Universitat Politècnica de Catalunya
Universitat Rovira i Virgili
Universitat de Barcelona

M. Sc. Artificial Intelligence

Master's Thesis Written During an Erasmus
Exchange at TU Delft, The Netherlands

Semi-Generative Modelling: Learning With Cause and Effect Features

Author:
Julius von Kügelgen

Supervisors:
Dr. Marco Loog
Alexander Mey

April 17, 2018

UNIVERSITAT ROVIRA i VIRGILI

UNIVERSITAT DE BARCELONA

Semi-Generative Modelling: Learning With Cause and Effect Features

Julius von Kügelgen julisuvk@gmail.com
Pattern Recognition Laboratory
Delft University of Technology, The Netherlands

Facultat d'Informàtica de Barcelona
Universitat Politècnica de Catalunya, Spain

Editor:

Abstract

Current methods for covariate shift (CS) adaptation use unlabelled data for computing importance weights or finding new feature representations, before training a model on the transformed labelled sample. When the amount of labelled data is the bottleneck, however, we would like to not only adapt, but also actively improve the supervised source model with unlabelled data. Yet, recent findings suggest that such semi-supervised learning is not possible in a causal setting ($X \to Y$) as is usually assumed implicitly in standard CS. We thus consider a case of CS where prior causal inference or expert knowledge has identified some features as effects, and show how this setting—when analysed from a causal perspective—gives rise to a semi-generative modelling framework: $P(Y, X_{\text{eff}}|X_{\text{cau}}, \theta)$. Our approach combines concepts from invariant prediction and semi-supervised learning, and at its heart is the idea to impose a model constraint by unsupervised learning of a map from causes to effects. Finally, our method for learning with cause and effect features is not exclusive to CS but provides a general approach for semi-supervised learning in changing environments when causal knowledge is available.

Keywords: domain adaptation, covariate shift, semi-supervised learning, independent causal mechanisms, semi-generative model

1. Introduction

With recent advances in both algorithms and hardware, the amount of high-quality, labelled training data is becoming the bottleneck for many machine learning tasks. Methods making use of unlabelled data are thus an active area of research with great potential. Semi-supervised learning (SSL) aims to improve a model of $P(Y|X)$ via a better estimate of the marginal $P(X)$ by linking these quantities through certain assumptions (Chapelle et al., 2010). Domain adaptation (DA), on the other hand, intends to adapt a model from a source domain to a different, but related target domain (Quionero-Candela et al., 2009; Pan and Yang, 2010). The subject of this paper is combining unsupervised DA under covariate shift (CS) (Sugiyama and Kawanabe, 2012) with SSL.

Some current methods for unsupervised CS adaptation use unlabelled data for importance reweighting of source-domain training data (Shimodaira, 2000; Sugiyama et al., 2007); others use it to find a transformation which leads to domain-invariant features (e.g., Pan

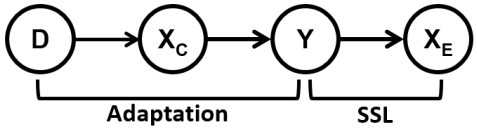


Figure 1: Our setting - CS holds, SSL possible.

et al., 2011). The final target model is then trained on the reweighted/transformed labelled source sample. Such methods thus focus on the adaptation task which can loosely be thought of as a preprocessing step, but do not involve unlabelled data in the model fitting. When the amount of labelled data is the bottleneck, however, we would like to not only adapt, but also actively improve the supervised source model with unlabelled data.

However, recent work on the independence of causal mechanisms (ICM) (Janzing and Schölkopf, 2010) suggests that such SSL is not possible in a causal learning setting ($X \to Y$) (Schölkopf et al., 2012). Since standard CS implicitly treats all features as causal (Storkey, 2009) (see Section 2.2 for more details), further assumptions are necessary to make SSL work in such a setting. We thus consider an unsupervised CS-adaptation setting for which the amount of labelled data is the main limiting factor, but for which the true causal structure is known. In doing so, we attempt to answer the following question: *How can causal knowledge guide a more principled and more effective use of unlabelled data in changing environments?*

Specifically, we assume that through prior causal inference, expert knowledge, or background information some features have been identified as effects X_E of a target variable Y while the remaining features represent causes X_C. This setting with effect features allows to combine CS adaptation and SSL, as shown schematically in Fig. 1. Since CS is assumed, it is required that the domain shift (D) does not directly influence X_E (as the v-structure at X_E would otherwise induce a domain dependence of Y, see Fig. 2a).

Two examples of real world scenarios compatible with our idea of prediction from cause and effect features are the following: (i) predicting disease, Y, from risk factors like genetic predisposition or smoking, X_C, and symptoms, X_E; while we might have (possibly unlabelled) data from multiple geographical regions or demographic groups leading to different distributions over risk factors ($D \to X_C$), we would not necessarily expect this to affect the behaviour of the disease itself ($X_C \to Y \to X_E$); (ii) predicting the hidden intermediate state Y of a physical system with inputs X_C and outputs X_E; again, we might have data from various experiments with differing input distributions ($D \to X_C$), but the laws of physics or nature ($X_C \to Y \to X_E$) would not be expected to change.

Our approach is a semi-generative model, $P(Y, X_E|X_C, \theta)$, which combines ideas from invariant prediction (Peters et al., 2016) and SSL, and which arises from the asymmetric roles played by cause and effect features in our setting and under the ICM assumption. This framework leads to a domain invariant model by conditioning on X_C and naturally allows to include unlabelled data in the fitting process by summing or integrating out Y. The latter allows for unsupervised learning of a map from cause to effect features, $P(X_E|X_C)$, which can be used to impose a soft model constraint.

1.1 Contributions and Organisation of the Paper

In Section 2 we present preliminaries and previous work on which this paper build upon in a common context. Section 3 then forms the theoretical main-part, where we clearly state problem setting and assumptions, and derive our semi-generative modelling framework. In Section 4 we show how our approach can be applied to classification and certain regression problems in practice by describing our experiments on synthetic and real data sets. Empirical results of these experiments are presented in Section 5. In Section 6 we critically discuss our results in the context of related work and comment on the general applicability of our approach. Finally, we summarise our findings in Section 7. Some supplementary information is included in the Appendices.

2. Preliminaries & Previous Work

This section covers preliminaries and related work such as the unsupervised domain adaptation setting, the covariate shift assumption, causal vs predictive models, the independence of causal mechanisms, the co-training algorithm, and hybrid generative-discriminative models. It may be skipped by a reader familiar with these concepts.

2.1 Unsupervised Domain Adaptation (DA)

In the unsupervised domain adaptation (DA) setting we have access to unlabelled data for the same task but from a different distribution (also called domain in this context). This setting occurs, for example, when training and test sets are not from the same distribution (Quionero-Candela et al., 2009). Formally, we are given a labelled sample $S_S = \{x^i, y^i\}_{i=1}^{n_S}$ from the source domain $P(X, Y|D = 0)$ and an unlabelled sample $S_T = \{x^j\}_{j=n_S+1}^{n_S+n_T}$ from the target domain $P(X, Y|D = 1)$, where D is the domain indicator.[1] The aim of unsupervised DA is to find a mapping from the shared feature to label space, $f : \mathcal{X} \to \mathcal{Y}$, which minimises the expected target loss w.r.t. a given loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. An optimal f is

$$f^* \in \operatorname{argmin}_f \mathbb{E}_{P(\cdot|D=1)}\Big[L\big(f(X), Y\big)\Big] = \operatorname{argmin}_f \int P(x, y|D = 1) L\big(f(x), y\big) \mathrm{d}x \, \mathrm{d}y. \quad (1)$$

Since the target distribution $P(X, Y|D = 1)$ is not only unknown but also cannot be estimated due to the lack of target labels, this learning problem is inherently ill posed. Thus, further assumptions about the similarities of source- and target domain are necessary.

2.2 Covariate Shift (CS)

One of the most commonly used assumptions for unsupervised DA is covariate shift (CS). CS states that the difference between domains is only due to a shift in the marginal distribution over covariates, or features, while the conditional distribution of labels given features remains invariant: $P(X|D = 0) \neq P(X|D = 1)$, but $P(Y|X, D = 0) = P(Y|X, D = 1) = P(Y|X)$. This can also be expressed as $Y \perp\!\!\!\perp D \,|\, X$ and is depicted as a graphical model in Fig. 2a. Note that the similar graph shown in Fig. 2b does not satisfy the CS assumption due to the v-structure at X, which causes Y to be conditionally dependent of D given X.

1. Another common notation is: $P(\cdot|D = 0) = P_S(\cdot)$ and $P(\cdot|D = 1) = P_T(\cdot)$.

(a) CS holds (b) CS does not hold

Figure 2: Two scenarios for domain adaptation: (a) when X causes Y, CS holds since $Y \perp\!\!\!\perp D|X$; (b) when Y causes X, CS does not hold since the V-structure at X makes Y dependent on the domain D given X. (It is usually assumed, that the domain shift does not directly affect the target Y, for work on such "target shift" see, e.g., Zhang et al. (2013).)

When changes in environment lead to CS (also termed "domain shift"), it makes sense to consider the edge $(D \to X)$ as being truly causal, and such CS can thus be seen as treating all features as causes of Y. We note, however, that other processes such as, for example, sample selection bias can also be seen as instance of CS. For the rest of this work though, we focus on CS in the sense of domain shift and use these words analogously.

As a consequence of CS, the target distribution can be rewritten as

$$P(X, Y|D = 1) = P(X|D = 1)P(Y|X) = w(X)P(X, Y|D = 0) \tag{2}$$

where $w(X) = \frac{P(X|D=1)}{P(X|D=0)}$ are so-called importance weights (Shimodaira, 2000; Sugiyama et al., 2007). This implicitly assumes that the target support is contained in the source support; however, different techniques exist to circumvent this requirement, see, e.g., Cortes et al. (2010) for some theoretical analysis of importance weighting.

Since the rightmost expression in Eq. (2) does not involve target labels, it can be used to approximate the expectation in Eq. (1) empirically:

$$\mathbb{E}_{P(\cdot|D=1)}\Big[L\big(f(X), Y\big)\Big] = \mathbb{E}_{P(\cdot|D=0)}\Big[w(X)L\big(f(X), Y\big)\Big] \approx \frac{1}{n_S}\sum_{i=1}^{n_S} w(x^i)L\big(f(x^i), y^i\big).$$

This corresponds to training a model on the importance-weighted source data where the weights can be interpreted as measure of how representative a certain source example is of the target domain (Sugiyama and Kawanabe, 2012).

Another family of approaches for unsupervised DA under CS avoids the use of importance weights, and is based instead on finding domain-invariant features in a common subspace (e.g., Gong et al., 2012; Fernando et al., 2013). Generally speaking, these methods first project the inputs to a new space $\mathcal{X}'$ via some map $\phi : \mathcal{X} \to \mathcal{X}'$, and then train a model on these transformed features. ϕ is usually chosen as to minimise the discrepancy between domains: $P(\phi(X)|D = 0) \approx P(\phi(X)|D = 1)$. Note that $P(\phi(x)|D = 0) = P(\phi(x)|D = 1)$ implies that $w(\phi(x)) = 1$, thus eliminating the need for importance weights when training on fully domain invariant features. Various criteria have been used to measure the discrepancy between domains from finite data, such as, e.g., MMD (Pan et al., 2011), HSIC (Yan et al., 2017), mutual information with a domain indicator (Shi and Sha, 2012), or performance of a domain classifier (Ganin et al., 2016).

2.3 Causal Models

Most of the field of machine learning is concerned with predictive models which learn dependencies and correlations between variables from training data. Such models are generally

good at making predictions for observational data, i.e., data observed in one environment, and therefore from the same distribution; however, when the underlying system is intervened upon—leading to a change in distribution—such predictive models can show arbitrarily poor performance. Causal models (Pearl, 2000; Spirtes et al., 2000), on the other hand, by capturing *causation* among variables, rather than *correlation*, are able to also predict the behaviour of a system under interventions. They are thus a more general model class than purely predictive models. The drawback of causal models, however, is their need for interventional training data which is often not available; learning cause-effect relationships from purely observational data is difficult in general, and, without further assumptions, sometimes even impossible.[2] The interested reader is referred to Pearl (2009) for an overview of causal inference techniques, and to Peters et al. (2017) for a more recent account.

Formally, a structural causal model (SCM) (Pearl, 2000) over a set of random variables $\{X_i\}_{i=1}^d$ is defined by the set of equations

$$X_i := f_i\big(\mathrm{Pa}(X_i), N_i\big) \qquad \text{for} \qquad i = 1, \ldots, d \tag{3}$$

where $\mathrm{Pa}(X_i)$ is the set of causal parents of X_i (i.e., those variables having a direct causal effect on X_i), N_i are mutually independent, random noise variables, and f_i are deterministic functions. An illustration of such models in the bivariate case can be found in Fig. 3. Since N_i are stochastic, the set of Equations (3) induces a distribution P over $\{X_i\}_{i=1}^d$ which depends on the noise distributions. The corresponding directed acyclic graph (DAG) with $\{X_i\}_{i=1}^d$ as nodes and $X_i \to X_j$ an edge iff. $X_i \in \mathrm{Pa}(X_j)$ can be thought of as a causal Bayesian network (Pearl, 1985), i.e., a directed probabilistic graphical model (PGM) in which the direction of edges truly indicates causal influence. The joint distribution factorises over this causal Bayesian network as

$$P(X_1, \ldots, X_d) = \prod_{i=1}^{d} P(X_i|\mathrm{Pa}(X_i)). \tag{4}$$

The true power of SCMs over general PGMs is that the former can model interventions, while the latter cannot. An intervention corresponds to setting one of the X_i to a fixed value x_i, and is denoted using Pearl's do-operator as $do(X_i = x_i)$. In the SCM framework this can be modelled by replacing all occurrences of X_i in Equations (3) by the new assignment x_i. The newly induced distribution (as there is no stochastic contribution from N_i anymore) is denoted $P(\cdot|do(X_i = x_i))$. It is obtained by replacing the factor $P(X_i|\mathrm{Pa}(X_i))$ in Eq. (4) with $\delta(X_i = x_i)$ and all occurrences of $X_i \in \mathrm{Pa}(X_j)$ by x_i. We stress here that intervening on a variable is fundamentally different from conditioning on it: an *intervention* on X_i only affects its causal descendants (as these are the only equations in which X_i occurs), but not its causal ancestors; *conditioning* on X_i, on the other hand, usually affects both its descendants and ancestors.

While causal models are not necessary for the standard supervised and i.i.d. learning setting for which distributions remain unchanged, they can play an important role in analysing and understanding variations to this classical scheme. This is reflected in numerous recent works drawing on ideas from causality to tackle ML problems such as, for example, data

2. E.g., for a linear model with Gaussian noise it is not possible to distinguish between $X \to Y$ and $X \leftarrow Y$.

(a) Causal learning, SSL not possible (b) Anticausal learning, SSL possible

Figure 3: Under the ICM assumption, (a) SSL should not be possible if $Y := f_Y(X, N_Y)$, as $P(X)$ and $P(Y|X)$ are independent in this case; (b) if $X := f_X(Y, N_X)$ then SSL is, in principle, possible as $P(X)$ and $P(Y|X)$ may share some information. [Figure reproduced from Schölkopf et al. (2012)]

fusion (Pearl and Bareinboim, 2014; Bareinboim and Pearl, 2016), transfer learning (Magliacane et al., 2017), multitask learning (Rojas-Carulla et al., 2015), and domain adaptation (Zhang et al., 2015).

2.4 Independence of Causal Mechanisms (ICM)

An assumption with roots in causal modelling which has received attention in recent years is the independence of causal mechanisms (ICM) which was inspired by Lemeire and Dirkx (2006) and formalised in terms of algorithmic complexity by Janzing and Schölkopf (2010). At its heart is the assumption that the f_i in Equations (3) are mutually independent, so that the conditionals $P(X_i|\text{Pa}(X_i))$, or *causal mechanisms*, on the RHS of Eq. (4) represent "autonomous modules that do not inform or influence each other" (Parascandolo et al., 2017).

