Proceedings of Machine Learning Research vol 140:1–22, 2022 1st Conference on Causal Learning and Reasoning

Towards efficient representation identification in supervised learning

Author names withheld

Editor: Under Review for CLeaR 2022

Abstract

Humans have a remarkable ability to disentangle complex sensory inputs (e.g., image, text) into simple factors of variation (e.g., shape, color) without much supervision. This ability has inspired many works that attempt to solve the following question: how do we invert the data generation process to extract those factors with minimal or no supervision? Several works in the literature on non-linear independent component analysis have established this negative result; without some knowledge of the data generation process or appropriate inductive biases, it is impossible to perform this inversion. Hence, in recent years much progress has been made on disentanglement under structural assumptions, e.g. when we have access to auxiliary information that makes the factors of variation conditionally independent. However, existing work requires a lot of auxiliary information, e.g., in supervised classification, it prescribes that the number of label classes should be at least equal to the total dimension of all the factors of variation. In this work, we depart from these assumptions and focus on the following questions: a) How can we get disentanglement when the auxiliary information does not provide conditional independence over the factors of variation? b) Can we reduce the amount of auxiliary information required for disentanglement? For a class of models where auxiliary information does not ensure conditional independence, we show theoretically and experimentally that disentanglement (to a large extent) is possible even when the auxiliary information dimension is much less than the dimension of the true latent representation.

Keywords: disentanglement, non-linear independent component analysis

1. Introduction

Representation learning (Bengio et al., 2013) aims to extract low dimensional representations from high dimensional complex datasets with the hope that if these representations succinctly capture factors of variation that describe the high dimensional data (e.g., extract features characterizing shape of object in an image), then these representations can be leveraged to achieve good performance on new downstream tasks with minimal supervision. Large scale pre-trained language models demonstrate major success of representation learning based approaches (Brown et al., 2020; Wei et al., 2021; Radford et al., 2021). However, we should look at these results with a dose of caution as neural networks have also been shown to be susceptible to out-of-distribution generalization failures (Beery et al., 2018; Geirhos et al., 2020; Peyrard et al., 2021).[1] To address out-of-distribution generalization failures, recent works (Schölkopf, 2019; Schölkopf et al., 2021; Wang and Jordan, 2021) have argued in favor of incorporating causal principles into standard training paradigms – supervised (Arjovsky et al., 2019) and unsupervised (von Kügelgen et al., 2021). The issue is that the current deep learning paradigm does not imbibe and exploit key principles of causality (Pearl, 2009; Schölkopf, 2019) – invariance principle, independent causal mechanisms principle, causal factorization. This is because the traditional causal inference requires access to structured random variables whose distributions can be decomposed using causal factorization, which is impossible

1. https://oatml.cs.ox.ac.uk/blog/2021/06/27/clip_mechanism.html

© 2022 .

with complex datasets such as images or text. Therefore, to leverage the power of deep learning and causal principles, we first need to disentangle raw datasets to obtain the causal representations that generated the data and then exploit tools from causal structure learning to pin down the relationships between the representations. (Ke et al., 2019; Brouillard et al., 2020).

It has been shown that the general process of disentanglement is impossible in the absence of side knowledge of the structure of the data generation process (Hyvärinen and Pajunen, 1999; Locatello et al., 2019). However, under additional structural assumptions on the data generation process, it is possible to invert the data generation process and recover the underlying factors of variation Hyvarinen and Morioka (2016). Recently, there have been works (Hyvarinen et al., 2019; Khemakhem et al., 2020a) which present a general framework that relies on auxiliary information (e.g., labels, timestamps) to disentangle the latents.

While existing works (Hyvarinen et al., 2019; Khemakhem et al., 2020a) have made remarkable progress in the field of disentanglement, these works make certain key assumptions highlighted below that we significantly depart from.

