convolutional neural networks for pairwise causality. *arXiv preprint arXiv:1701.00597*, 2017.
Peter Spirtes, Clark N Glymour, Richard Scheines, David Heckerman, Christopher Meek, Gregory Cooper, and Thomas Richardson. *Causation, prediction, and search*. MIT press, 2000.
Ingo Steinwart and Andreas Christmann. Support vector machines. *Information Science and Statistics*, 1, 2008.
Caroline Uhler, Garvesh Raskutti, Peter Bühlmann, and Bin Yu. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, pages 436–463, 2013.
Vladimir N Vapnik. *Statistical learning theory*. Adaptive and learning systems for signal processing, communications and control series. John Wiley & Sons, New York. A Wiley-Interscience Publication, 1998.
Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.
Jiji Zhang and Peter Spirtes. Strong faithfulness and uniform consistency in causal inference. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 632–639. Morgan Kaufmann Publishers Inc., 2002.
Kun Zhang and Aapo Hyvärinen. On the identifiability of the post-nonlinear causal model. *UAI*, 2009.

Appendices

4.A The Cause-Effect Pair Challenges

Two challenges pioneering the above causal setting were organized by ChaLearn [Guyon, 2013, 2014]. This section reports on the data and the experimental setting of both challenges, together with their results.

4.A.1 Kaggle Cause-Effect Pair Challenge

The Cause-Effect Pair Challenge organized in 2013 on the Kaggle platform by Guyon [2013] is the first competition focusing on pairwise causal discovery, pioneering the supervised learning setting presented in Section 4.2. The training data involves 12,081 pairs of variables; the test data involves 4,050 other pairs of variables. Each pair of variables is associated its ground truth causal label, ranging in four classes respectively corresponding to $X \to Y$, $X \leftarrow Y$, $X \perp\!\!\!\perp Y$ and $\exists Z, X \leftarrow Z \to Y$.

The training and test pairs of variables included circa 18% real pairs and 82% artificial pairs with continuous, categorical and binary variables.

Real Data originate from multiple domains: demographics, medicine, ecology, genetics, economics and engineering. The causal labels are determined by considering exogenous variables as causes or independent variables, and using prior knowledge to assess the plausibility of the causal relationship. For example, the causal label of pair $(Age, Wage)$ is $Age \rightarrow Wage$ as i) interventions on the $Wage$ variable do not affect the Age variable; ii) Age increase does increase the $Wage$ due to the seniority bonus. The circa 4,000 real pairs included in the challenge are equidistributed among the 3 causal relationship classes. Independent pairs are built by randomly shuffling one of the variables, thus breaking the causal relationship. The generation of pairs falling in class 4 (involving a confounding variable) proceeds by i) considering a real pair X,Y; ii) generating three artificial variables $Z,\tilde{X},\tilde{Y}$ such that $\tilde{X} \leftarrow Z \rightarrow \tilde{Y}$; iii) replacing $\tilde{X}$ values by X values using a monotonous transformation, and likewise replacing $\tilde{Y}$ values by Y values. Care was taken to make sure that the causal relationship between X and Y could not be determined solely on the basis of simple statistics of the marginal distributions of X and Y: all variables were standardized and quantized on a number of levels distributed similarly for X and Y in all four causal relationship classes.

Artificial Data are generated by perturbing real-world data as follows. The cause variable X is selected among the real variables, and the effect variable Y is generated using four causal equations involving a fixed causal mechanism f and an additive or multiplicative noise E:

1. $Y = f(X) + E$
2. $Y = f(X) \times E$
3. $Y = f(X + E)$
4. $Y = f(X \times E)$

Performance metric. The hypothesis learned by either a classification or a regression algorithm associates with each variable pair X,Y an estimated causation coefficient $\widehat{C(X,Y)}$ in $\mathbb{R}$; a positive $\widehat{C(X,Y)}$ is interpreted as X causing Y while a negative $\widehat{C(X,Y)}$ is interpreted as Y causing X. Two criteria are considered: L_1 denotes the Area Under the ROC Curve (AUC) associated with the prediction of $X \rightarrow Y$ against the other three classes, and L_2 denotes the AUC associated with the prediction of $X \leftarrow Y$ against the other three classes. The score of the algorithm is the half sum of both AUCs. While this score does not directly account for class 0 (independent variables or dependent variables due to a confounding variable), this class is implicitly taken into account through the pair ordering based on $\widehat{C(X,Y)}$, as wrongly classified independent pairs penalize one of the AUC scores.