The implications of this assumption for different learning settings have been discussed by Schölkopf et al. (2012), which is one of the main inspirations for the current work. In particular, the authors argue that since SSL relies on linking $P(X)$ and $P(Y|X)$ it should not be possible in the causal direction (Fig. 3a) as $P(X)$ and $P(Y|X)$ represent independent causal mechanisms in this case. In an "anticausal" learning setting (Fig. 3b), on the other hand, when the causal mechanisms are $P(Y)$ and $P(X|Y)$, $P(X)$ and $P(Y|X)$ might still exhibit some dependence so that SSL is, in principal, possible. Empirical evidence supports the validity of this argument (Schölkopf et al., 2012).

2.5 Other Previous Work

In addition to those referred to already in the previous sections, we review similarities and differences to some other previous work below.

Co-Training: One of the earlier and most influential works on combining labelled and unlabelled data is the "Co-Training" algorithm by Blum and Mitchell (1998). Motivated by boosting performance of a web-page classifier using a large unlabelled sample the authors assume that: (i) the feature set can be split into two disjoint views, $X = (X_1, X_2)$, which are conditionally independent given Y; and (ii) each view is sufficient for learning the task given enough data. They then train a weak classifier on each view, and use these to iteratively label unlabelled examples and add them to the training set. While there are some clear parallels to our approach, e.g., assumption (i) above also holds in our setting, we do *not*

assume that each view (X_C and X_E in our case) is sufficient on its own (assumption (ii) above). More importantly, our approach does not rely on assigning labels to unlabelled data and is therefore not part of the family of self-learning approaches to SSL of which co-training is a member.

Hybrid Generative-Discriminative Models: Since Ng and Jordan (2002) shed light on some of the advantages and disadvantages of discriminative vs. generative models, some works have attempted to combine these two approaches for SSL. Multi-conditional learning (McCallum et al., 2006) considers objective functions like $P(Y|X)^\alpha P(X)^\beta$ and has been successfully applied to SSL (Druck et al., 2007). An example of combining SVM and naive Bayes for SSL can be found in Jiang et al. (2013). The idea of combining supervised and unsupervised components in the objective function is also reflected in our approach. However, the above-mentioned works train both a generative and a discriminative model using the former for unlabelled and the latter for labelled data; on the other hand, we only train *one* model which is neither fully generative nor fully discriminative, and which can be used to include both unlabelled and labelled data.

Learning with Cause and Effect Features: To the best of our knowledge no previous works explicitly consider learning from both cause and effect features. The closest may be that of Kang and Tian (2006) where—even though not in a causal context—they consider two sets of features X_1 and X_2 on either side of Y. The resulting likelihood $P(Y|X_1)P(X_2|Y)$ is similar to ours, but is not used in combination with unlabelled data.

3. Semi-Generative Modelling

In this Section we explicitly state our assumptions, use them to derive our semi-generative framework, and show how it gives rise to a modelling approach which naturally allows for including unlabelled data, while remaining robust to distribution shifts over causal features.

3.1 Assumptions

We consider the unsupervised domain adaptation problem of predicting the outcome of a random variable Y in a target domain ($D = 1$) from an observation of a set of random variables, or features, X. We assume that the set of features can be partitioned into two disjoint, non-empty sets: $X = X_C \cup X_E$ and $X_C \neq \varnothing \neq X_E$. As training data we are given a labelled, typically small set of observations $S_S = \{(x_C^i, y^i, x_E^i)\}_{i=1}^{n_S}$ from a source domain ($D = 0$) and an unlabelled, typically large set ob observations $S_T = \{(x_C^j, x_E^j)\}_{j=n_S+1}^{n_S+n_T}$ from a target domain ($D = 1$). Moreover, we make the following main assumption.

Assumption 1 (Known Causal Structure & ICM) *The relationship between the variables X_C, Y, X_E and the domain indicator D is accurately captured by the following SCM:*

$$\begin{aligned} X_C &:= f_C(D, N_C) \\ Y &:= f_Y(X_C, N_Y) \\ X_E &:= f_E(Y, N_E) \end{aligned} \tag{5}$$

where N_C, N_Y, and N_E are mutually-independent random noise variables, and the functions f_C, f_Y, and f_E represent independent causal mechanisms (in the sense of Section 2.4).

This SCM is shown schematically in Fig. 4. The (unknown) distributions over noise variables together with Equations (5) induce a distribution over (X_C, Y, X_E) which depends on D: $P(X_C, Y, X_E|D)$. We will refer to the two distributions $P(X_C, Y, X_E|D = 0)$ and $P(X_C, Y, X_E|D = 1)$ as source and target distributions, or domains, respectively.[3]

3.2 Analysis

From Eq. (4) and Assumption 1 it follows that the distribution P factorises as

$$P(X_C, Y, X_E|D) = P(X_C|D)P(Y|X_C)P(X_E|Y). \tag{6}$$

where the factors on the RHS correspond to the three independent causal mechanisms. This factorisation can be used to show that the CS assumption is satisfied in our setting (as intended by construction).

Proposition 1 *CS holds for the setting described by Assumption 1.*
Proof *By the factorisation of P given in Eq. (6), it follows that:*

$$P(Y|X_C, X_E, D) = \frac{P(X_C, Y, X_E|D)}{P(X_C, X_E|D)} = \frac{P(Y, X_E|X_C)P(X_C|D)}{P(X_E|X_C)P(X_C|D)} = \frac{P(Y, X_E|X_C)}{P(X_E|X_C)}.$$

The last equality shows that the conditional distribution does not depend on the domain D. ■

In fact, we can make a stronger statement than Proposition 1. This is due to the assumed chain-like problem structure which results in a *direct* shift only in the distribution over causes, $P(X_C|D)$. Whereas this change in distribution is propagated through the mechanisms $P(Y|X_C)$ and $P(X_E|Y)$ thereby also affecting Y and X_E, the only shift that needs to be corrected for is that in X_C. This follows directly from the factorisation (6):

$$P(X_C, Y, X_E|D = 1) = w(X_C)P(X_C, Y, X_E|D = 0) \tag{7}$$

where $w(X_C) = \frac{P(X_C|D=1)}{P(X_C|D=0)}$ are importance weights (Shimodaira, 2000). Hence, conditioning on X_C is sufficient to obtain domain-invariance.

Eq. (7) also reveals a first side-benefit of identifying some features as effects in a CS setting: training a model on reweighted source data only requires estimation of the distribution over causes, but not over effects. Since the correct weights are generally not known, estimating such ratios of (potentially high-dimensional) distributions from limited data is often a challenge a practice. Performing this estimation over a lower-dimensional space could thus be advantageous.

The focus of our work, however, lies on actively improving the source model from unlabelled data, rather than on the adaptation task. By improvement here we refer to obtaining a better estimate of the conditional distribution $P(Y|X_C, X_E)$ and ultimately a lower error rate on an unseen test set drawn from the target domain. According to the ICM assumption, such semi-supervised learning is not possible in the causal direction but might work in the anticausal direction (see Fig. 3) (Schölkopf et al., 2012). This has the following implications for our setting:

3. Note that even though we focus on the case $D \in \{0, 1\}$ here, it should be simple to include additional labelled or unlabelled data from different sources as in domain generalisation.

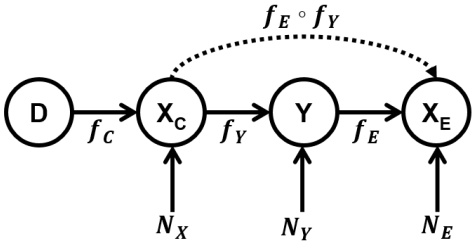


Figure 4: A schematic illustration of the assumed causal problem structure. The dashed arrow illustrates the main idea of our approach, namely learning a map from cause to effect features from unlabelled data and using it as a soft model constraint.

- The distribution over causal features does not share any information with the conditional distribution of Y. A better estimate of $P(X_C|D)$ obtainable from unlabelled data will therefore not help to improve our estimate of the conditional, and so explicitly modelling the cause distribution unnecessarily introduces model complexity without apparent benefits.
- $P(Y)$ and $P(X_E|Y)$ are independent mechanisms, but $P(X_E)$ and $P(Y|X_E)$ may contain shared information. We can therefore hope to improve our estimate of the conditional via a better estimate of $P(X_E)$ from the unlabelled sample. This suggests explicitly modelling the distribution over effects in our case.

One way to convince oneself of the above is from the domain invariant form of the conditional, $\frac{P(Y,X_E|X_C)}{P(X_E|X_C)}$, where X_C only appears as a variable which we condition on, whereas X_E appears explicitly. We note here that while it is also possible to write the conditional $P(Y|X_C, X_E)$ differently, only conditioning on X_C leads to a domain invariant form. Such invariance is required as those terms involving Y can only be estimated from the source domain, while we wish to make predictions for the target domain.

Another way to understand the different roles played by cause and effect features is from the SCM (5) in Assumption 1. Even if one could perfectly learn the data generating process for the causal features, $X_C := f_X(D, N_C)$, this does not reveal anything about Y. The generating process for the effects, $X_E := f_E(Y, N_E)$, on the other hand, clearly depends on Y. In particular, when Y is unknown but X_C is known—as is the case for the large unlabelled sample from the target domain—we can substitute for Y to obtain

$$X_E := f_E(Y, N_E) = f_E\big(f_Y(X_C, N_Y), N_E\big). \tag{8}$$

Equation (8) above demonstrates our main idea: by learning a (noisy) map from causes to effects from unlabelled data we hope to improve our estimates of the functions f_Y and f_E, and thereby also our predictive model. Such a map can be viewed as a noisy composition of functions, $f_E \circ f_Y$, as indicated by the dashed arrow in Fig. 4.

In terms of the distribution P this idea corresponds to improving our estimate of $P(X_E|X_C)$. Factorising the distribution from which the unlabelled sample is drawn as

$$P(X_C, X_E|D = 1) = P(X_C|D = 1)P(X_E|X_C), \tag{9}$$

helps to illustrate the point of our approach. Whereas CS adaptation by importance weighting would only use the first term on the RHS of Eq. (9) thereby disregarding half the unlabelled sample, our approach will make full use of all unlabelled data by using the second term for semi-supervised learning.

3.3 Modelling Approach

The previous analysis of the asymmetric roles played by cause and effect features suggests using a model of the form $P(Y, X_E|X_C, \theta)$. We refer to this modelling framework as semi-generative, as it can be seen as an intermediate between a fully generative, $P(X_C, Y, X_E|\theta)$, and a fully discriminative, $P(Y|X_C, X_E, \theta)$, framework. As opposed to a fully generative model, the semi-generative model is domain invariant due to conditioning on X_C. At the same time, the semi-generative framework also allows including unlabelled data by summing (if $\mathcal{Y}$ is discrete) or integrating (if $\mathcal{Y}$ is continuous) out Y,

$$P(X_E|X_C, \theta) = \sum\!\!\!\!\!\!\int_{y\in\mathcal{Y}} P(Y = y, X_E|X_C, \theta)\,[\mathrm{d}y] \tag{10}$$

which is not possible for fully discriminative models which condition on all features. For our setting, a semi-generative framework thus combines the best from both worlds: domain invariance and the possibility to include unlabelled data in the fitting process.

As it does not depend on labels, we will also refer to Eq. (10) as the unsupervised model. It is clear that we can always obtain the unsupervised model for classification tasks, while for regression we are restricted to special types of submodels for which the integral can be computed analytically. When other models are desired, approximating the integral is an option, but this is left for future work.

Moreover, the semi-generative formulation has another advantage in our setting: it factorises into the two independent mechanisms

$$P(Y, X_E|X_C, \theta) = P(Y|X_C, \theta_Y)P(X_E|Y, \theta_E) \tag{11}$$

where $\theta = (\theta_Y, \theta_E)$ are the parameters of the two submodels. Note that this can also help avoid problems with missing data. E.g., if x_C^i is missing for some i, the pair (y^i, x_E^i) can still be used to train $P(X_E|Y, \theta_E)$.

With our model $P(Y, X_E \,|\, X_C, \theta)$, a way to include unlabelled data via Eq. (10), and the factorisation into two submodels given by Eq. (11) we are now ready to give a high-level summary of our approach:

1. Train two supervised submodels, $P(Y|X_C, \theta_Y)$ and $P(X_E|Y, \theta_E)$, on labelled pairs (x_C^i, y^i) and (y^i, x_E^i), such that the corresponding unsupervised model (10) 'agrees well' with the unlabelled cause-effect pairs (x_C^j, x_E^j).

2. Construct the probabilistic conditional from $P(Y|X_C, \theta_Y)$ and $P(X_E|Y, \theta_E)$ as:

$$P(Y|X_C, X_E, \theta) = \frac{P(Y|X_C, \theta_Y)P(X_E|Y, \theta_E)}{\sum\!\!\!\!\!\!\int_{y\in\mathcal{Y}} P(Y = y|X_C, \theta_Y)P(X_E|Y = y, \theta_E)\,[\mathrm{d}y]} \tag{12}$$

A likelihood-based version of the above is described in detail in the next section and summarised for classification in Algorithms 1 and 2 in Appendix B.

3.4 (Log-)Likelihoods

The average supervised source likelihood, $P(Y, X_E|X_C, \theta)$, of our model given the observed labelled data is given by:

$$\mathcal{L}_S(\theta) = \mathcal{L}(\theta|S_S) = \prod_{i=1}^{n_S} P(y^i, x_E^i|x_C^i, \theta)^{\frac{1}{n_S}} = \prod_{i=1}^{n_S} P(y^i|x_C^i, \theta_Y)^{\frac{1}{n_S}} P(x_E^i|y^i, \theta_E)^{\frac{1}{n_S}}.$$

We consider average (log-)likelihoods in order to be able to compare models trained on different amounts of data; such averaging does, of course, not affect the resulting parameter MLEs. The corresponding average log-likelihood ℓ is given by

$$\ell_S(\theta) = \frac{1}{n_S} \sum_{i=1}^{n_S} \log P(y^i|x_C^i, \theta_Y) + \log P(x_E^i|y^i, \theta_E) \tag{13}$$

and can be maximised w.r.t. θ_Y and θ_E separately. Closed form MLEs are thus often available for simple enough models. The same holds true for $\mathcal{L}_{WS}$ and ℓ_{WS} which additionally importance-reweigh term i in $\mathcal{L}_S$ and ℓ_S, respectively, by $w(x_C^i)$ (Shimodaira, 2000).

$$\ell_{WS} = \frac{1}{n_S} \sum_{i=1}^{n_S} w(x_C^i)\big(\log P(y^i|x_C^i, \theta_Y) + \log P(x_E^i|y^i, \theta_E)\big) \tag{14}$$

Note that importance weights arise from an empirical source approximation to the expected target loss and are thus not subject to logarithms.

Our approach suggests additionally including unlabelled data via the average unsupervised target likelihood, $P(X_E|X_C, \theta)$:

$$\mathcal{L}_T(\theta) = \mathcal{L}(\theta|S_T) = \prod_{j=n_S+1}^{n_S+n_T} P(x_E^j|x_C^j, \theta)^{\frac{1}{n_T}} = \prod_{j=n_S+1}^{n_S+n_T} \Big(\sum\!\!\!\!\!\!\int_{y\in\mathcal{Y}} P(y|x_C^j, \theta_Y) P(x_E^j|y, \theta_E)[\mathrm{d}y] \Big)^{\frac{1}{n_T}}$$

where the last equality follows from Equations (10) and (11). The corresponding target log-likelihood is

$$\ell_T(\theta) = \frac{1}{n_T} \sum_{j=n_S+1}^{n_S+n_T} \log \Big(\sum\!\!\!\!\!\!\int_{y\in\mathcal{Y}} P(y|x_C^j, \theta_Y) P(x_E^j|y, \theta_E)[\mathrm{d}y] \Big).$$

We propose to combine labelled and unlabelled data in the pooled likelihood as

$$\mathcal{L}_P(\theta) = \mathcal{L}(\theta|S_S, S_T) = \mathcal{L}_S(\theta)^\lambda \mathcal{L}_T(\theta)^{1-\lambda}$$

with corresponding log-likelihood

$$\ell_P(\theta) = \lambda\, \ell_S(\theta) + (1-\lambda)\, \ell_T(\theta) \tag{15}$$

where the hyperparameter $\lambda \in (0, 1)$ interpolates between the average source- and target likelihoods and can be chosen depending on n_S and n_T. E.g., $\lambda_r = \frac{n_S}{n_S+n_T}$ gives equal weight to all observations and is therefore a natural choice, while changing this ratio leads to different weights for labelled and unlabelled data.