• **Labels cause the latents.** In supervised learning datasets, where a human generates the label, there are two ways to think about the data generation process – a) labels cause the latent variables and b) latent variables cause the labels. Schölkopf et al. (2012) argue for the former view, i.e., labels generate the latents, while Arjovsky et al. (2019) argue for the latter view, i.e., latents generate the label (See Figure 1). Current non-linear ICA literature (Khemakhem et al., 2020a; Hyvarinen et al., 2019) assume that the label knowledge renders latent factors of variation conditionally independent and is thus compatible with the perspective that the labels cause the latents (Schölkopf et al., 2012). Our goal is to enable disentanglement when latents cause the label (Arjovsky et al., 2019).

• **Amount of auxiliary information.** Existing works (Khemakhem et al., 2020a) (Theorem 1), Khemakhem et al. (2020b) require a lot of auxiliary information, e.g., the number of label classes should be twice the total dimension of the latent factors of variation to guarantee disentanglement. We seek to enable disentanglement with lesser auxiliary information.

Contributions. We consider the following data generation process – latent factors generate the observations (raw features) and the labels for multiple tasks, where the latent factors are mutually independent. We study a natural extension of the standard empirical risk minimization (ERM) (Vapnik (1992)) paradigm. The most natural heuristic for learning representations is to train a neural network using ERM and use the output from the representation layer before the final layer. In this work, we propose to add a constraint on ERM – all the components of the representation layer are mutually independent. Our main findings for the representations learned by constrained ERM are summarized below.

• If the number of tasks is at least equal to the dimension of the factors of variation and latent factors are not Gaussian; then we can recover latents up to permutation and scaling.

• If the number of tasks is only one and the latents come from exponential family whose log-density can be expressed as a polynomial, then under further constraints on both the learner's inductive bias and the function being inverted, we can recover latents up to permutation and scaling.

• To actually implement constrained ERM, we propose a simple two-step approximation. In the first step, we train a standard ERM based model, and in the subsequent step, we carry out linear ICA (Comon, 1994) on the representation extracted from ERM. We carry out experiments with the above procedure for regression and classification. Our experiments show that even with the approximate procedure it is possible to recover the true latents up to permutation and scaling when the number of tasks is smaller than the latent dimension.

2. Related work

Non-linear ICA with auxiliary information. We first describe the set of works in non-linear ICA where the time index itself serves as auxiliary information. Hyvarinen and Morioka (2016) showed that if each component of the latent vector evolves independently and follows a non-stationary time series without temporal dependence, then non-linear identification is possible. In contrast, Hyvarinen and Morioka (2017) showed that if the latent variables are mutually independent, with each component evolving in time following a stationary time series with temporal dependence, then also identification is possible. Khemakhem et al. (2020a); Hyvarinen et al. (2019); Khemakhem et al. (2020b) further generalized the previous results; in these works instead of using time, the authors require observation of auxiliary information. Note that we mentioned that Khemakhem et al. (2020a) has a limitation that auxiliary information renders latents conditionally independent. This assumption was relaxed to some extent in Khemakhem et al. (2020b), however, the model in Khemakhem et al. (2020b) is not compatible with the perspective that we consider, i.e, latent variables generate the label. Recently, Roeder et al. (2020) studied the issue of representation identification for classification (and self-supervised learning models), where it was shown that if there is sufficient number of class labels (at least as many as the dimension of the latent), then two neural network representations are related by a linear transformation. Their work does not focus on recovering the true latents and instead studies whether neural networks learn similar representations across different seeds.

Other works. In another line of work Locatello et al. (2020); Shu et al. (2019), the authors study the role of weak supervision in assisting disetanglement. Note that this line of work is very different from us – these models do not consider labelled supervised learning datasets and instead use different sources of supervision, as an example in Locatello et al. (2020) the authors use multiple views of the same object as a form of weak supervision.