4.A.2 Codalab Fast Causation Coefficient Challenge

Most approaches submitted to the Cause-Effect Pair Challenge involve a heavy feature construction process, associating to each sample of any joint distribution $P(X,Y)$ a real-valued vector of feature values (up to 20,000 features), on the top of which a standard learning algorithm is used. Due to the high computational effort required to achieve this statistical feature construction, a follow-up two-month challenge, the Fast Causation Coefficient challenge has been proposed by Guyon [2014], aimed at algorithms achieving a reasonable trade-off between predictive causal accuracy and computational efficiency. The assessment of algorithms was made possible as the Fast Causation Coefficient challenge (with same setting as the previous challenge) was hosted on the Codalab challenge platform. This Codalab platform allows participants to submit an executable code, that can therefore be assessed in a fair and reproducible way.

4.A.3 Results of the challenges

The Cause-Effect Pair Challenge spanned over 5 months in 2013. 266 teams participated to the competition and submitted over 4,578 entries. As said, most submissions relied on a two step procedure: i) data pre-processing and computation of predefined statistical features describing the empirical distributions; ii) learning of a classifier on top of these features. The pre-processing and the feature extraction were diversified, ranging from normalization, binning numerical variables and grouping categorical variables, to independence tests, entropy measures, and computing fit residuals. On the contrary, the classifiers used were mainly based on decision trees or random forests (85%).

(a) Results of the Cause-Effect Pairs challenge

Algorithm	Author	Ladder Score
ProtoML[1]	Diogo Moitinho de Almeida	0.820
Jarfo[2]	José A.R. Fonollosa	0.811
FirfID[3]	Spyridon Samothrakis, et Al.	0.800

(b) Results of the Fast Causation Coefficient challenge

Algorithm	Author	Ladder Score	Execution Time
Jarfo[2]	José A.R. Fonollosa	0.826	1891 s
FastCausation[4]	Wei Zhang	0.818	1057 s
RCC[5]	David Lopez-Paz	0.719	316 s

Table 4.1: The two Causality challenges: winning algorithms.

The results obtained[8] were quite promising (Table 4.1) with a final score of .82 on the test set (where the score is the half sum of the AUCs associated with the $X \rightarrow Y$ and the $X \leftarrow Y$ classes).

The best performing algorithms were further tested using an additional 3,648 new cause-effect pair benchmark generated by the organizers using the GeneNetWeaver 3.0 software [Schaffter et al., 2011] based on the E.Coli transcriptional regulatory network. Two experiments were performed: the first one was to apply the given algorithms with no training on the new dataset and the second experiment was to train the algorithms on one half of the dataset and to test on the other half. The experiments were conclusive (Table 4.2): the AUC score for the first experiment was of 0.80 for ProtoML and of 0.87 for Jarfo, and over 0.99 for both algorithms on the second experiment. These experiments empirically establish the merit of the winning algorithms.

Table 4.2: Post-Challenge (Cause-Effect Pairs) experiment based on a new 3,648 pairs dataset generated with GeneNetWeaver [Schaffter et al., 2011]

	Algorithm	Experiment 1	Experiment 2
AUC	Jarfo	**0.873**	**0.997**
	FirfiD	0.596	0.984
	ProtoML	0.8085	0.991
Time	Jarfo	**5 hrs**	**5 hrs**
	FirfiD	7 hrs	8 hrs
	ProtoML	10 hrs	12 hrs

The goal of the follow-up Fast Causation Coefficient challenge is to reduce the computational cost of causal discovery with no performance loss compared to the first challenge. This challenge attracted 7 participants, who were given a light version of the **Jarfo** algorithm[9], achieving the best performance *vs* computational cost tradeoff on the first challenge. As shown in Table 4.1b, the original version of **Jarfo** by Fonollosa [2016] came on top of the ranking. The second algorithm, **FastCausation**, managed to almost preserve **Jarfo** predictive accuracy while reducing the computational cost by 44%. The third **RCC** algorithm used a distribution embedding instead of manual feature extraction, and achieved a score of 0.72 for 17%

[1] https://github.com/diogo149/CauseEffectPairsChallenge
[2] https://github.com/jarfo/cause-effect
[3] https://github.com/ssamot/causality
[4] https://github.com/waynezhanghk/FastCausation
[5] https://bitbucket.org/lopezpaz/causality_challenge

[8] The monetary rewards ranged from 1500 USD (with 1000 USD for travel expenses) to 500 USD for the best performing teams that made their software publicly available.

[9] version that does not include some of the most computationally expensive features

of the computational cost of **Jarfo**. This follow-up challenge thus deliver practical algorithms, with decent predictive accuracy at an affordable computational cost.