4. Experiments

In order to analyse our approach, we perform some regression and classification experiments on synthetic data sets. Moreover, we apply our semi-generative approach to a real world protein-signalling network data set (Sachs et al., 2005) to investigate its applicability in practice. For classification, we consider maximum likelihood and Bayesian approaches under both correct parametrisation and model misspecification to identify potential strengths and weaknesses in different settings. This Section contains the necessary experimental details to reproduce our results, and can be used as a guide to applying our semi-generative modelling approach for new classification and regression problems.

4.1 Estimators

Since our method requires making certain assumptions about the causal structure, and in order to make comparisons as fair as possible, we study our approach relative to versions of the source-only baseline and the importance-weighting technique, which take the known causal structure into account. In particular, we compare:

- $\hat{\theta}_S$: training a model on the labelled source data only. It ignores the unlabelled data and thus corresponds to no adaptation, but takes causal knowledge into account.

- $\hat{\theta}_{WS}$: training a model on the importance-weighted source data. It thus uses only the "X_C-part" of the unlabelled sample and ignores the "X_E-part", see Eq. (7). In our experiments we use known weights; however, in practice these would also have to be estimated from data.

- $\hat{\theta}_P$ **(our proposed estimator)**: training a model on the pooled data set combining unweighted labelled and unlabelled data via λ as explained in Section 3.4. (A second pooled estimator $\hat{\theta}_{WP}$ using weighted, instead of unweighted, labelled data was initially considered, but was then dropped for clarity of results as it suffered from the same problems as $\hat{\theta}_{WS}$ related to reweighting.)

- $\hat{\theta}_{LR}$: training a standard linear/logistic regression model on the joint feature set (X_C, X_E), i.e., ignoring the known causal structure. Where applicable, we report the performance of θ_{LR} for comparison.

Moreover, we compare our method to some state-of-the-art feature-transformation based approaches to unsupervised DA such as transfer component analysis (TCA, Pan et al., 2011), maximum independence domain adaptation (MIDA, Yan et al., 2017), subspace alignment (SA Fernando et al., 2013), geodesic flow kernel (GFK Gong et al., 2012), and information theoretical learning (ITL, Shi and Sha, 2012, for classification only). As would be the case without any knowledge of the causal structure, we apply these methods on the pooled data set (including source labels where applicable), and then train a linear/logistic regression model (as for $\hat{\theta}_{LR}$) on the new set of transformed features. We use the implementation and default parameters provided in the MATLAB domain adaptation toolbox by Yan, Ke (2016).

4.2 Model Fitting: Maximum Likelihood and Bayesian Approaches

For model fitting by maximum likelihood (ML), we minimize the negative log-likelihood (NLL) which acts as a proxy for the 0-1 loss in the case of classification and root mean squared error (RMSE) in the case of regression. The ML estimates $\hat{\theta}_S$, $\hat{\theta}_{WS}$, and $\hat{\theta}_P$ are thus found by maximising ℓ_S (13), ℓ_{WS} (14), and ℓ_P (15), respectively. We use analytical solutions where available and gradient descent to minimise the negative log-likelihoods otherwise.

For a Bayesian approach, we place a rather flat (i.e., with large σ) normal prior π on θ, so as to not include much prior knowledge on how the data is generated. We can then compute the log-posterior distribution up to additive constants:

$$\log P(\theta_P \,|\, S_S, S_T) = \log \pi(\theta) + (n_S + n_T)\ell_P(\theta) + \text{const.}, \tag{16}$$

In order to make predictions for new data $(x_C^{\text{new}}, x_E^{\text{new}})$, we estimate the required integral using a Monte Carlo approximation:

$$\begin{aligned} P(Y = y \,|\, x_C^{\text{new}}, x_E^{\text{new}}) &= \int_\theta P(Y = y \,|\, x_C^{\text{new}}, x_E^{\text{new}}, \theta) P(\theta \,|\, S_S, S_T)\, \mathrm{d}\theta \\ &\approx \frac{1}{K} \sum_{k=1}^{K} P(Y = y \,|\, x_C^{\text{new}}, x_E^{\text{new}}, \theta^{(k)}) \end{aligned}$$

where $\theta^{(k)}$ are samples from the posterior distribution. We use a Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) with a multivariate normal proposal distribution to sample from the corresponding unnormalised log-posterior distribution (16). In our experiments we use a step size of 0.1 and generate 10 randomly-initialised Markov chains of length 1100, in order to avoid the sampler getting stuck in local maxima of spiky, multimodal posteriors. Discarding the first 100 samples from each chain as burn-in, this leaves 10,000 samples for prediction. (Of course, more elaborate sampling schemes are possible.)

4.3 Synthetic Classification Experiments

For the classification experiments, we focus on binary classification. However, it is straightforward to extend our method to multi-class classification as well. To generate synthetic classification data sets with a simple linear decision boundary, but which are still somewhat challenging due to class-overlap, and which comply with our assumptions, we use the following SCM:

$$\begin{aligned} X_C &:= \begin{cases} \mu_C + \epsilon_C & \text{if} \quad D = 0, \\ -\mu_C + \epsilon_C & \text{if} \quad D = 1, \end{cases} \quad \text{where} \quad \epsilon_C \sim \mathcal{N}(0, \sigma_C^2) \\ Y &:= \begin{cases} 1 & \text{if} \quad \epsilon_Y \leq \operatorname{sigm}\big(s(X_C - m)\big), \\ 0 & \text{if} \quad \epsilon_Y > \operatorname{sigm}\big(s(X_C - m)\big), \end{cases} \quad \text{where} \quad \epsilon_Y \sim U(0,1) \\ X_E &:= \begin{cases} \mu_0 + \sigma_0 \epsilon_E & \text{if} \quad Y = 0, \\ \mu_1 + \sigma_1 \epsilon_E & \text{if} \quad Y = 1, \end{cases} \quad \text{where} \quad \epsilon_E \sim \mathcal{N}(0,1) \end{aligned} \tag{17}$$

with all noise variables ϵ_i mutually independent, and where sigm is the logistic sigmoid function, $\text{sigm}(x) = (1 + e^{-x})^{-1}$. This induces the distributions

$$\begin{aligned} Y \,|\, (X_C = x_C) &\sim \text{Bernoulli}\Big(\text{sigm}\big(s(x_C - m)\big)\Big) \\ X_E \,|\, (Y = y) &\sim \begin{cases} \mathcal{N}(\mu_0, \sigma_0^2) & \text{if} \quad y = 0 \\ \mathcal{N}(\mu_1, \sigma_1^2) & \text{if} \quad y = 1 \end{cases} . \end{aligned} \tag{18}$$

From (18) we can compute the unsupervised model (Eq. (10)) by summing out Y:

$$\begin{aligned} P(X_E = x_E | X_C = x_C, \theta) &= \sum_{y=0}^{1} P(X_E = x_E | Y = y, \theta_E) P(Y = y | X_C = x_C, \theta_C) \\ &= \big(1 + e^{-s(x_C - m)}\big)^{-1} \Big(\phi(x_E | \mu_0, \sigma_0^2) e^{-s(x_C - m)} + \phi(x_E | \mu_1, \sigma_1^2) \Big) \end{aligned}$$

where $\phi(x|\mu, \sigma^2) = (2\pi\sigma^2)^{-\frac{1}{2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ is the pdf of a normally distributed random variable with mean μ and standard deviation σ. Substituting this expression in ℓ_P and maximising w.r.t. θ yields our estimator $\hat{\theta}_P$. We can then classify new examples using the probabilistic conditional $P(Y|X_C, X_E, \hat{\theta}_P)$ from Eq. (12).

To reduce the number of unknown parameters fitted from very little data and to make it comparable to the three parameters of a standard logistic regression model for our setting, we assume all σ^2 and the sigmoid shape parameter s to be known and equal to one in our simulations. At each iteration, we then draw a new set of parameters μ_C, $\theta_Y = m$, $\theta_E = (\mu_0, \mu_1)$, where $\mu_C, \mu_1 \sim U(0, 1)$, $m \sim U(-1, 1)$, and $\mu_0 \sim U(-1, 0)$. Next, we generate a synthetic data set by drawing n_S labelled and n_T unlabelled samples from the source- and target domains, respectively, according to Eq. (17). One such classification data set is shown examplary in Fig. 5a. For $\hat{\theta}_S$ and $\hat{\theta}_{WS}$ we use the closed-form weighted least squares solutions as MLEs for μ_0 and μ_1, whereas all other estimates need to be found by gradient descent due to the non-linearity of the sigmoid function.

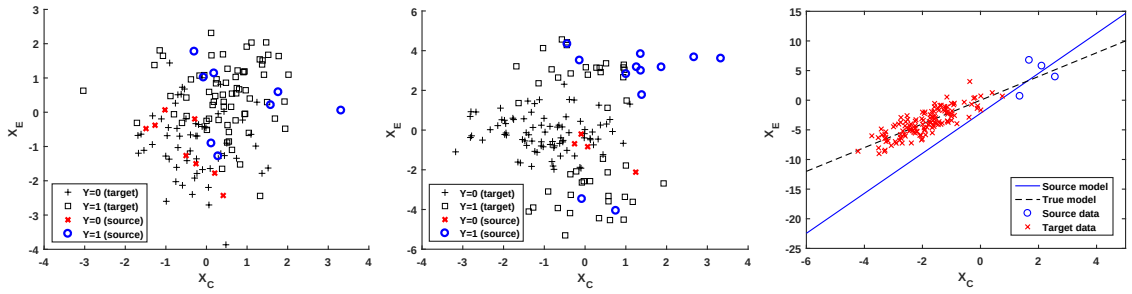
Furthermore, we perform some additional classification experiments to investigate the behaviour of our approach under model-misspecification. To achieve this, we fit exactly the same model as before (i.e., a linear decision boundary) while changing one of the normal distributions into a mixture of Gaussians (MoG). Specifically, we set $\mu_0 = 0$ and $\mu_1 = 3$ to ensure strong non-linearity and then draw the class-1 effects according to

$$X_E \,|\, (Y = 1) \sim \frac{1}{2}\mathcal{N}(-\mu_1, 1) + \frac{1}{2}\mathcal{N}(\mu_1, 1).$$

An example of such a classification data set generated with a MoG is shown in Fig. 5b.

4.4 Synthetic Regression Experiments

For our synthetic regression experiments, we focus on a linear setting with Gaussian noise. Albeit a very simple model, linear regression is still widely used on small data sets. Morevoer, it has the advantage of high interpretability of parameter estimates and is thus still a popular choice, e.g., in social sciences. Finally, the integral in Eq. (10) required for continuous $\mathcal{Y}$ is easily computed in the linear Gaussian case (but may be tricky for more complicated



(a) Classification (b) Classification with MoG (c) Regression

Figure 5: Our synthetic data sets. Shown are classification data under normal conditions in (a), and with a mixture of Gaussians for $X_E \,|\, Y = 1$ as used for the model-misspecification experiments in (b). For both, target labels are not available during training. (c) shows the regression data along with the true model and the corresponding fit from source data.

models). We thus generate synthetic regression data according to the following linear SCM:

$$\begin{aligned} X_C &:= \begin{cases} \alpha + \epsilon_C & \text{if} \quad D = 0, \\ -\alpha + \epsilon_C & \text{if} \quad D = 1, \end{cases} \quad \text{where} \quad \epsilon_C \sim \mathcal{N}(0, \sigma_C^2) \\ Y &:= a + bX_C + \epsilon_Y, \quad \text{where} \quad \epsilon_Y \sim \mathcal{N}(0, \sigma_Y^2) \\ X_E &:= c + dY + \epsilon_E, \quad \text{where} \quad \epsilon_E \sim \mathcal{N}(0, \sigma_E^2) \end{aligned} \tag{19}$$

with all noise variables ϵ_i mutually independent. This induces the distributions

$$\begin{aligned} Y \,|\, (X_C = x_C) &\sim \mathcal{N}(a + bx_C, \sigma_Y^2) \\ X_E \,|\, (Y = y) &\sim \mathcal{N}(c + dy, \sigma_E^2). \end{aligned} \tag{20}$$

Substituting for Y in the last line of Eq. (19), we obtain

$$X_E = c + ad + bdX_C + d\epsilon_Y + \epsilon_E.$$

From a standard result about the sum of two normally distributed random variables applied to $d\epsilon_Y$ and ϵ_E, it then follows that

$$X_E \,|\, (X_C = x_C) \sim \mathcal{N}(c + ad + bdx_C, d^2\sigma_Y^2 + \sigma_E^2) \tag{21}$$

which allows us to compute and maximise ℓ_P to obtain our proposed estimator $\hat{\theta}_P$. In order to predict in the regression setting, we also need to provide a closed form solution to the argmax of the probabilistic conditional $P(Y|X_C, X_E, \hat{\theta}_P)$:

Proposition 2 *Given the regression model in Eq. (19) and a parameter estimate θ, the most likely outcome for a new observation (x_C^*, x_E^*) is given by*

$$y^* = \frac{\sigma_E^2(a + bx_C^*) + d^2\sigma_Y^2(\frac{x_E^* - c}{d})}{\sigma_E^2 + d^2\sigma_Y^2}.$$

Proof *See Appendix C.* ■

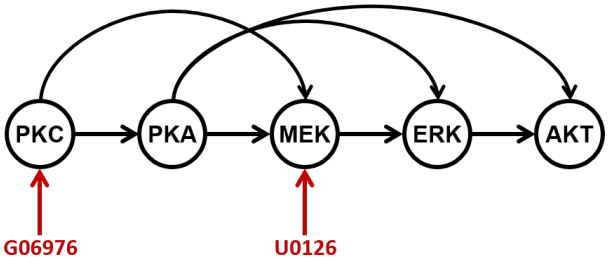


Figure 6: Shown is the subset of variables from the protein-signalling network which we used for our experiments. Arcs indicate causal links, and shown in red are interventions on PKC and MEK via the substances G06976 and U0126, respectively. For more details the reader is referred to the original paper (Sachs et al., 2005).

The prediction y^* in Proposition 2 can be seen as a weighted average of the two submodels' predictions, where inverse weights correspond to each model's uncertainty.

Simulations are then performed as follows: Firstly, all σ^2s are assumed to be known and equal to one, while the remaining parameters α, $\theta_Y = (a, b)$, and $\theta_E = (c, d)$ are drawn anew from a hyperprior at each iteration, where $\alpha \sim U(0, 2)$ and $a, b, c, d \sim U(-2, 2)$. Next, we generate a synthetic data set by drawing n_S labelled and n_T unlabelled samples from the source- and target domains, respectively, according to Eq. (19). We then compute the different estimators using the analytical (weighted) least squares solutions for θ_S, θ_{WS}, and θ_{LR}, whereas θ_P is found using gradient descent, as the unsupervised target model is non-linear in the parameters, see Eq. (21).

4.5 Real-Data Regression Experiments

As a real-world example we use the "Causal Protein-Signalling Network" data set published by Sachs et al. (2005). It contains single-cell measurements of 11 phospho-proteins and phospho-lipids under 14 different experimental conditions, as well as—importantly for our method—the corresponding causal Bayesian network inferred from this interventional data. In our experiments, we focus on a subset of variables which is shown in Fig. 6. This subset was selected to be most compatible with our assumptions.

We extract two data sets of different difficulty from this subset of variables: $\mathcal{D}_1$, which corresponds to MEK (X_C) $\rightarrow$ ERK (Y) $\rightarrow$ AKT (X_E), and $\mathcal{D}_2$, which coresponds to PKC (X_C) $\rightarrow$ PKA (Y) $\rightarrow$ AKT (X_E). For both, source data consists of measurements under normal conditions and target data is obtained by intervening on MEK in the case of $\mathcal{D}_1$ and on PKC in the case of $\mathcal{D}_2$. The two real-world data sets are shown in normal and log-log scale in Fig. 7. As can be seen, $\mathcal{D}_1$ shows a high similarity between domains, whereas $\mathcal{D}_2$ appears to be more challenging due to the high domain discrepancy.