3. Efficient Identification of Latent Factors

3.1. Data generation process

Before we describe the data generation process that we use, we give an example of the data generation process that is compatible with the assumptions in Khemakhem et al. (2020a).

$$\begin{aligned} Y &\leftarrow \mathsf{Bernoulli}\left(\frac{1}{2}\right) \\ Z &\leftarrow \mathcal{N}(Y\mathbf{1}, \mathsf{I}) \\ X &\leftarrow g(Z) \end{aligned} \tag{1}$$

where $\mathsf{Bernoulli}(\frac{1}{2})$ is the uniform Bernoulli distribution over $\{0, 1\}$, $\mathcal{N}$ is normal distribution, $\mathbf{1} \in \mathbb{R}^d$ is the vector that together with the label Y selects the mean of the latent Z, I is a d dimensional identity matrix, g is a bijection that generates the observations X. In Figure 1 (a), we show the causal directed acyclic graph (DAG) corresponding to the above data generation process. In the above data generation process labels cause the latents (Schölkopf et al., 2012). However, (Arjovsky et al., 2019) take the opposite perspective and say that the labels are generated from the latents. Most of the current non-linear ICA models are compatible with the former view, i.e., the labels generate the latents, we believe it is important to study non-linear ICA when latents generate the label. We now describe our data generation process, which is consistent with this perspective.

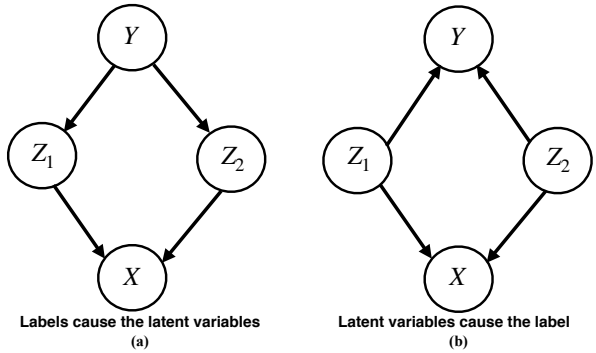

Figure 1: Data generation process in Khemakhem et al. (2020b) is shown in (a) and comparing it with the data generation process in this work (b).

Assumption 1 *The data generation process for regressions is described as*

$$\begin{aligned} Z &\leftarrow h(N_Z), \\ Y &\leftarrow \Gamma Z + N_Y, \\ X &\leftarrow g(Z), \end{aligned} \tag{2}$$

where $N_Z \in \mathbb{R}^d$ is noise, h generates $Z \in \mathbb{R}^d$ (the components of latent Z are mutually independent, non-Gaussian, and have finite second moments), $g : \mathbb{R}^d \to \mathbb{R}^d$ is a bijection that generates the observations X, $\Gamma \in \mathbb{R}^{k \times d}$ is a matrix that generates the label $Y \in \mathbb{R}^k$ and $N_Y \in \mathbb{R}^k$ is the noise vector (N_Y is independent of Z and $\mathbb{E}[N_Y] = 0$). Note that d is the dimension of the latent and k corresponds to the number of tasks. We adapt the data generation process to multi-task classification as follows. We only need to change the label generation process.

$$Y \leftarrow \mathsf{Bernoulli}\Big(\sigma\Big(\Gamma Z\Big)\Big), \tag{3}$$

where $\mathsf{sigmoid}(\sigma)$ *is applied elementwise to ΓZ to output the probabilities of the tasks and* $\mathsf{Bernoulli}$ *operates on these probabilities elementwise to generate label vector $Y \in \{0,1\}^k$ for the k tasks.*

We now contrast the above data generation process in Assumption 1, where latents generate the label, to the one compatible with Khemakhem et al. (2020a) described in equation (1), where labels generate the latents. We contrast the the DAGs of the two data generation processes in Figure 1. The nature of the auxiliary information in the above data generation process in Assumption 1 is very different from the one in equation 1; conditioning on the auxiliary information in our data generation process does not make the latents independent.

Remark. Recall that the classic non-identifiability results in Locatello et al. (2019) and Hyvärinen and Pajunen (1999) assumed that the latent factors of variation are all mutually independent. These results implied that without some knowledge, e.g., auxiliary information (labels), it is impossible to disentangle the latent variables. While Khemakhem et al. (2020a) showed that it is possible to succeed in the presence of auxiliary information, their data generation process assumes that the latent variables are not mutually independent and is thus not consistent with assumptions in Locatello

et al. (2019) and Hyvärinen and Pajunen (1999). Our data generation process for the latent variables is consistent with the models in Locatello et al. (2019) and Hyvärinen and Pajunen (1999) and introduces auxiliary information as downstream of the latents.