4.B Error bounds for a classical classification problem

In this section, we will remind the error bounds for a traditional learning problem, where the goal is to classify samples $\{x_i\}_{i=1}^n$ in the label space $\mathcal{G}$, where x_i is a k-dimensional feature vector. Given a loss function $\mathscr{L}$, the learning goal thus is to find a classifier $h : \mathbb{R}^k \to \mathcal{G}$ with minimal expected risk $R(h)$ [Vapnik, 1998]:

$$R(h) = \mathbb{E}_{x,g\sim\mathbb{R}^k\times L}[\mathscr{L}(h(x),g)] \tag{4.15}$$

with $g \in \mathcal{G}$ The expected risk is classically related to the empirical risk $\hat{R}(h)$ measured on the available training set:

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\mathscr{L}(h(x_i),g_i) \tag{4.16}$$

The standard loss function is the 0-1 loss $\mathscr{L}_{01}(\hat{g},g) = |\hat{g}-g|$, for which $R(h)$ comes down to the probability of misclassification. While the consistency of the 0-1 loss is established [Boucheron et al., 2005], it defines a non-convex optimization problem. For tractability, real-valued classifiers $f : \mathbb{R}^k \mapsto \mathbb{R}$ are considered [Steinwart and Christmann, 2008], and margin-based loss functions $varphi : \mathbb{R} \to \mathbb{R}_+$, with $\varphi(f(x),g) = [m - f(x)g]_+$ (where $[A]_+ = max(A,0)$) are used [Bartlett et al., 2006], inducing a smooth optimization problem. The associated expected and empirical risks respectively read:

$$R_\varphi(f) = \mathbb{E}_{x,g\sim\mathbb{R}^k\times\mathscr{L}}[\varphi(f(x),g)] \tag{4.17}$$

$$\hat{R}_\varphi(f) = \frac{1}{n}\sum_{i=1}^{n}\varphi(f(x_i),g_i) \tag{4.18}$$

Letting f^* (respectively $\hat{f}_n$) denote the hypothesis minimizing the expected risk, Eq 4.17 (resp. the empirical risk, Eq. 4.18), the excess risk $\mathscr{E}_\mathscr{F}(\hat{f}_n)$ is bounded after Boucheron et al. [2005]:

Theorem 4.1. *Let $\mathcal{F}$ be a class of functions mapping $\mathbb{R}^k$ onto $\mathbb{R}$. Let $\varphi\colon \mathbb{R} \to \mathbb{R}^+$ be a κ-Lipschitz function such that $\varphi(\varepsilon) \geq \mathbb{1}_{\varepsilon>0}$. Let B be a uniform upper bound on $\varphi\big(-f(\varepsilon)\ell\big)$. Let $\{(x_i,\ell_i)\}_{i=1}^n \sim \mathbb{R}^k \times L$ and $\{\sigma_i\}_{i=1}^n$ be i.i.d. in $\{1,-1\}$ (Rademacher random signs). Then, with probability at least $1-\delta$,*

$$\mathscr{E}_\mathscr{F}(\hat{f}_n) = R_\varphi(\hat{f}_n) - R_\varphi(f^*) \leq 4\kappa e\left[\sup_{f\in\mathcal{F}}\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i f(x_i)\right|\right] + B\sqrt{\frac{\log(1/\delta)}{2n}},$$

where the expectation is taken w.r.t. $\{\sigma_i, x_i\}_{i=1}^n$.

Naturally, in our case of learning through empirical distributions, the expected risk (and thus the performance of $\hat{f}_n$) crucially depends on the feature construction step mapping each data distribution sample $\mathscr{S}_j$ onto a k-dimensional real-valued vector.

4.C Error bounds in the cause-effect pairs setting for kernel-based embeddings

Lopez-Paz et al. [2015] exploits functional representations of empirical distributions based on Reproducing Kernel Hilbert Spaces (RKHS). Letting k denote a kernel on the sample space, given an n-sample $x_1 \dots x_n$ drawn iid from distribution P, a functional representation of these samples is given as:

$$\mu_k(P) = \frac{1}{n}\sum_{i=1}^{n} k(x_i, \cdot) \tag{4.19}$$

with $\mu_k(P)$ being a function in the RKHS $\mathscr{H}_k$ associated with kernel k. This representation enables to refine Theorem 4.1 as follows:

Theorem 4.2. *[Lopez-Paz et al. 2015] With same notations as in Theorem 4.1, let* $\mathcal{H}_k$ *denote the RKHS associated with some bounded, continuous kernel function k, such that* $\sup_z k(z,z) \leq 1$. *Let* $\mathcal{F}_k$ *be a class of functions mapping* $\mathcal{H}_k$ *to* $\mathbb{R}$ *with Lipschitz constant uniformly bounded by* $\kappa_{\mathcal{F}}$. *Let* $\varphi\colon \mathbb{R} \to \mathbb{R}^+$ *be a* κ*-Lipschitz function such that* $\phi(\varepsilon) \geq \mathbb{1}_{\varepsilon>0}$. *Let* $\varphi\big(-f(\varepsilon)\ell\big) \leq B$ *for every* $f \in \mathcal{F}_k$, $\varepsilon \in \mathcal{H}_k$, *and* $\ell \in L$. *Then, with probability greater than* $1-\delta$ *(over all sources of randomness)*

$$\mathscr{E}_{\mathscr{F}}(\hat{f}_n) = R_\varphi(\hat{f}_n) - R_\varphi(f^*) \leq 4\kappa R_n(\mathcal{F}_k) + 2B\sqrt{\frac{\log(2/\delta)}{2n}} + \frac{4\kappa\kappa_{\mathcal{F}}}{n}\sum_{i=1}^{n}\left(\sqrt{\frac{\mathrm{e}_{z\sim P_{S_j}}[k(z,z)]}{n_i}} + \sqrt{\frac{\log(2n/\delta)}{2n_i}}\right),$$

with $R_n(\mathcal{F}_k) = \mathrm{e}\left[\sup_{f\in\mathcal{F}_k}\frac{1}{n}\left|\sum_{i=1}^n \sigma_i f(x_i)\right|\right]$ *the Rademacher complexity of* $\mathcal{F}_k$.

Theorem 4.2 represents the bound for our causal pairs learning problem with kernel embeddings as features. Compared to Theorem 4.1, an additional term is added to cope with the feature construction step: if the kernel embedding manages to capture all information out of the distributions useful for the classification of the causal pairs, then we obtain the bound given by Theorem 4.1.

Lopez-Paz et al. [2015] goes towards a more scalable approach for kernel computation relying on Fourier-based approximations of real-valued and shift invariant kernels [Rudin, 1962], defined as:

$$\forall x, x' \in \mathbb{R}^{\mathrm{d}}, \mathrm{k}(\mathrm{x}, \mathrm{x}') = 2\mathrm{C}_{\mathrm{k}}\, \mathrm{e}_{\mathrm{w,b}} \left[\cos(\langle \mathrm{w}, \mathrm{x} \rangle + \mathrm{b}) \cos(\langle \mathrm{w}, \mathrm{x}' \rangle + \mathrm{b}) \right] \tag{4.20}$$

where $w \sim \frac{1}{C_k} p_k$, $b \sim \mathscr{U}[0, 2\pi]$, $p_k \colon \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$ is the positive and integrable Fourier transform of k, and $C_k = \int_{\mathbb{Z}} p_k(w) dw$.

For example, the shift-invariant Gaussian kernel with kernel width γ can be approximated using Eq. 4.20 with $p_k(w) = Pr(w|\mathscr{N}(0, 2\gamma I))$, and $C_k = 1$ [Rahimi and Recht, 2008, 2009, Lopez-Paz et al., 2015], with linear complexity, as:

$$\hat{v}_m^x(\cdot) = \frac{1}{m} \sum_{i=1}^{m} 2C_k \cos(\langle w, x \rangle + b) \cos(\langle w, \cdot \rangle + b) \tag{4.21}$$

with (w_j, b_j) iid sampled in $\mathbb{N}_{0,2} \times [0, 2\pi]$. After Lopez-Paz et al. [2015], this approximation enables to refine Theorem 4.1:

Lemma 4.1. *[Lopez-Paz et al., 2015] Let $\mathbb{Z} = \mathbb{R}^{\mathrm{d}}$. For any shift-invariant kernel k s.t. $\sup_{z \in \mathbb{Z}} k(z, z) \leq 1$, any fixed $S = \{z_i\}_{i=1}^n \subset \mathbb{Z}$, any probability distribution Q on $\mathbb{Z}$, and any $\delta > 0$, it comes:*

$$\left\| \mu_k(P_S) - \frac{1}{n} \sum_{i=1}^{n} \hat{g}_m^{z_i}(\cdot) \right\|_{L_2(Q)} \leq \frac{2C_k}{\sqrt{m}} \left(1 + \sqrt{2 \log(n/\delta)} \right)$$

with probability greater than $1 - \delta$ over $\{(w_i, b_i)\}_{i=1}^m$.