As is often the case with biological data, features span multiple orders of magnitude and are thus more easily visualised in log-space. Moreover, all relationships between variables, i.e., protein-protein interactions, seem to be reasonably-well approximated by power laws ($Y = AX^b$) which is also often true for natural systems. In our case, we can think of protein X as either facilitating or inhibiting expression of protein Y, roughly resulting in an either linear or inverse relationship, respectively. For these reasons, we decide to first transform

the data by taking logarithms (which is not a problem as all features, being protein counts, are ≥ 1). We then fit a linear model in log-space ($\log Y = a + b \log X$) which corresponds to fitting a power law in the original space.

We then perform simulations on the real data sets as follows. At each iteration, we draw a fixed number n_S of labelled observations from the source domain, and reserve 200 observations from the target domain as a test set. From the remaining target data, we draw $n_T = 2, 4, ..., 512$ additional observations as unlabelled training data. We then fit a linear model as described in the previous Section for synthetic regression experiments, with the difference that we also treat σ_Y^2 and σ_E^2 as unknown. The model parameters are thus $\theta_Y = (a, b, \sigma_Y^2)$, and $\theta_E = (c, d, \sigma_E^2)$, with mechanisms as in Eq. (20), the unsupervised model as in Eq. (21), and predictions are made according to Proposition 2.

Finally, to investigate how background knowledge can aid our approach in real world applications, we also perform simulations on $\mathcal{D}_2$ under the constraint $b, d \leq 0$, i.e., fitting lines with negative slope. This constraint captures that both PKC $\rightarrow$ PKA and PKA $\rightarrow$ AKT are inverse relationships which might be known from domain expertise. To accommodate parameter constraints in our optimisation routine, we maximise over $\beta, \delta, s_Y, s_E \in \mathbb{R}$ setting $b = -e^{\beta}$, $d = -e^{\delta}$, $\sigma_Y^2 = e^{s_Y}$, and $\sigma_E^2 = e^{s_E}$.

4.6 Choosing λ

To choose the hyperparameter $\lambda \in (0, 1)$ we perform a grid search, considering different combinations of n_S and n_T. This is done for synthetic classification and regression data generated as described in the previous section with a fixed choice of parameters. The results are shown in Fig. 8. For classification, we find that $\lambda_r(n_S, n_T) = \frac{n_S}{n_S + n_T}$ giving equal weight to all observations (i.e., more weight to the unsupervised model as n_T is increased) appears to be a good choice across different settings, see Fig. 8a.

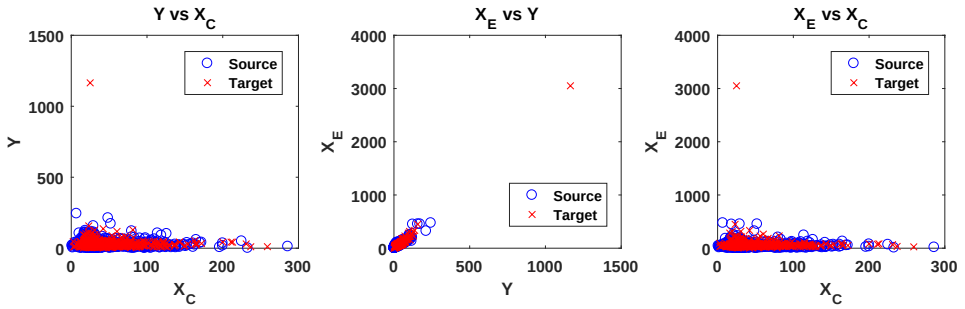
In contrast, for regression a good choice of λ does not seem to depend on n_T. Rather than weighting all observations equally, values of λ giving large weight to the supervised model appear to be preferred for regression. Following the results of Fig. 8b, we thus choose a constant $\lambda = 0.8$ for our regression experiments.

Finally, we note that as the amount of labelled data is increased, the unsupervised model seems to become obsolete for simple linear regression problems. One could thus also consider choosing $\lambda(n_S)$ to approach 1 for large n_S: e.g., a choice like $\lambda(n_S) = 1 - \frac{1}{n_S}$ could be reasonable.

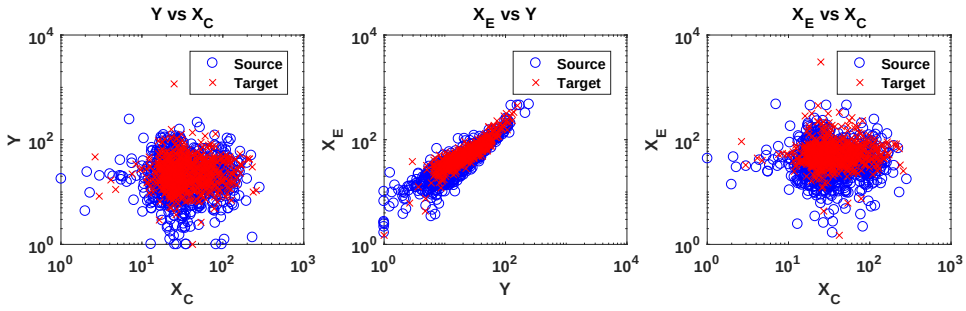
4.7 Evaluation

For both classification and regression experiments on synthetic data, we draw a test set of size 10^3 from the target domain and use it to evaluate the different estimators. The simulation steps described in Sections 4.3 and 4.4 are then repeated $n_{\text{iter}} \geq 10^3$ times, which each iteration corresponding to a different set of model parameters. This yields an average performance over many different classification/regression problems, thus avoiding overfitting to any particular parameter configuration and increasing the robustness of our findings.

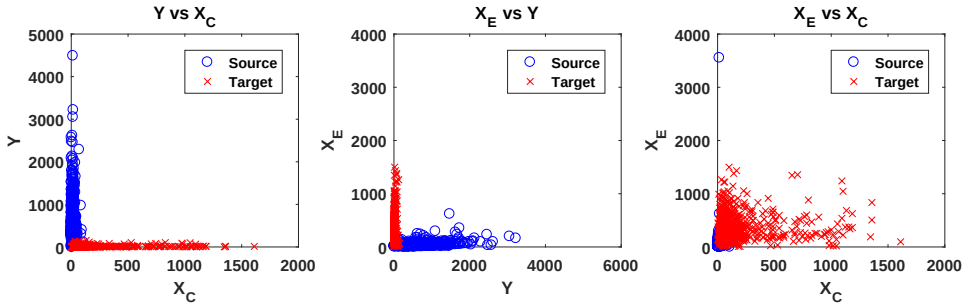
We report both negative log-likelihood (NLL) and actual loss (RMSE/error rate) test-set averages, as recommended by Loog and Jensen (2015), for the following reason: while



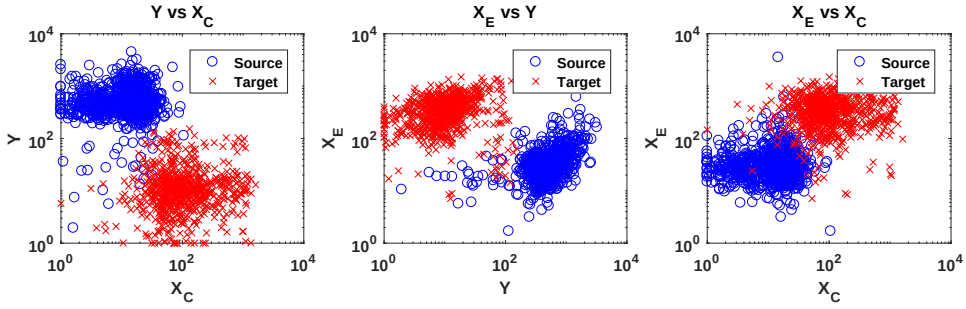
(a) $\mathcal{D}_1$ in normal scale



(b) $\mathcal{D}_1$ in log-log scale

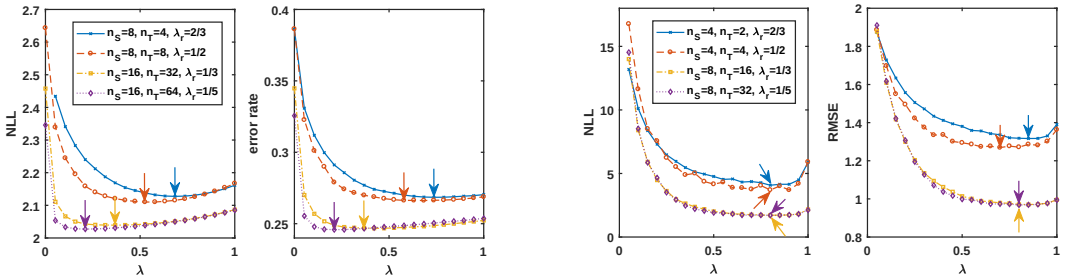


(c) $\mathcal{D}_2$ in normal scale



(d) $\mathcal{D}_2$ in log-log scale

Figure 7: Shown are the two real-world data sets used in our experiments. $\mathcal{D}_1$ corresponds to MEK(X_C) $\to$ ERK(Y) $\to$ AKT(X_E), and $\mathcal{D}_2$ to PKC(X_C) $\to$ PKA(Y) $\to$ AKT(X_E). Target data is obtained by interventions on MEK and PKC for $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively.



(a) Classification (b) Regression

Figure 8: Tuning the hyperparameter λ - Shown are negative log-likelihood and RMSE/error rate against $\lambda \in (0, 1)$ for different combinations of n_S and n_T (see legends); arrows mark the minima of each curve. All results are test set averages over 10^4 runs.

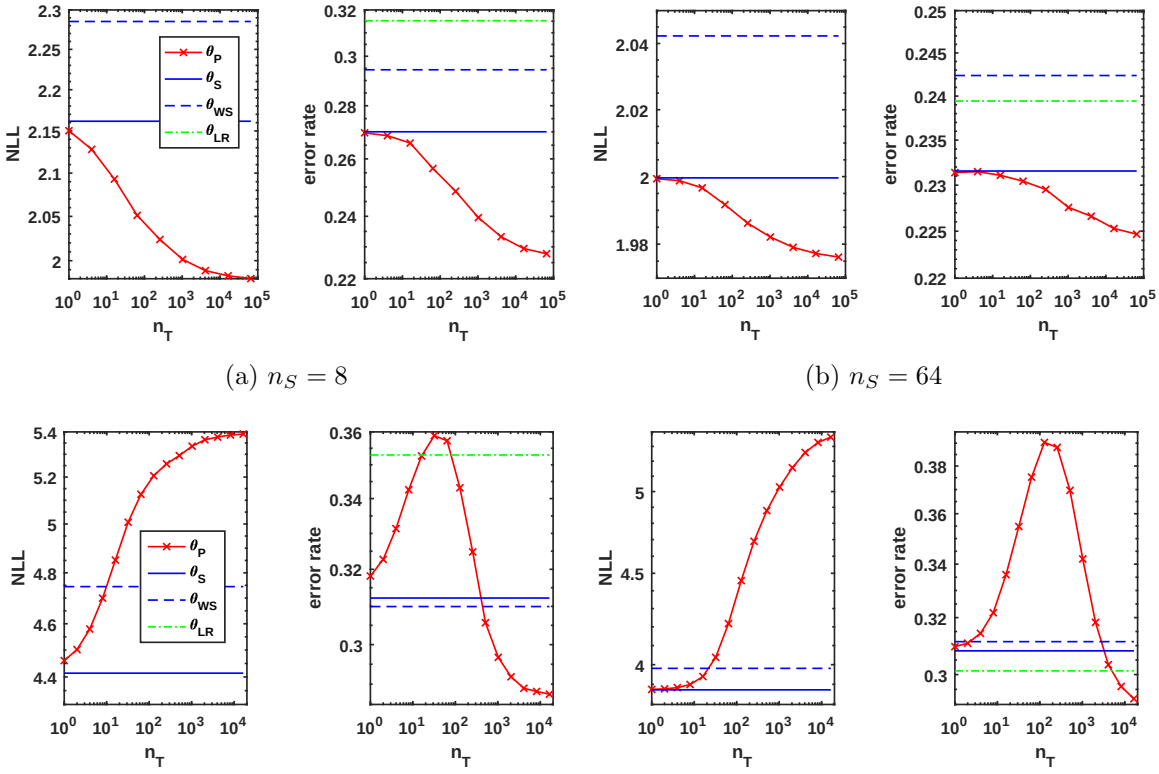
we are often mainly interested in a small RMSE or 0-1 loss, optimisation of this quantity generally suffers from non-convexity-related issues so that our model is instead trained to minimise a surrogate loss. In our case, this surrogate loss is the semi-generative, negative log-likelihood: $-\log P(Y, X_E|X_C, \theta) = -\log P(Y|X_C, \theta_Y) - \log P(X_E|Y, \theta_E)$.

5. Results

Here we present the results of our experiments described in the previous section. Unless explicitly stated, all curves show test-set averages over 10^4 simulations using $\lambda = \frac{n_S}{n_S+n_T}$ for classification and $\lambda = 0.8$ for regression. As we report averages over many different synthetic data sets, variances can be quite large, and so we omit error bars for clarity. However, applying a paired t-test to, e.g., the results from Fig. 9 indicates statistical significance with $p < 0.05$ (and much smaller p-values for large n_T), and similar results can be expected for the other plots, provided that sufficiently many simulations are performed.

Throughout, we mainly consider two settings: one with a very small amount of labelled data ($n_S = 8$ for classification, $n_S = 4$ for regression) and the other with a medium amount of labelled data ($n_S = 64$ for classification, $n_S = 16$ for regression). We then investigate the relative performance of our approach as the amount of unlabelled data (n_T) is increased. This reflects our aim of improving the source model with unlabelled data, when labelled training data is very scarce.

Note that while $n_S = 8$ (or even 4 for regression) may seem like an unrealistically small amount of labelled data, this always has to be considered relative to the dimensionality. As our synthetic (and real) data sets have two-dimensional feature space, this corresponds to ≈ 3 (or 2 for regression) observations per dimension. Due to the curse of dimensionality (i.e., the number of data required increase exponentially with the dimensionality of the problem) our setting of 2 or 3 observations per dimension, is thus quite realistic for high-dimensional problems.



(a) $n_S = 8$ (b) $n_S = 64$

(c) $n_S = 8$ (misspecified) (d) $n_S = 64$ (misspecified)

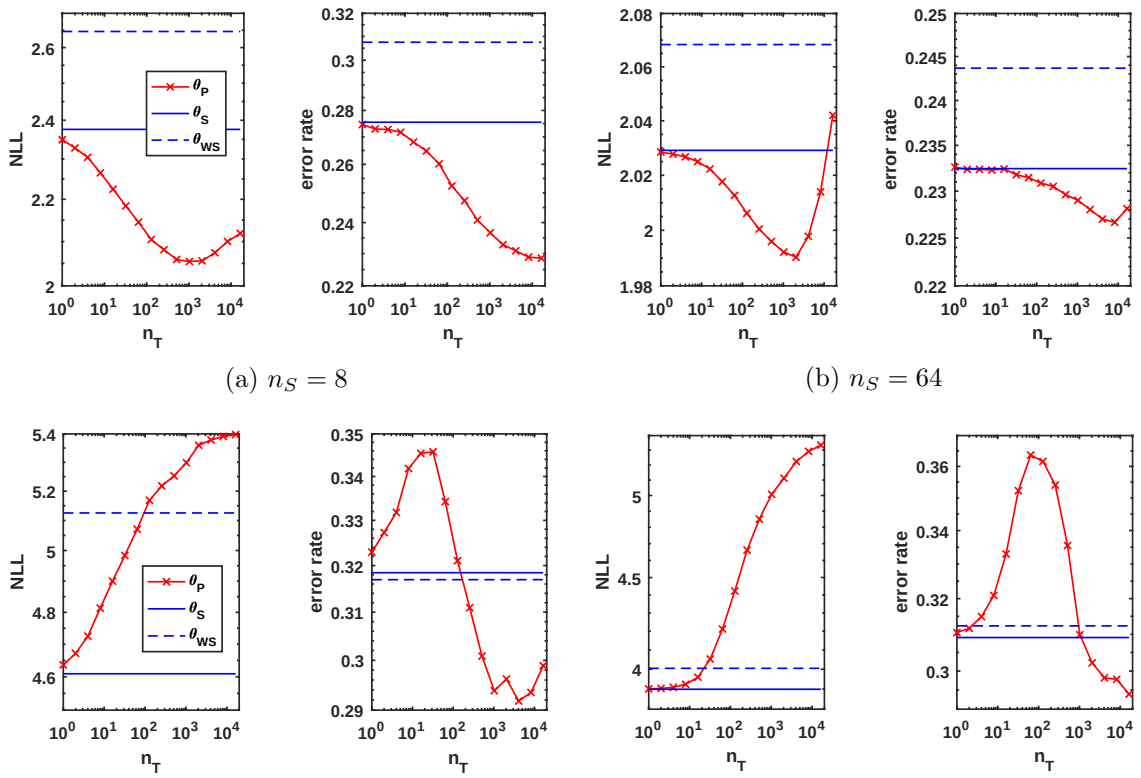
Figure 9: Classification results using maximum likelihood parameter estimates under correctly-specified (top row) and misspecified models (bottom row).