3.2. Identifiability

Our objective is to learn the model g^{-1} (or the generator g) from data about observations and labels (X, Y) such that for new observations X we can recover the latent Z that generated X. We can not always learn the exact latent variables Z but may only identify them to some degree. This section formalizes the notion of identification commonly used in literature. We denote the model learned as $\tilde{g}^{-1}$ (with its inverse $\tilde{g}$). We now describe a general notion of identification for the learned map $\tilde{g}^{-1}$ with respect to the true map g^{-1}.

Definition 1 ***Identifiability up to $\mathcal{A}$.*** *If the learned map $\tilde{g}^{-1}$ and the true map g^{-1} are related by some bijection $a \in \mathcal{A}$, such that $\tilde{g}^{-1} = a \circ g^{-1}$ (or equivalently $\tilde{g} = g \circ a^{-1}$), then $\tilde{g}^{-1}$ is said to learn g^{-1} up to bijections in $\mathcal{A}$. We denote this $\tilde{g}^{-1} \sim_{\mathcal{A}} g^{-1}$.*

In the above definition, if $\mathcal{A}$ was the set of all the permutation maps $\mathcal{P}$, then $\tilde{g}^{-1} \sim_{\mathcal{P}} g^{-1}$ denotes identification up to permutations. Permutation identification guarantees that we learn the the true latent vector but do not learn the indices of the true latent, which for most purposes is not important.

3.3. Independence Constrained Empirical Risk Minimization

As stated earlier, that we want to learn the model g^{-1} (or the generator g) from data about observations and labels (X, Y). We first train a supervised learning model to predict the labels Y from X. From this model we extract representations that we hope recover the true latent Z. For the rest of the work, we will assume that the predictor that we learn takes the form $\Theta \circ \Phi$, where $\Theta \in \mathbb{R}^{k \times d}$ is a linear predictor that operates on the representation $\Phi : \mathbb{R}^d \to \mathbb{R}^d$. As a result, the hypothesis space of the functions that the learner searches over has two parts: $\Theta \in \mathcal{H}_\Theta$ corresponding to the hypothesis class of linear maps, and $\Phi \in \mathcal{H}_\Phi$, where $\mathcal{H}_\Phi$ corresponds to the hypothesis class over the representations. We measure the performance of the predictor on an instance (X, Y) using the loss ℓ (e.g., square error for regression, cross-entropy loss for classification) as $\ell\big(Y, \Theta \circ \Phi(X)\big)$. Define the risk achieved by a predictor $\Theta \circ \Phi$ as $R\big(\Theta \circ \Phi\big) = \mathbb{E}\Big[\ell\big(Y, \Theta \circ \Phi(X)\big)\Big]$, where the expectation is taken with respect to the data (X, Y).

Independence constrained ERM (IC-ERM). We want the learner to carry out constrained empirical risk minimization, where the constraint is placed on the representation layer that the all components of the representation are mutually independent. We provide the formal definition of mutual independence for the convenience of the reader below.

Definition 2 ***Mutual independence.*** *A random vector $V = [V_1, \cdots, V_d]$ is said to be mutually independent if for each subset $\mathcal{M} \subseteq \{1, \cdots, d\}$ the following identity holds where P denotes the probability.*

$$P(\{V_i\}_{i \in \mathcal{M}}) = \prod_{i \in \mathcal{M}} P(V_i), \tag{4}$$

The independence constrained ERM (IC-ERM) is stated as follows

$$\min_{\Theta \in \mathcal{H}_\Theta, \Phi \in \mathcal{H}_\Phi} R(\Theta \circ \Phi) \;\; \text{s.t. } \Phi(X) \text{ is mutually independent (Definition 2)} \tag{5}$$

ERM with linear ICA (ERM-ICA). We propose a simple two step procedure as an approximate approach to solve the above problem. The learner first carries out standard ERM stated as

$$\Theta^\dagger, \Phi^\dagger \in \underset{\Theta \in \mathcal{H}_\Theta, \Phi \in \mathcal{H}_\Phi}{\arg\min} \; R(\Theta \circ \Phi) \tag{6}$$

The learner then searches for a linear transformation Ω that when applied to $\Phi^\dagger$ yields a new representation with components that are mutually independent. We state this as follows. Find an invertible $\Omega \in \mathbb{R}^{d \times d}$ such that

$$Z^\dagger = \Omega \circ \Phi^\dagger(X) \text{ where the components of } Z^\dagger \text{ are mutually independent} \tag{7}$$

Note that a solution to the above equation (7) does not always exist. However, if we can find a Ω that satisfies the above (7), then the classifier $\Theta \circ \Omega^{-1}$ and the representation $\Omega \circ \Phi^\dagger(X)$ together solve the IC-ERM (5) assuming $\Theta \circ \Omega^{-1} \in \mathcal{H}_\Theta$ and $\Omega \circ \Phi^\dagger \in \mathcal{H}_\Phi$. To find a solution to the equation (7) we resort to the approach of linear ICA (Comon, 1994). The approach has two steps. We first whiten $\Phi^\dagger(X)$. [2] Define V to be the covariance matrix of $\Phi^\dagger(X)$. If the covariance V is invertible, [3] then the eigendecomposition of V is given as $V = U\Lambda^2 U^\mathsf{T}$. We obtain the whitened data $\Phi^*(X)$ as follows $\Phi^*(X) = \Lambda^{-1} U^\mathsf{T} \Phi^\dagger(X)$. Consider a linear transformation of the whitened data and denote it as $Z^* = \Omega \circ \Phi^*(X)$ and construct another vector $Z^{'}$ such that its individual components are all independent and are equal in distribution to the corresponding components in Z^*. Our goal is to find an Ω such that the mutual information between Z^* and $Z^{'}$ is minimized; to make dependence on Ω explicit we denote the mutual information between $Z^{'}$ and Z^* as $I(\Omega \circ \Phi^*(X))$. We state this as the following optimization

$$\Omega^\dagger \in \underset{\Omega, \Omega \text{ is invertible}}{\arg\min} \; I(\Omega \circ \Phi^*(X)) \tag{8}$$

We have followed the steps of linear ICA (Comon, 1994) on the ERM based representation $\Phi^\dagger(X)$, i.e., we first whiten $\Phi^\dagger(X)$ to get $\Phi^*(X)$ and then solve the above equation (8). The above ERM-ICA procedure outputs a classifier $\Theta^\dagger \circ (\Omega^\dagger)^{-1}$ and representation $\Omega^\dagger \circ \Phi^\dagger$ that is an approximate solution to the IC-ERM problem (5). While the proposed ERM-ICA procedure is a simple approximation, we do not know of other works that have investigated this approach theoretically and experimentally for recovering the latents. We believe that building better approximations to directly solve the IC-ERM, which do not involve a two-step approach like ERM-ICA approach is a fruitful future work. In the next section, we discuss how does the IC-ERM objective (equation (5)) and the proposed approximation ERM-ICA (equations (6) and (8)) perform in terms of recovering the latent factors when the number of tasks k is equal to the dimension of the latent d.

3.4. Identification when number of tasks is equal to the latent data dimension

In this section, we study the setting when the number of tasks k is equal to the dimension of the latent d. In the next theorem, we state the results for latent recovery for data generation process described in Assumption 1. Assumption 1 has two data generation processes – one for regression and the other for classification and the theorem holds for both the settings.

2. For simplicity, we assume $\Phi^\dagger(X)$ is zero mean, although our analysis extends to the case with non-zero mean as well. We also assume that $\Phi^\dagger(X)$ has finite second moments.
3. If it is not invertible, then we first need to project $\Phi^\dagger(X)$ into the range space of V

Assumption 2 ***Assumption on $\mathcal{H}_\Phi$ and $\mathcal{H}_\Theta$.*** *$g^{-1} \in \mathcal{H}_\Phi$. For the case when $k = d$ the set $\mathcal{H}_\Theta$ corresponds to the set of all invertible matrices. $\Gamma \in \mathcal{H}_\Theta$*

The above assumption ensures that the the true g^{-1} and Γ are in the respective hypothesis classes that the learner searches over.