5.1 Synthetic Classification Results

Using our synthetic classification data set, we perform maximum likelihood estimation and Bayesian modelling under both correctly specified and misspecified models. Moreover, in the case of a correct model and fitting by maximum likelihood, we also compare our estimators with different feature-transformation methods followed by fitting a logistic regression model.

Results of using maximum likelihood estimation are shown in Fig. 9. For a correct model (top row), the curves of NLL and error rate look very similar and follow the same behaviour (despite the NLL curve being slightly smoother). For both small and medium n_S, the source-only baseline θ_S performs better than the importance-weighted model θ_{WS}. Our pooled-data estimator θ_P performs best, with error and NLL decreasing monotonically with n_T. This decrease is much more pronounced for $n_S = 8$ with an absolute improvement of 4% in error rate for large n_T, compared to θ_S. With $n_S = 64$ labelled examples, on the other hand, this improvement in error rate drops to only a little over 0.5%.

Under model-misspecification (bottom row), the observed behaviour is very different as the curves of NLL and error rate are no longer aligned. Specifically, while NLL is monotonically increasing with n_T, error rate increases initially, then peaks for an intermediate value of n_T, and eventually decreases again reaching the lowest value among all estimators for very large n_T. For $n_S = 8$, the peak in error rate occurs earlier and the minimum



(a) $n_S = 8$ (b) $n_S = 64$

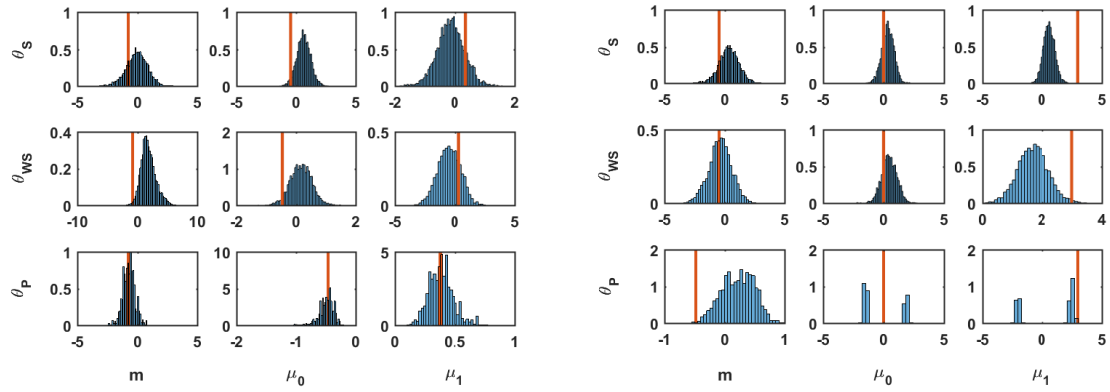
(c) $n_S = 8$ (misspecified) (d) $n_S = 64$ (misspecified)

Figure 10: Classification results using a Bayesian approach for correctly-specified (top row) and misspecified (bottom row) models. Shown are averages over 10^3 simulations.

attained over the range of n_T considered is lower than for $n_S = 64$. In the case of the latter, the structure-agnostic logistic regression model, θ_{LR}, yields the lowest error rate for $n_T < 5 \times 10^3$. θ_S and θ_{WS} show similar performance in terms of error rate both for small and medium n_S, and yield the lowest error for $n_S = 8$ and $n_T < 4 \times 10^2$ beyond which θ_P is to be preferred.

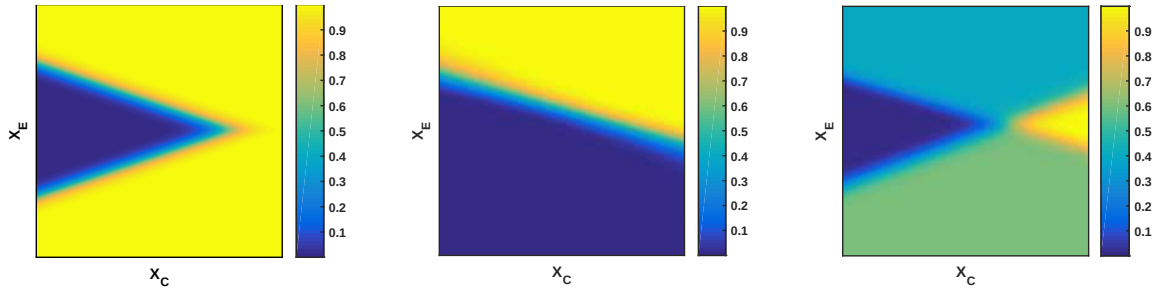
Bayesian results for the same experiments as in Fig. 9 are shown in Fig. 10. While the overall behaviour is very similar between ML- and Bayesian approaches, mainly two differences can be observed. Firstly, using a Bayesian approach under a correct model (top row), we find that some learning curves are no longer strictly decreasing, but instead reach a minimum and increase again for very large n_T. For both NLL curves this minimum occurs at about $n_T \approx 10^3$. Moreover, for $n_S = 64$, the later increase in NLL is much more pronounced and even error rate reaches a minimum before increasing again, which is not the case for $n_S = 8$.

The second observed difference using a Bayesian as opposed to a maximum likelihood approach is that the former seems to be somewhat more robust under model misspecification than the latter. While error rate initially increases with n_T in either approach, the maximum error occurs at smaller n_T in the case Bayesian modelling, and less unlabelled data is required for θ_P to return to an error lower than the source model in this case (compare



(a) correct model ($n_S = 8$, $n_T = 1024$) (b) misspecified model ($n_S = 8$, $n_T = 256$)

Figure 11: Metropolis-Hastings sampling-based approximations to the posterior distributions over classification parameters in the case of correctly- (a) and incorrectly-specified (b) models; vertical red lines indicate the corresponding true values of m, μ_0, and μ_1.



(a) ground truth (b) using θ_S (c) using θ_P

Figure 12: Visualization of the probabilistic conditional $P(Y = 1|X_C, X_E)$ under model-misspecification, and its Bayesian approximations with $n_S = 8$, $n_T = 256$, corresponding to the posteriors shown in Fig. 11b. Both X_C and X_E range from -10 to 10.

Figures 9c and 10c). Moreover, with a Bayesian approach to model misspecification and given 8 labelled observations the peak error reached is about 1.5% lower than with maximum likelihood. For $n_S = 64$ this difference rises to 2.5% in absolute error rate.

Fig. 11 shows two examples of posterior distributions over θ_S, θ_{WS}, and θ_P given labelled and unlabelled training data, along with the true parameter values marked by red lines. (Note, however, that the role of μ_1 changes between correct and incorrect models, see Section 4.3 for details.) For a correct model and given 8 labelled and 1024 unlabelled data (Fig. 11a), the posterior over θ_P, unlike those over θ_S and θ_{WS}, is approximately centred around the true parameter values. Moreover, it is more spiked as indicated by the scaling of axes. Under model-misspecification as shown in Fig. 11b, on the other hand, the posterior over θ_P appears to be bimodal with respect to μ_0 and μ_1, whereas posteriors over θ_S and θ_{WS} seem to remain unimodal.

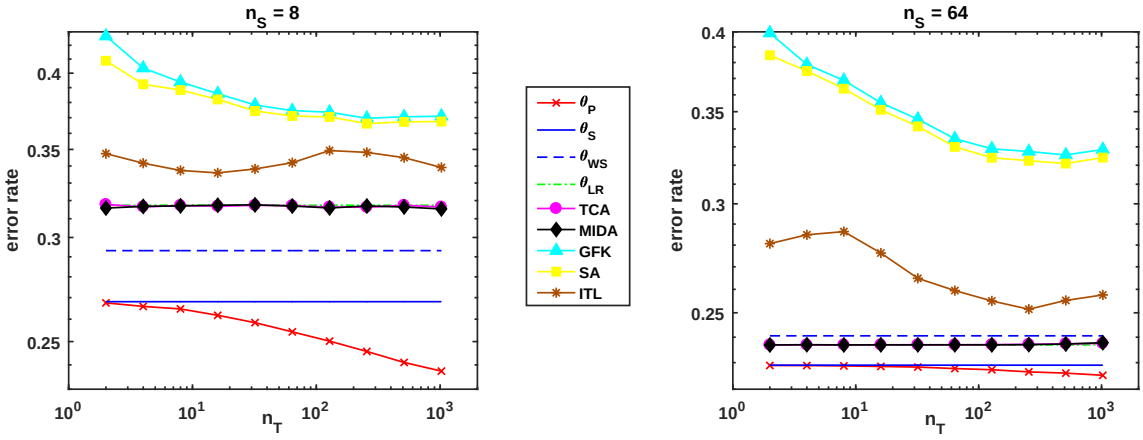


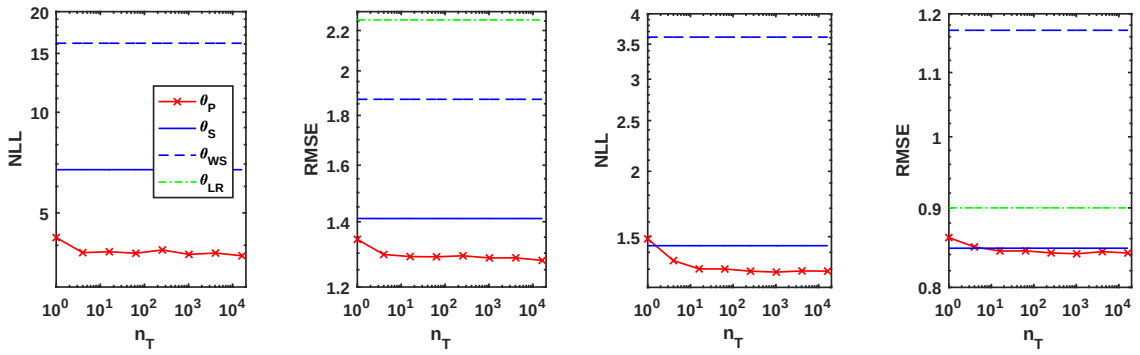
Figure 13: Comparison of our pooled ML estimator with feature-transformation DA methods on synthetic classification data sets with 8 (left) and 64 (right) labelled training examples. Results of TCA and θ_{LR} mostly coincide with MIDA and are thus hard to see.

The decision surfaces, $P(Y = 1 \,|\, X_C, X_E)$, resulting from the different posteriors in Fig. 11b are shown in Fig. 12. It also contains the ground truth as used for generating the data in Fig. 5b. As can be seen, the true decision boundary, $P(Y = 1|X_C, X_E) = 0.5$, is formed by two straight lines separating the"$Y = 0$-cluster" from the Gaussian mixture for $Y = 1$. The decision boundary found using θ_S corresponds to one of these linear segments, whereas that found using θ_P is more differentiated. It appears to be the average of both linear segments taken individually, resulting in class probabilities close to 0.5 over a wide range of (X_C, X_E). This observation is consistent with the bimodal posterior over μ_0 and μ_1 found for θ_P in Fig. 11b, with each mode corresponding to one of the two linear boundaries.

Finally, a comparison of our pooled-data maximum likelihood estimator with logistic regression models trained after different feature transformations is shown in Fig. 13. For 8 labelled examples (left), estimators based on the submodels $P(Y|X_C)$ and $P(X_E|Y)$ (θ_S, θ_{WS}, θ_P) outperform the remaining ones, which train one model on the joint feature set; for $n_S = 64$, this holds only for θ_S, and θ_P. Across both settings, TCA and MIDA lead to almost identical results as θ_{LR}, while ITL, SA, and GFK yield the highest errors. Our pooled estimator θ_P is the only one which is strictly improving with more unlabelled data throughout; however, this improvement is very slim in the case of $n_S = 64$.

5.2 Synthetic Regression Results

On the synthetic regression data set we perform maximum likelihood estimation and investigate the effect of the interpolating hyperparameter λ further. Figure 14 shows results for using $\lambda = 0.8$ given 4 (a) or 16 (b) labelled training points. As in Fig. 9 (top) with error rate, also for regression with a correct model the surrogate loss curve aligns well with the RMSE. For both values of n_S, θ_S performs better than θ_{WS}. The pooled estimator θ_P achieves the lowest RMSE overall and quickly improves upon θ_S given only a few unlabelled observations. However, adding more unlabelled data beyond $n_T > 10$ does not lead to any noticeable reduction in RMSE. In general, gains are more pronounced for $n_S = 4$ when θ_P



(a) $n_S = 4$ (b) $n_S = 16$

Figure 14: Regression results using maximum likelihood estimates with $\lambda = 0.8$.

results in a reduction of about 0.1 in RMSE compared to θ_S. For $n_S = 16$, on the other hand, this gain is at least an order of magnitude smaller and thus negligible.

The same experiment as is Fig. 14a is repeated in Fig. 15, but this time with $\lambda = \frac{n_S}{n_S+n_T}$ (as used for classification). Learning curves of θ_P (top row) for both NLL and RMSE decrease initially, quickly reach a minimum at $n_T = 2$, and then rise again leading to increasingly worsened performance as more unlabelled data is added. For the minimum at $n_T = 2$ and another point at $n_T = 128$ (black arrows), example model fits (lines) are shown in the middle and bottom rows, respectively. With 2 unlabelled points (middle row), all estimators fit the submodel $X_E \,|\, Y$ well, but the line fits of θ_P to $X_E \,|\, X_C$ and $Y \,|\, X_C$ are closer to the true model than those of θ_S and θ_{WS}. For $n_T = 128$ (bottom row), all estimators result in a decent fit to $X_E \,|\, X_C$ with θ_P almost coinciding with the true model. The two submodels $Y \,|\, X_C$ and $X_E \,|\, Y$, however, are fitted very poorly by θ_P, as opposed to θ_S and θ_{WS}.

5.3 Real-Data Regression Results

On the real-world data sets, we compare the performance of our approach using maximum likelihood estimation with linear regression models trained after different feature transformations. Results from using $\lambda = 0.8$ on $\mathcal{D}_1$ and $\mathcal{D}_2$ are shown in Fig. 16. As can be seen, RMSEs of all methods are much lower on $\mathcal{D}_1$ than on $\mathcal{D}_2$ supporting our claim that $\mathcal{D}_2$ is the more challenging one.

On $\mathcal{D}_1$ with $n_S = 4$ (a), both θ_S and θ_P outperform the structure-agnostic feature transformation methods; θ_P reaches the lowest RMSE of a little under 0.6—an absolute improvement of 0.1 over θ_S with an RMSE of 0.7. For small n_T, MIDA outperforms TCA, GFK, and SA but for large n_T all feature tranformation methods converge in RMSE to the linear regression model θ_{LR} (coinciding with TCA). With 16 labelled examples (b), on the other hand, all feature transformation methods except GFK outperform θ_S and θ_P, with MIDA achieving the lowest RMSE, followed by TCA and θ_{LR}. SA and GFK lead to the highest errors on $\mathcal{D}_1$, especially for small n_T, but both are strictly improving as more unlabelled data becomes available.