Theorem 1 *If Assumptions 1, 2 hold and the number of tasks k is equal to the dimension of the latent d, then the solution $\Theta^\dagger \circ \Phi^\dagger$ to IC-ERM (5) with ℓ as square loss for regression and cross-entropy loss for classification identifies true Z up to permutation and scaling.*

The proofs to all the theorems are in the Appendix. In the above theorem, we show that when the number of tasks is equal to the dimension of the latent variable, then the solution to equation (5) recovers the true latents up to permutation and scaling. In the next theorem, we show that the same result holds for the two step ERM-ICA approach.

Theorem 2 *If Assumptions 1, 2 hold and the number of tasks k is equal to the dimension of the latent d, then the solution $\Omega^\dagger \circ \Phi^\dagger$ to ERM-ICA ((6), (8)) with ℓ as square loss for regression and cross-entropy loss for classification identifies true Z up to permutation and scaling.*

Significance of Theorem 1 and 2. From the above theorem we learn that for the DAGs in Figure 1 (b), we can recover the true latents up to permutation and scaling. This result extends the current disentanglement guarantees (Khemakhem et al., 2020b) that exist for models where labels cause the latents (latents are conditionally independent) to the settings where latents cause the label (latents are not conditionally independent). In multi-task learning literature Caruana (1997); Zhang and Yang (2017) it has been argued that learning across multiple tasks with shared layers leads to internal representations that transfer better. The above results show conditions when the ideal data generating representation that is shared across tasks can be recovered.

Remark. Since we use a linear model for the label generation process, one can ask what happens if we apply noisy linear ICA techniques (Davies, 2004; Hyvarinen, 1999) on the label itself when $k = d$ to recover Z followed by a regression to predict Z from X. Noisy linear ICA require the noise distribution to be Gaussian and would not work when N_Y is not a Gaussian. Since we do not make such distributional assumptions on N_Y, we cannot rely on noisy linear ICA on labels.

3.5. Identification when the number of tasks is less than the dimensions of the latent

In this section, we study the setting when the number of tasks k is equal to one. Since this setting is extreme; we need to make stronger assumptions to show latent identification guarantees. Before we lay down the assumptions, we provide some notation. Since we now only have one task instead of using the matrix $\Gamma \in \mathbb{R}^{k\times d}$, we use $\gamma \in \mathbb{R}^d$ to signify the coefficients that generate the label in the one task setting. We assume each component of γ is non-zero. In one-task setting for regressions, the label generation is written as $Y \leftarrow \gamma^\mathsf{T} Z + N_Y$ and the rest of the notation is same as the data generation process in Assumption 1. We rewrite the data generation process in Assumption 1 for one task case in terms of normalized variables $U = Z \odot \gamma$.

Assumption 3 *The data generation process for regressions is described as*

$$\begin{aligned} Z &\leftarrow h(N_Z) \\ Y &\leftarrow \mathbf{1}^\mathsf{T} U + N_Y, \\ X &\leftarrow g^{'}(U), \end{aligned} \tag{9}$$

where $g^{'}(U) = g(U \odot \frac{1}{\gamma})$, where $U \odot \frac{1}{\gamma} = [\frac{U_1}{\gamma_1}, \cdots, \frac{U_d}{\gamma_d}]$ (Note that $g^{'}$ is invertible since g is invertible and each element of γ is also non-zero). We assume that all the components of U are mutually independent and identically distributed (i.i.d.).

In the rest of the section for simplicity, we deal with identification of U. If we identify U up to permutation and scaling, then Z is automatically identified up to permutation and scaling. The predictor we learn is a composition of linear predictor θ and a representation Φ, which is written as $\theta \circ \Phi(X) = \theta^{\mathsf{T}}\Phi(X)$. The learner searches for θ in the set $\mathcal{H}_\theta$, where $\mathcal{H}_\theta$ consists of linear predictors with all non-zero components, and Φ in the set $\mathcal{H}_\Phi$. If the predictor is $\theta^{\mathsf{T}}\Phi(X) = \mathbf{1}^{\mathsf{T}}(\Phi(X) \odot \frac{1}{\theta})$, where $\Phi(X) \odot \frac{1}{\theta}$ is componentwise division expressed as $\Phi(X) \odot \frac{1}{\theta} = [\frac{\Phi_1(X)}{\theta_1}, \cdots, \frac{\Phi_d(X)}{\theta_d}]$. Therefore, instead of searching over $\mathcal{H}_\theta$ such that all components of θ are non-zero, we can fix $\mathcal{H}_\theta = \{\mathbf{1}\}$ and carry out the search over representations $\mathcal{H}_\Phi$ only. For the rest of the section, without loss of generality, we assume that the predictor is of the form $\mathbf{1} \circ \Phi(X) = \mathbf{1}^{\mathsf{T}}\Phi(X)$. We restate the IC-ERM (5) with this parametrization and an additional constraint that all the components are now required to be independent and identically distributed (i.i.d.). We provide a formal definition for the convenience of the reader below, where $\stackrel{d}{=}$ denotes identical in distribution.

Definition 2 ***Independent & Identically Distributed (i.i.d.).*** *A random vector $V = [V_1, \cdots, V_d]$ is said to be i.i.d. if 1) $\Phi_i(X) \stackrel{d}{=} \Phi_j(X)$ $\forall i, j \in \{1, \cdots, d\}$ 2) $P(\{V_i\}_{i \in \mathcal{M}}) = \prod_{i \in \mathcal{M}} P(V_i)$ $\forall \mathcal{M} \subseteq \{1, \cdots, d\}$ where P denotes the probability.*

The reparametrized IC-ERM (5) constraint is stated as follows.

$$\min_{\Phi \in \mathcal{H}_\Phi} R(\mathbf{1} \circ \Phi) \;\; \text{s.t. } \Phi(X) \text{ is i.i.d. (Definition 2)} \tag{10}$$

Next, we state the assumptions on each component of U (recall each component of U is i.i.d. from Assumption 3) and $\mathcal{H}_\Phi$ under which we show that the above procedure in equation (10) recovers true latents U up to permutations. We assume each component of U is a continuous random variable with probability density function (PDF) r. Define the support of each component of U as $\mathcal{S} = \{u \mid r(u) > 0, u \in \mathbb{R}\}$. Define a ball of radius $\sqrt{2}p$ as $\mathcal{B}_p = \{u \mid |u|^2 \leq 2p^2, u \in \mathbb{R}\}$.

Assumption 4 *Each component of U is a continuous random variable from the exponential family with probability density r. $\log(r)$ is a polynomial with degree p (where p is odd) written as*

$$\log\big(r(u)\big) = \sum_{k=0}^{p} a_k u^k$$

where the absolute value of the coefficients of the polynomial are bounded by a_{max}, i.e., $|a_k| \leq a_{\mathsf{max}}$ for all $k \in \{1, \cdots, p\}$, and the absolute value of the coefficient of the highest degree term is at least a_{min}, i.e., $|a_p| \geq a_{\mathsf{min}} > 0$. The support of r is sufficiently large that it contains $\mathcal{B}_p$, i.e., $\mathcal{B}_p \subseteq \mathcal{S}$. The moment generation function of each component i of U, $M_{U_i}(t)$, exists for all t.

Remark on the PDFs under the above assumption. The above assumption considers distributions in the exponential family where the log-PDF can be expressed as a polynomial. Note that as long as the support of the distribution is bounded, every polynomial with bounded coefficients leads to a valid PDF (i.e., it integrates to one), we only need to set the value of a_0 appropriately.