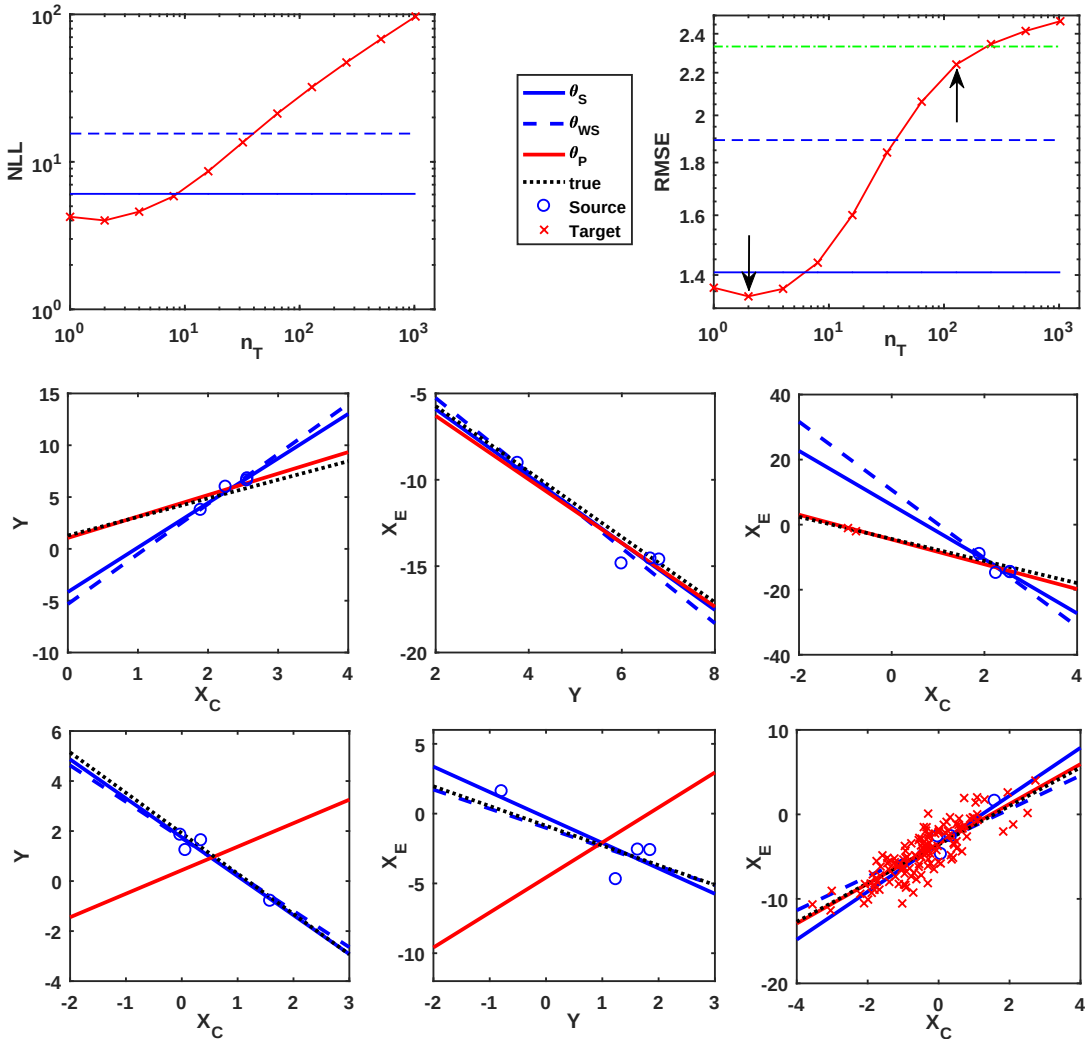
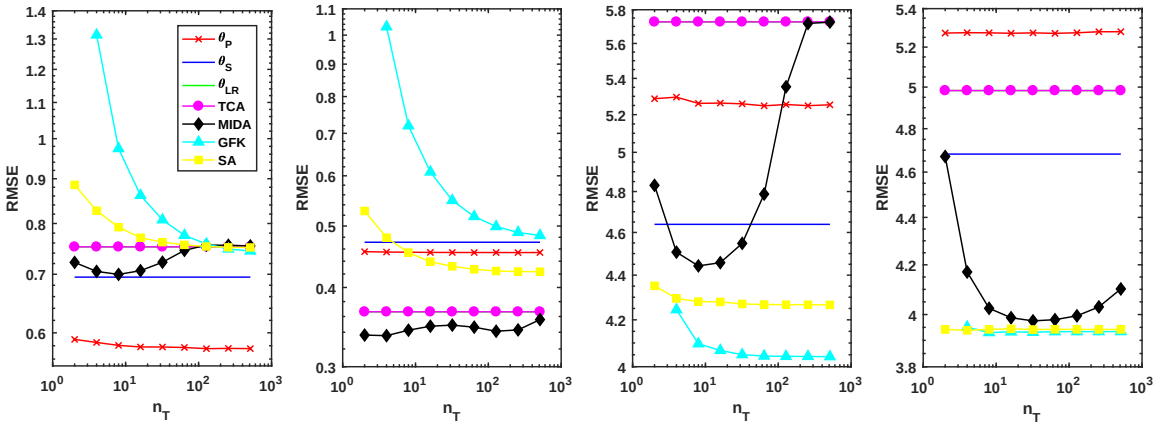


Figure 15: ML regression results with $n_S = 4$ and $\lambda = \frac{n_S}{n_S+n_T}$, showing learning curves of NLL and RMSE vs n_T (top). Arrows mark $n_T = 2$ with $\lambda = \frac{2}{3}$, and $n_T = 128$ with $\lambda \approx 0.03$ for which example model fits are shown in the middle and bottom rows, respectively.



(a) $\mathcal{D}_1$: $n_S = 4$ (b) $\mathcal{D}_1$: $n_S = 16$ (c) $\mathcal{D}_2$: $n_S = 4$ (d) $\mathcal{D}_2$: $n_S = 16$

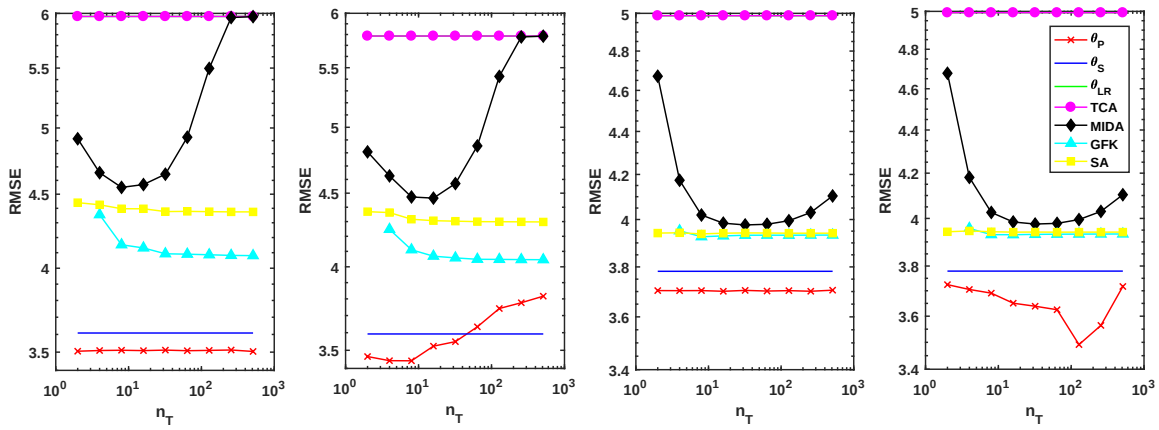
Figure 16: Regression learning curves for the real data sets $\mathcal{D}_1$ and $\mathcal{D}_2$ with 4 and 16 labelled source observations. Throughout, we used $\lambda = 0.8$ and fitted an (unrestricted) linear model in log-space, see Fig. 7. θ_{LR} is not visible as its curve coincides with that of TCA here.

On $\mathcal{D}_2$, GFK, despite leading to the worst results on $\mathcal{D}_1$, achieves the lowest error for both settings, closely followed by SA. As on $\mathcal{D}_1$, both show a monotonic decrease in RMSE with respect to n_T, which is not the case for MIDA on either of the data sets. Both θ_S and our pooled estimator θ_P show poor performance on $\mathcal{D}_2$, and, as opposed to the other methods, do not experience improvements in RMSE as n_S is increased from 4 to 16.

In the case of $\mathcal{D}_2$, we also analyse the effect of additional model restrictions in Fig. 17, considering both $\lambda = 0.8$ and $\lambda = \frac{n_S}{n_S+n_T}$. Restricting the two linear submodels used in θ_S and θ_P to have negative slope (capturing the inverse relationship between the involved proteins) leads to considerably lower RMSEs. Unlike their unrestricted counterparts in Fig. 16, the restricted versions of θ_S and θ_P clearly outperform the feature tranformation methods to which no restrictions were applied (to be discussed further later on).

For $\lambda = 0.8$, θ_P consistently performs better than θ_S by an absolute difference in RMSE of roughly 0.1. However, this difference can be achieved using as little as 2 unlabelled data points beyond which learning curves for θ_P seem to remain constant; this can also be observed in Fig. 16, where $\lambda = 0.8$ was used throughout.

For $\lambda = \frac{n_S}{n_S+n_T}$, on the other hand, learning curves of θ_P show more complex behaviour. For both 4 (b) and 16 (d) labelled training examples, RMSE initially decreases, then reaches a minimum, and eventually increases again. For $n_S = 4$, though, this minimum occurs much sooner so that an initial improvement compared to using $\lambda = 0.8$ as in (a) is almost not noticeable; for large n_T the learning curve in (b) even crosses that of θ_S attaining a higher maximum RMSE. With $n_S = 16$ (d), the minimum occurs much later leading to considerable improvements compared to using $\lambda = 0.8$ as in (c).



(a) $n_S = 4$, $\lambda = 0.8$ (b) $n_S = 4$, $\lambda = \frac{n_S}{n_S+n_T}$ (c) $n_S = 16$, $\lambda = 0.8$ (d) $n_S = 16$, $\lambda = \frac{n_S}{n_S+n_T}$

Figure 17: Regression learning curves for the more challenging data $\mathcal{D}_2$ using a restricted linear model with negative slope for θ_S and θ_P. In (a) and (c) λ is kept constant, whereas in (b) and (d) all examples carry equal weight leading to λ decreasing with n_T.

6. Discussion

Before discussing and interpreting our empirical results, first, let us recall our goal, which was improving the (adapted) source model with unlabelled data when the amount of labelled data is the main bottleneck. In general, it seems that this goal was met to a larger extent in our classification experiments than in the regression ones. Given the correct model, our approach led to significant and consistent improvements with more unlabelled data in the case of classification (see, e.g., Figures 9a or 10a); in the case of regression, however, improvements were generally smaller and stagnated after only a few unlabelled examples (see Figures 14 and 16). This is, of course, also linked to different choices of λ used for classification and regression, as we shall discuss in more detail. Nevertheless, our results point to a more general phenomenon, as illustrated in Fig. 15.

6.1 Model Flexibility and the Role of λ

A first possible explanation is linked to model flexibility—a topic which we consider of central importance to understanding the strengths and weaknesses of our method. Probably the most obvious difference in our approach between classification and regression problems is how the unsupervised model $P(X_E \,|\, X_C)$ is obtained. For classification, we sum over a finite number of possible classes, whereas for regression we integrate over all possible values of Y, see Eq. (10). It thus seems as though the unsupervised classification model is more strongly constrained, or less flexible, than the unsupervised regression model: given an unlabelled observation (x_C, x_E) from a classification problem, there are only finitely many (in the case of binary classification only two) possible outcomes y which x_C could have caused, and which then in turn resulted in effect x_E; in a regression problem, on the other hand, the number of possible outcomes y is infinite, and many of them could potentially explain the observed effect x_E reasonably well.

We hypothesize that this higher flexibility of the unsupervised model is one of the factors which make our approach somewhat less suitable for regression than for classification problems. Some support for this claim can be found in Fig. 15, where using $\lambda = \frac{n_S}{n_S+n_T}$ for regression given 4 labelled examples results in much higher RMSE when more than a few unlabelled training points are included. The problem in this case is apparent from the bottom row, which shows that our estimator θ_P perfectly fits the unsupervised model $X_E \,|\, X_C$, but at the cost of completely mismatching the two mechanisms $Y \,|\, X_C$ and $X_E \,|\, Y$. Recalling our regression models, $Y = a + bX_C$, $X_E = c + dY$, and so $X_E = bdX_C + \text{const.}$, it is clear that the positive slope of the unsupervised model, $bd > 0$, can be explained by either $b, d < 0$, which is the true model, or by $b, d > 0$, which is the model found by θ_P. With this flexibility, it thus only seems logical that overfitting of $X_E \,|\, X_C$ occurs eventually when equal weight is given to labelled and unlabelled observations, i.e., using $\lambda = \frac{n_S}{n_S+n_T}$. Being aware that this issue exists, it is quite striking that such overfitting is not observed for classification, not even in extreme cases such as $n_T > 10^4 \times n_S$ when essentially all weight is given to unlabelled data.

It is not completely clear whether such overfitting is a problem of our method for regression in general, or whether it is an artefact of the linear Gaussian model we use, which is known to be a very rich model class (Peters et al., 2017). Since such simple models are required to compute the unsupervised model in closed form, however, the general applicability of the semi-generative framework to regression problems remains less clear. Yet, by choosing a constant λ it can be ensured that the labelled fraction always carries a certain weight thus apparently avoiding overfitting to $X_E \,|\, X_C$, see Fig. 14. This choice can lead to a small but notable drop in error (see for example Figures 16a, 16b, 17a, or 17c), but can also backfire on tough data sets such as $\mathcal{D}_2$ (see Figures 16c and 16d). Either way, not much seems to be gained from adding more unlabelled data beyond the first few unlabelled examples when using a constant λ, which makes sense as new unlabelled data only contributes towards the unsupervised model *average* in this case. In order to obtain continual improvements as n_T increases, it thus seems necessary to choose λ to decrease with n_T. This is demonstrated by Fig. 17, where we fitted restricted linear models with negative slopes and used varying λ. As can be seen from Figures 17c and 17d, using equal weights led to clear improvements over using a constant λ up until $n_T > 128$, when overfitting of the unsupervised model appears to occur and RMSE starts rising again. For $n_S = 4$ when $\lambda = \frac{n_S}{n_S+n_T}$ goes down much faster with n_T (roughly by a factor 4), this rise in RMSE occurs much earlier.

All of the above suggests that, for regression, λ needs to be chosen with great care. While for constant λ no long-term gain from collecting more unlabelled data is observed, allowing λ to become too small as n_T is increased can lead to severe overfitting and worsened performance—especially if the underlying model class is too flexible. One possibility to avoid λ becoming too small would be to ensure a certain fraction of total weight, e.g. 20%, is always given to labelled data, so that $\lambda \to 0.2$ as $n_T \to \infty$, e.g. $\lambda = 0.2 + \frac{0.8}{1+n_T}$. Another possibility would be to have λ decrease as $\frac{1}{\log n_T}$ instead. Anyway, more theoretical work will be necessary to justify suitable forms of λ beyond the intuition and heuristics provided here. For classification, to the contrary, choosing equal weights appears to be a good choice. With $\lambda = \frac{n_S}{n_S+n_T}$ significant gains can be observed, provided the model is correctly specified.

6.2 Behaviour Under Model Misspecification

Another interesting point of discussion is the behaviour of the semi-generative approach under model misspecification. From a quantitative point of view, our experiments show very clearly that including unlabelled data can hurt performance when the underlying model is misspecified. This is not very surprising, however, as such deterioration under misspecification can occur even when adding more *labelled* data (Loog and Duin, 2012), and is well reported for semi-supervised learning (Yang and Priebe, 2011). With this in mind, it thus seems rather promising that in all our misspecification experiments performance recovered after initial deterioration, and the initial supervised guess was even improved upon, when sufficient unlabelled data was available. What constitutes a "sufficient" amount in this case seems to depend on n_S: the more labelled data is given, the more unlabelled data is required to improve, and the smaller the expected gains. For the setting of n_S very small and n_T large (for which our method is intended), there thus seems to be reason to believe that our method will not suffer much more from misspecification than the supervised baseline. For n_T very large we can even hope to improve our model with unlabelled data—despite of misspecification.

Note also, that importance-weighting, while yielding clearly inferior results to θ_S under a correct model, leads to similar error as the supervised baseline when the model is misspecified. This is consistent with current belief that such reweighting is unnecessary for correct models, but can be beneficial for misspecified ones (Quionero-Candela et al., 2009).

Further, our results suggest that, when the underlying model is misspecified, a Bayesian semi-generative approach is to be preferred to fitting parameters by maximum likelihood. While this is primarily meant quantitatively as described in Section 5, we argue that the Bayesian approach also yields qualitatively interesting results. Here, we are referring to the conditional probability surface resulting from our pooled estimator, see Fig. 12. Even though the decision boundaries, $P(Y = 1 \mid X_C, X_E) = 0.5$, in Figures 12b and 12c are both linear and lead to approximately the same error, the true distribution is arguably better captured by our estimator. While Fig. 12b looks as though a linear boundary is a good model, Fig. 12c raises some serious doubts about this and hints at the fact that the true model might instead be non-linear. This is most likely a consequence of the multi-modal posterior over θ_P which leads to a weighted average over multiple different linear models. Because of its potential to detect model misspecification from lots of unlabelled data, our approach thus also has potential applications in model diagnostics.

Moreover, it generally seems desirable for a probabilistic model to be certain about instances which are correctly classified and uncertain about those which are not. This helps to separate learned concepts from knowledge gaps and can therefore make future learning more efficient. In this regard, our semi-generative Bayesian model achieves better such uncertainty estimates via additional unlabelled data—even if the resulting error does not necessarily reflect this due to a misspecified model. This property of our approach may be useful for applications where the model is allowed to abstain from making predictions, or when additional labelled data can be queried as in active learning.

Finally, we note that our approach—at least for classification tasks, when the unsupervised model can always be obtain by summation—is model-free, i.e., does not depend on any particular choice of $P(Y \mid X_C)$ and $P(X_E \mid Y)$. When no background-knowledge is available

to reason about model choices, or when misspecification is assumed to be a problem, it is therefore always possible to choose a non-parametric approach instead. For example, placing kernels at the labelled data points, unlabelled data could be used to improve estimates of kernel coefficients or to learn hyperparameters. However, since non-parametric models are by definition flexible, care should be taken with this approach to avoid the before-mentioned overfitting problems related to too flexible models.

6.3 Comparison With Feature Transformation Methods

Comparing our approach with some feature transformation methods first of all showed that there is not a single best transform which outperforms all others across different data sets. Moreover, we found that more unlabelled data was not always beneficial in these approaches: while error curves of our approach, GFK, and SA are mostly decreasing or constant with n_T, ITL and MIDA show strongly non-monotonic behaviour.

Furthermore, we try to identify certain situations in which our approach is to be preferred to feature transformation methods, and vice-versa. From our experiments, it seems that the biggest advantage of semi-generative modelling over feature transformations occurs when (i) n_S truly is very small, thus posing the main limitation; (ii) there is sufficient overlap between domains for the unsupervised constraint to be useful; and (iii) the true model is known or sufficiently constrained. These criteria were fulfilled in the classification experiment (Fig. 13, left) and on $\mathcal{D}_1$ (Fig. 16a), but not on $\mathcal{D}_2$ (Fig. 16c). Recall that the domain shift is much smaller on $\mathcal{D}_1$, and that the true $X_E \,|\, Y$ is more easily estimated from a few labelled examples than on $\mathcal{D}_2$, see Fig. 7. If, on the other hand, sufficient labelled data is available or a large discrepancy between domains is the main complication, using unlabelled data to find a good feature representation seems to be preferred.

It is also important to point out that the comparisons with other DA techniques are, in some sense, not completely fair. This is because θ_S, θ_{WS}, and θ_P make use of additional knowledge about the causal structure, which the other methods do not. Precisely, our estimators train a semi-generative model $P(Y \,|\, X_C)P(X_E \,|\, Y)$ as the underlying causal structure is known to be $D \to X_C \to Y \to X_E$, while the other methods rely on the more general CS assumption $D \to X \to Y$ and train a discriminative linear (or logistic) regression model $P(Y \,|\, \phi(X))$ on transformed features $\phi(X)$. Both approaches result in a linear model (regression) or decision boundary (classification) and are thus, in principle, comparable, but the semi-generative approach is more modular.

While it could be desirable to combine the benefits of both approaches, it is not completely clear how additional knowledge like conditional independence statements can be included in feature transformation methods. A first idea could be to only apply a transformation on causal features (as only these are directly affected by the domain change in our setting), and then train a model $P(Y \,|\, \phi(X_C))P(X_E \,|\, Y)$. However, such a transformation would only be applied after the data was generated, meaning that domain changes will have already propagated through the mechanisms $Y \,|\, X_C$ and $X_E \,|\, Y$ to also affect X_E. More importantly, transforming X_C, but not X_E will likely impair the domain independence of the unsupervised map $X_E \,|\, X_C$ which is central for the semi-generative approach.

Another idea would be to apply the feature transformation as usual, but to then train a semi-generative model on the transformed features, instead of a discriminative one. The

problem with this, however, is that applying a joint feature transform to (X_C, X_E) may introduce dependencies between the transformed features, so that our assumptions are no longer satisfied. Additionally, it would no longer be clear which of the new features would correspond to causes and effects.

This observation points to another potential benefit of our method: since features are kept as they are, it is possible to include domain and prior knowledge in the modelling process, e.g., through priors or restrictions in the model class. When transformations—especially non-linear ones—are applied over the joint feature set, it is no longer clear how to do this. This point is illustrated by the results in Fig. 17, where restricting the linear submodels to have negative slope (capturing background knowledge about the underlying proteins' interactions) led to considerable improvements when using our estimators. Transformed features, on the other hand, no longer have a clear interpretation—in this case they would represent (non-)linear combinations of protein counts—and so integrating background knowledge is considerably harder.

6.4 General Applicability

Finally, we wish to reflect on the general applicability of our semi-generative approach. To do so, we recall that throughout this work we have assumed an underlying causal structure of the form $D \to X_C \to Y \to X_E$, see Section 3.1. It should thus be clear, that our derivations and ideas may no longer be valid when these assumptions are violated. Consequently, our semi-generative framework should not be applied as a blackbox to domain adaptation problems in general, but rather is specifically tailored to situations where expert/domain knowledge, interventional data, or causal inference can be used to identify the causal structure (or at leasts parts thereof).

Hypothetically, allowing either the set of cause or effect features to be empty can illustrate how our semi-generative model, $P(Y, X_E \,|\, X_C)$, interpolates between adaptation and SSL. If X_E is empty, we recover standard domain shift ($D \to X_C \to Y$) and our model reduces to a discriminative one, $P(Y \,|\, X_C)$. If X_C is empty ($Y \to X_E$), on the other hand, we recover a generative-modelling approach to semi-supervised learning, $P(Y, X_E)$. To combine these two, however, we stress that both cause and effect features are necessary.

Moreover, a direct consequence of Assumption 1 is that in our setting cause and effect features are conditionally independent given the label: $X_C \perp\!\!\!\perp X_E \,|\, Y$. This is of central importance for our idea of improving the supervised model by learning an unsupervised map from causes to effects, because this conditional independence creates a bottleneck at Y. Since X_C and X_E depend on each other only through Y, we can hope to improve our models of both $Y \,|\, X_C$ and $X_E \,|\, X_C$ by learning to predict X_E from X_C. This idea would probably not work if X_C had a direct influence on X_E. In this case, however, one could attempt to learn and correct for such direct influence to restore the required conditional independence.

It is also worth noting, that the same conditional independence of features given label is also assumed in co-training (Blum and Mitchell, 1998), which considers the setting of two complementary feature views X_1 and X_2. Using the two mechanisms $P(Y \,|\, X_C)$ and $P(X_E \,|\, Y)$ to construct weak classifiers h_C and h_E, and additionally assuming that each feature on its own is sufficient for learning (as in co-training), it should thus be possible

to directly apply their PAC-learning results to our setting when no domain shift occurs. Extending such PAC-learning results to the domain adaptation case, however, is likely not possible without further strong restrictions on the discrepancy between domains (Ben-David et al., 2007, 2010). Furthermore, Krogel and Scheffer (2004) have shown that, in fact, conditional independence of features given label is necessary for co-training to succeed, and that performance indeed deteriorates when features are correlated beyond a certain degree. Combined with our findings, this suggests that such a conditional independence statement may be a more general requirement for learning from unlabelled data with multiple feature views.

Apart from not allowing a direct link between cause and effect features, the other part of our setting which may be seen as a restriction is not allowing the domain shift to directly influence effect features, $D \not\to X_E$. As explained in Section 2.2 and illustrated in Appendix A.1, this is necessary to ensure that CS holds and to guarantee invariance of the unsupervised model $X_E \,|\, X_C$. However, it might be possible to relax this assumption by using what could be an interesting combination of feature transformation methods and our approach. The idea would be to learn a feature transformation ϕ applied to X_E which maximises domain invariance of $\phi(X_E) \,|\, X_C$—similar to how other CS feature-transform methods are trained to maximise domain invariance of $\phi(X)$—and thereby correcting for the shift in X_E due to D. The point would be to make the unsupervised model domain-invariant, so that unlabelled data can again be included as suggested by our approach. This idea is inspired by the work of Zhang et al. (2013) on correcting location-scale transformations in generalised target shift.

Another issue which is not directly addressed by our framework, but which is often encountered in practice, is that of (hidden) confounders. As such confounding features usually introduce further dependencies between variables, we can expect our method to work less well in the presence of confounders. However, it should be noted that in our real-world data sets such confounders are present. On $\mathcal{D}_1$, for example, PKA influences all of X_C (MEK), Y (ERK), and X_E (AKT). That our approach still yields good results on $\mathcal{D}_1$ though (see Fig. 16a) gives hope that it is at least somewhat robust to such hidden confounders.

Finally, we wish to point out that our semi-generative model provides a general approach to SSL with cause and effect features which can also be applied to purely observational data, i.e., when no domain change occurs. It is then natural to ask whether we can give any guarantees that the semi-supervised solution will actually improve over—or at least not get worse than—the supervised one on the entire training set, when performing such SSL with cause and effect features. Recent work on such safe, pessimistic SSL has shown that for many popular surrogate losses such guarantees are impossible (Krijthe and Loog, 2016). However, in terms of the likelihood this is, in principle, possible, and for particular classifiers improvement can be shown to occur almost surely (Loog, 2016). As the semi-generative likelihood optimised by our approach seems to be a good performance indicator when our model is correct, such guarantees would thus be quite reassuring. Since our problem structure ($X_C \to Y \to X_E$) gives rise to a more complex likelihood than the generative models ($Y \to X$) considered by Loog (2016), however, it is unclear whether, and if so for which classifiers, safe pessimistic SSL is possible with a semi-generative model. Yet, even without worst-case guarantees, improvements can still occur in expectation.

7. Conclusions

We have developed a novel framework for semi-supervised learning under changes in the distribution of input (or causal) features resulting from domain shift. This approach is suitable for domain adaptation problems for which the causal structure is known and for which the amount of labelled data is the main limitation, so that benefits from improving the supervised model outweigh the adaptation task. Our semi-generative model $P(Y, X_E \mid X_C)$ for learning with cause and effect features is inspired by recent work on independent mechanisms, and does not require self-training or expectation maximisation. Rather it directly imposes constraints on the supervised model by learning a map from causes to effects from unlabelled data. Our analysis and empirical results suggest that this unsupervised model needs to be sufficiently constrained in order to improve estimates of the mechanisms $P(Y \mid X_C)$ and $P(X_E \mid Y)$ without supervision. For too flexible models, on the other hand, overfitting of unlabelled data can result in deteriorated performance. Our approach thus works best for classification where we have demonstrated significant improvements over the supervised baseline, importance-reweighting, and feature-transformation methods on synthetic data sets when only very few labelled training examples are available. Moreover, on a domain-adaptation regression problem for protein signalling networks we have shown that our method can be beneficial also on real-world data, even when the underlying causal assumptions are no longer strictly satisfied. Especially in this context, it is useful that background and expert knowledge can easily be integrated in our modelling framework—something which is not straight-forward when using a feature tranformations approach to domain adaptation.

Acknowledgments

First and foremost, I would like to express my gratitude to my supervisor Marco Loog for his support and guidance throughout this project, and for asking the right questions and making me a better scientist. A big thanks also goes to my second supervisor Alexander Mey for his continued interest in and advice on the project. Moreover, I would like to thank the Empirical Inference Group at the Max-Planck Institute for Intelligent Systems in Tübingen, and Isabel Valera in particular, for interesting ideas and insightful discussions. Finally, thanks to Michele Tonutti for proofreading of- and helpful feedback on the paper.

References

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in neural information processing systems*, pages 137–144, 2007.

Shai Ben-David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *International Conference on Artificial Intelligence and Statistics*, pages 129–136, 2010.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Advances in neural information processing systems*, pages 442–450, 2010.

Gregory Druck, Chris Pal, Andrew McCallum, and Xiaojin Zhu. Semi-supervised classification with hybrid generative/discriminative methods. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 280–289. ACM, 2007.

Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013.

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.

Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012.

Wilfred Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

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

Zhen Jiang, Shiyong Zhang, and Jianping Zeng. A hybrid generative/discriminative method for semi-supervised classification. *Knowledge-Based Systems*, 37:137–145, 2013.

Changsung Kang and Jin Tian. A hybrid generative/discriminative bayesian classifier. In *FLAIRS Conference*, pages 562–567, 2006.

Jesse H Krijthe and Marco Loog. The pessimistic limits of margin-based losses in semi-supervised learning. *arXiv preprint arXiv:1612.08875*, 2016.

Mark-A Krogel and Tobias Scheffer. Multi-relational learning, text mining, and semi-supervised learning for functional genomics. *Machine Learning*, 57(1-2):61–81, 2004.

Jan Lemeire and Erik Dirkx. Causal models as minimal descriptions of multivariate systems, 2006.

Marco Loog. Contrastive pessimistic likelihood estimation for semi-supervised classification. *IEEE transactions on pattern analysis and machine intelligence*, 38(3):462–475, 2016.

Marco Loog and Robert PW Duin. The dipping phenomenon. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 310–317. Springer, 2012.

Marco Loog and Are Charles Jensen. Semi-supervised nearest mean classification through a constrained log-likelihood. *IEEE transactions on neural networks and learning systems*, 26(5):995–1006, 2015.

Sara Magliacane, Tom Claassen, and Joris M Mooij. Ancestral causal inference. In *Advances in Neural Information Processing Systems*, pages 4466–4474, 2016.

Sara Magliacane, Thijs van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij. Causal transfer learning. *arXiv preprint arXiv:1707.06422*, 2017.

Andrew McCallum, Chris Pal, Gregory Druck, and Xuerui Wang. Multi-conditional learning: Generative/discriminative training for clustering and classification. In *AAAI*, pages 433–439, 2006.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Andrew Y Ng and Michael I Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in neural information processing systems*, pages 841–848, 2002.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

Giambattista Parascandolo, Mateo Rojas-Carulla, Niki Kilbertus, and Bernhard Schölkopf. Learning independent causal mechanisms. *arXiv preprint arXiv:1712.00961*, 2017.

Judea Pearl. Bayesian networks: A model of self-activated memory for evidential reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society, 1985*, pages 329–334, 1985.

Judea Pearl. *Causality: Models, reasoning, and inference.* Cambridge University Press, 2000.

Judea Pearl. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

Judea Pearl and Elias Bareinboim. External validity: From do-calculus to transportability across populations. *Statistical Science*, pages 579–595, 2014.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference - Foundations and Learning Algorithms*. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge, MA, USA, 2017.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

Mateo Rojas-Carulla, Bernhard Schölkopf, Richard Turner, and Jonas Peters. Causal transfer in machine learning. *arXiv preprint arXiv:1507.05333*, 2015.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. In *29th International Conference on Machine Learning (ICML 2012)*, pages 1–8. International Machine Learning Society, 2012.

Yuan Shi and Fei Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1275–1282. Omnipress, 2012.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

Amos Storkey. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, pages 3–28, 2009.

Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8 (May):985–1005, 2007.

Ke Yan, Lu Kou, and David Zhang. Learning domain-invariant subspace using domain features and independence maximization. *IEEE transactions on cybernetics*, 2017.

Ting Yang and Carey E Priebe. The effect of model misspecification on semi-supervised classification. *IEEE transactions on pattern analysis and machine intelligence*, 33(10): 2093–2103, 2011.

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *International Conference on Machine Learning*, pages 819–827, 2013.

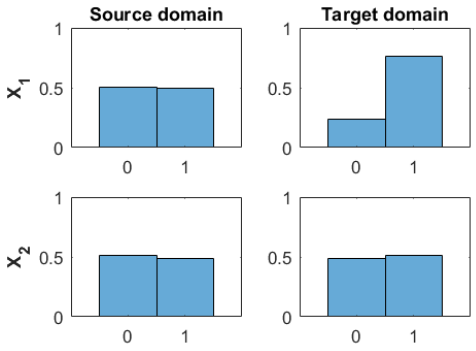
Kun Zhang, Mingming Gong, and Bernhard Schölkopf. Multi-source domain adaptation: A causal view. In *AAAI*, pages 3150–3157, 2015.

Appendix A. Causality and Domain Adaptation: Illustrative Examples

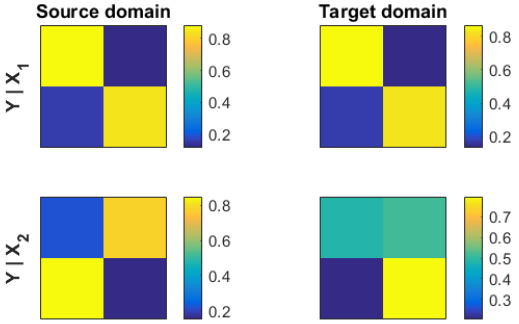
In this appendix we illustrate some of the connections between causality and DA.

A.1 Why Does Causal Structure Matter for DA?

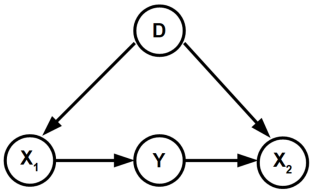
In order to illustrate the role and importance of causal structure for domain adaptation, we include a simple example below in Fig. 18. In the example, we try to predict a binary variable Y from two binary features X_1 and X_2, given labelled source data and unlabelled target data. It demonstrates the necessity for assumptions like CS by showing that features with similar marginal distributions across domains can still experience a radical shift in conditionals (X_2); conversely, features which are invariant for prediction need not show similar marginal distributions across domains (X_1). Since due to the lack of target domain labels it is not possible to simply check for which features X the conditionals are domain invariant, $P(Y|X, D = 0) = P(Y|X, D = 1)$, some assumptions on the causal structure have to be made for adaptation to be possible.



(a) Marginal distributions



(b) Conditional distributions



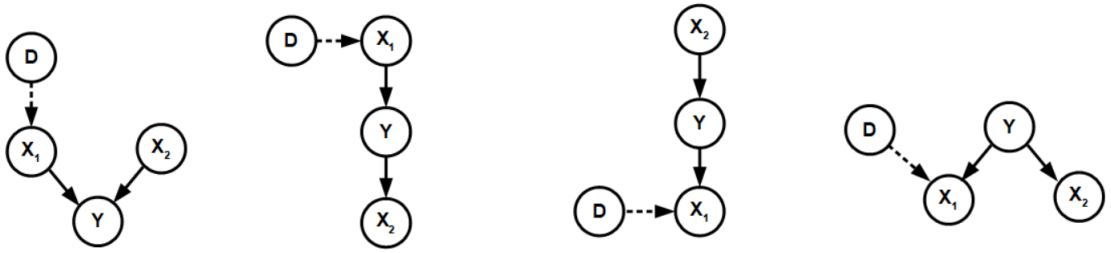
$$
\begin{array}{cc}
P_S(X_1 = 0) = 0.5, & P_T(X_1 = 0) = 0.25, \\
P(Y = 0|X_1 = 0) = 7/8, & P(Y = 0|X_1 = 1) = 1/6, \\
P_S(X_2 = 0|Y = 0) = 0.2, & P_S(X_2 = 0|Y = 1) = 0.8, \\
P_T(X_2 = 0|Y = 0) = 0.7, & P_T(X_2 = 0|Y = 1) = 0.38 (2d.p.)
\end{array}
$$

Figure 18: A simple example of binary classification with two binary features. An equal number ($n = 1000$) of samples were drawn from each domain according to the causal graph (bottom left) and the given probabilities (bottom right). The estimated marginal distributions (top left) suggest X_2 as domain invariant feature, whereas X_1 seems to vary heavily between domains. However, the conditional distributions (top right) reveal that the causal parent X_1 is actually invariant for prediction, whereas X_2 is not. Thus, invariant marginals do not imply invariance for prediction (see X_2), and the converse also does not hold (see X_1).

A.2 Inferring Causal Graphs via Independence Testing and Prior Knowledge

As mentioned in the main body of this work, inferring causal structure from observational data is challenging and a subject of ongoing research. However, when only few variables are involved and background knowledge can be used to limit the number of possible graphs, it is sometimes possible to clearly identify the causal DAG through conditional independence testing (assuming sufficient data is available to obtain significant results). Below we give an example of such a scenario in the context of DA to illustrate the necessary assumptions and procedure. For some interesting recent work on a logic-based approach to causal discovery from conditional independence test results we refer to Magliacane et al. (2016).

Assume that in an unsupervised DA setting with two features X_1 and X_2, we know from expert knowledge that the domain shift only directly causally influences X_1. Moreover, assume that there are no hidden confounders so that each feature is either a direct cause or a direct effect of the target variable Y. Importantly, we do not assume that we know whether X_1 and X_2 are cause or effect. Then there are exactly four possible causal DAGs which are consistent with our assumptions. These four graphs are shown in Fig. 19. Considering the conditional independences which are testable from labelled source and unlabelled target data alone (i.e., no statement involving both D and Y), we find that each graph satisfies a different set of such statements, which are listed in Table 1. Hence, given that we can test these statements and obtain reliable results, we are able to uniquely identify the correct causal DAG.



(a) (b) (c) (d)

Figure 19: The four possible DAGs coherent with the assumptions that (i) D only directly causally influences X_1 and (ii) there are no hidden confounders, i.e., X_1 and X_2 are either direct causes or effects of Y.

DAG	**(a)**	**(b)**	**(c)**	**(d)**
testable on source data	$X_1 \perp\!\!\!\perp X_2$	$X_1 \perp\!\!\!\perp X_2 \vert Y$	$X_1 \perp\!\!\!\perp X_2 \vert Y$	$X_1 \perp\!\!\!\perp X_2 \vert Y$
testable on pooled data	$X_2 \perp\!\!\!\perp D$	$X_2 \perp\!\!\!\perp D \vert X_1$	$X_2 \perp\!\!\!\perp D$	$X_2 \perp\!\!\!\perp D$, $X_2 \perp\!\!\!\perp D \vert X_1$

Table 1: Conditional independence statements for the four DAGs from Fig. 19, testable on only the source data (those involving Y and not D) and on the pooled data (those involving D and not Y).

Appendix B. Algorithms

In this Appendix, we provide pseudo-code for training a semi-generative model using MLEs or a Bayesian approach, and how to use such a model to make predictions for new data. Recall that our approach rests on the assumption that cause and effect features are conditionally independent given Y, and that the domain shift only directly affects causal features, see Section 3 for details.

Algorithm 1 describes how to train a semi-generative model for a multi-class classification task under unsupervised covariate shift adaptation using maximum likelihood estimation.

Algorithm 1 Semi-generative maximum likelihood estimation for classification

Input: labelled source data $\{(x_C^i, y^i, x_E^i)\}_{i=1}^{n_S}$, unlabelled target data $\{(x_C^j, x_E^j)\}_{j=n_S+1}^{n_S+n_T}$, mechanisms $P(Y|X_C, \theta_Y)$ and $P(X_E|Y, \theta_E)$, hyperparameter $\lambda \in (0, 1)$, initial guess θ_0, learning rate α

Output: pooled-data, semi-generative MLE $\hat{\theta} = (\hat{\theta}_Y, \hat{\theta}_E)$

$\ell_S(\theta) \leftarrow \sum_{i=1}^{n_S} \log P(y^i|x_C^i, \theta_Y) + \log P(x_E^i|y^i, \theta_E)$
$\ell_T(\theta) \leftarrow \sum_{j=n_S+1}^{n_S+n_T} \log\big(\sum_{y=1}^{k} P(y|x_C^j, \theta_Y) P(x_E^j|y, \theta_E)\big)$
$\ell(\theta) \leftarrow \frac{\lambda}{n_S}\ell_S(\theta) + \frac{1-\lambda}{n_T}\ell_T(\theta)$
$t \leftarrow 0$
while not converged **do**
    $\theta_{t+1} \leftarrow \theta_t + \alpha\nabla_\theta \ell_P(\theta_t)$
    $t \leftarrow t + 1$
end while
$\hat{\theta} \leftarrow \theta_t$

Algorithm 2 details how to predict class probabilities from a semi-generative model and parameter estimate $\hat{\theta} = (\hat{\theta}_Y, \hat{\theta}_E)$.

Algorithm 2 Label prediction from semi-generative model

Input: new observation from target domain $(x_C^{\text{new}}, x_E^{\text{new}})$, meachanisms $P(Y|X_C, \theta_Y)$ and $P(X_E|Y, \theta_E)$, parameter estimate $\hat{\theta} = (\hat{\theta}_Y, \hat{\theta}_E)$

Output: label probabilities $p_1, ..., p_k$

$Z = \sum_{y=1}^{k} P(y|x_C^{\text{new}}, \hat{\theta}_Y) P(x_E^{\text{new}}|y, \hat{\theta}_E)$
for $y = 1, ..., k$ **do**
    $p_y = P(y|x_C^{\text{new}}, \hat{\theta}_Y) P(x_E^{\text{new}}|y, \hat{\theta}_E)/Z$
end for

Algorithm 3 describes how to perform Bayesian inference with a semi-generative model. It uses the Metropolis-Hastings algorithm as an example for sampling from the posterior distribution. However, more elaborate sampling approach are, of course, possible.

Algorithm 3 Semi-generative Bayes for classification (Metropolis-Hastings sampling)

Input: labelled source data $\{(x_C^i, y^i, x_E^i)\}_{i=1}^{n_S}$, unlabelled target data $\{(x_C^j, x_E^j)\}_{j=n_S+1}^{n_S+n_T}$, mechanisms $P(Y|X_C, \theta_Y)$ and $P(X_E|Y, \theta_E)$, prior $\pi(\theta)$, hyperparameter $\lambda \in (0, 1)$, proposal distribution $\mathcal{N}(\theta_{t+1}|\theta_t, \sigma^2)$

Output: samples from posterior distribution $\theta^{(k)}$

$\ell_S(\theta) \leftarrow \sum_{i=1}^{n_S} \log P(y^i|x_C^i, \theta_Y) + \log P(x_E^i|y^i, \theta_E)$
$\ell_T(\theta) \leftarrow \sum_{j=n_S+1}^{n_S+n_T} \log \big(\sum_{y=1}^{k} P(y|x_C^j, \theta_Y) P(x_E^j|y, \theta_E) \big)$
$\ell(\theta) \leftarrow \log \pi(\theta) + (n_S + n_T) \big(\frac{\lambda}{n_S} \ell_S(\theta) + \frac{1-\lambda}{n_T} \ell_T(\theta) \big)$
$t \leftarrow 0$
$\theta^{(0)} \leftarrow 0$
while Markov chain not mixed **do**
    $\theta_{\text{cand}} \leftarrow \mathcal{N}(\theta_{\text{cand}} \,|\, \theta^{(t)}, \sigma^2)$
    $u \leftarrow U(0, 1)$
    if $\exp(\ell(\theta_{\text{cand}}) - \ell(\theta^{(t)})) > u$ **then**
        $\theta^{(t+1)} \leftarrow \theta_{\text{cand}}$
    else
        $\theta^{(t+1)} \leftarrow \theta^{(t)}$
    end if
    $t \leftarrow t + 1$
end while

Appendix C. Proofs

This Appendix contains the proofs omitted in the main document.

Proof of Proposition 2: *Denoting the pdf of a normally distributed random variable with mean μ and standard deviation σ by $\phi(x|\mu,\sigma^2)$ it follows that:*

$$\begin{aligned}
y^* &= \text{argmax}_y \, P(y \,|\, x_C^*, x_E^*, \theta) \\
&= \text{argmax}_y \, \frac{P(y \,|\, x_C^*, \theta)\, P(x_E^*|y,\theta)}{P(x_E^*|x_C^*,\theta)} \\
&= \text{argmax}_y \, P(y \,|\, x_C^*, \theta_Y)\, P(x_E^*|y,\theta_E) \\
&= \text{argmax}_y \, \phi(y \,|\, a + bx_C^*, \sigma_Y^2)\, \phi(x_E^* \,|\, c + dy, \sigma_E^2) \\
&= \text{argmax}_y \, \phi(y \,|\, a + bx_C^*, \sigma_Y^2)\, \phi\Big(y \,|\, \frac{x_E^* - c}{d}, \frac{\sigma_E^2}{d^2}\Big) \\
&= \text{argmax}_y \, \phi\Big(y \,|\, \frac{\sigma_E^2(a + bx_C^*) + d^2\sigma_Y^2(\frac{x_E^* - c}{d})}{\sigma_E^2 + d^2\sigma_Y^2}, \frac{\sigma_E^2\sigma_Y^2}{\sigma_E^2 + d^2\sigma_Y^2}\Big) \\
&= \frac{\sigma_E^2(a + bx_C^*) + d^2\sigma_Y^2(\frac{x_E^* - c}{d})}{\sigma_E^2 + d^2\sigma_Y^2}
\end{aligned}$$

where the penultimate equality follows from a standard result about the product of two normal pdfs:

$$\phi(x \,|\, \mu_1, \sigma_1^2)\, \phi(x \,|\, \mu_2, \sigma_2^2) = \phi\Big(x \,|\, \frac{\sigma_2^2\mu_1 + \sigma_1^2\mu_2}{\sigma_1^2 + \sigma_2^2}, \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2}\Big)$$

■

Appendix D. Code

The MATLAB code used to implement our semi-generative modelling approach on synthetic and real-world data is publicly available on github through:

`https://github.com/Juliusvk/Semi-Generative-Modelling-`

It also includes the MATLAB domain adaptation toolbox by Ke Yan (2016) which we used to compare our approach with feature transformation methods for DA (TCA, MIDA, SA, GFK, and ITL), as well as the paper and full protein-protein signalling network data set by Sachs et al. (2005). To reproduce our results, simply download all the code from the link above, and run the scripts `ClassificationExperiments.m`, `regression_experiment.m`, and `sachs_experiment.m` from the corresponding folders with an appropriate choice of parameters.

Contents

1 Introduction **1**
1.1 Contributions and Organisation of the Paper . . . 3

2 Preliminaries & Previous Work **3**
2.1 Unsupervised Domain Adaptation (DA) . . . 3
2.2 Covariate Shift (CS) . . . 3
2.3 Causal Models . . . 4
2.4 Independence of Causal Mechanisms (ICM) . . . 6
2.5 Other Previous Work . . . 6

3 Semi-Generative Modelling **7**
3.1 Assumptions . . . 7
3.2 Analysis . . . 8
3.3 Modelling Approach . . . 10
3.4 (Log-)Likelihoods . . . 11

4 Experiments **12**
4.1 Estimators . . . 12
4.2 Model Fitting: Maximum Likelihood and Bayesian Approaches . . . 13
4.3 Synthetic Classification Experiments . . . 13
4.4 Synthetic Regression Experiments . . . 14
4.5 Real-Data Regression Experiments . . . 16
4.6 Choosing λ . . . 17
4.7 Evaluation . . . 17

5 Results **19**
5.1 Synthetic Classification Results . . . 20
5.2 Synthetic Regression Results . . . 23
5.3 Real-Data Regression Results . . . 24

6 Discussion **27**
6.1 Model Flexibility and the Role of λ . . . 27
6.2 Behaviour Under Model Misspecification . . . 29
6.3 Comparison With Feature Transformation Methods . . . 30
6.4 General Applicability . . . 31

7 Conclusions **33**

References **33**

A Causality and Domain Adaptation: Illustrative Examples **37**
A.1 Why Does Causal Structure Matter for DA? . . . 37
A.2 Inferring Causal Graphs via Independence Testing and Prior Knowledge . . . 38

B Algorithms **39**

C Proofs **41**

D Code **41**