Observe that $\Phi(X)$ can be written as $h(U) = \Phi\Big(g^{'}(U)\Big)$ (since $X = g^{'}(U)$). We write the set of all the maps h constructed from composition of $\Phi \in \mathcal{H}_\Phi$ and $g^{'}$ as $\mathcal{H}_\Phi \circ g^{'}$. Define $w(u_1, \cdots, u_d) = \log\Big(\big|\mathsf{det}\big[J(h(u_1, \cdots, u_d))\big]\big|\Big)$, where det is the determinant, $J(h(u_1, \cdots, u_d))$ is the Jacobian of h computed at $(u_1, \cdots, u_d)$. The set of all the w's obtained from all $h \in \mathcal{H}_\Phi \circ g^{'}$ is denoted as $\mathcal{W}$

Assumption 5 *$\mathcal{H}_\Phi$ consists of bijections that are analytic. For each $\Phi \in \mathcal{H}_\Phi$, the moment generating function of each component $i \in \{1, \cdots, d\}$, $\Phi_i(X)$, denoted as $M_{\Phi_i(X)}(t)$ exists for all t. Each $w \in \mathcal{W}$ is a finite degree polynomial with degree at most q, where the absolute values of the coefficients in the polynomial are bounded by b_{max}.*

Define $p_{\mathsf{min}} = \max\Big\{\kappa \log(8(d-1)), \frac{4a_{\mathsf{max}}(d-1)}{a_{\mathsf{min}}}, \frac{\log\left(\frac{4b_{\mathsf{max}} * n_{\mathsf{poly}}}{a_{\mathsf{min}}}\right)}{2} + q\Big\}$, where n_{poly} is the maximum number of non-zero coefficients in any polynomial $w \in \mathcal{W}$, κ is small constant (see Appendix).

Theorem 3 *If Assumptions 3, 4, 5 hold, $(g')^{-1} \in \mathcal{H}_\Phi$ and p is sufficiently large ($p \geq p_{\mathsf{min}}$), then the solution $\Phi^\dagger(X)$ of equation (10) recovers the true latent U in data generation process in Asssumption 3 up to permutations.*

Proof sketch. We use the optimality condition that the prediction made by the learned model, $\mathbf{1}^\mathsf{T} h(U)$, exactly matches the true mean, $\mathbf{1}^\mathsf{T} U$, along with the constraints that each component of U are i.i.d. and each component of $h(U)$ are i.i.d., to derive that the distributions of U and $h(U)$ are the same. We substitute this condition in the change of variables formula that relates the densities of U and $h(U)$. Using the Assumption 5, we are able to show that if the highest absolute value of U and the highest absolute value of $h(U)$ are not equal, then the term with highest absolute value among U and $h(U)$ will dominate, leading to contradiction in the identity obtained by change of variables formula. Baseed on this, we can conclude that the highest absolute value of U and the highest absolute value of $h(U)$ have to be equal. We iteratively apply this argument and are able to show that all the absolute values of U and $h(U)$ are related by permutation. We can extend this argument to the actual values instead of absolute values. Since h is analytic we argue that the relationship of permutation holds in a neighborhood. Then we use properties of analytic functions (Mityagin, 2015) to conclude that the relationship holds on the entire space.

Why does the bound on p grow linearly in d? We provide some geometric intuition into why the degree of the polynomial, p, of the log-PDF, $\log(r)$, needs to be large. If the dimension of the latent space d is large, then the second term in p_{min} dominates, i.e., p has to grow linearly in d. The simplification in the proof yields that the mapping h has to satisfy the following condition – $\|u\|_k = \|h(u)\|_k$ for all $k \in \{1, \cdots, p\}$, where $\|\cdot\|_k$ is the kth norm. Hence, h is a bijection that preserves all norms up to the pth norm. If U is 2 dimensional, then the a bijection h that preserves the ℓ_1 norm and the ℓ_2 norm is composed of permutations and sign-flips. In general, since U is d dimensional, we need at least d constraints on h in the form $\|u\|_k = \|h(u)\|_k$ and thus $p \geq d$, which ensure that the only map that satisfies these constraints is composed of permutations and sign-flips.

Significance and remarks on Theorem 3. Theorem 3 shows that if we use EC-ERM principle, i.e., constraint the representations to be independent, then we continue to recover the latents even if the number of tasks is small. We can show that the above theorem also extends to binary classification. We admit that some strong assumptions were made to arrive at the above result, while other assumptions such as bound on p growing linearly in d seem necessary, but we would like to

remind the reader that we are operating in the extreme one task regime. In the previous section, when number of tasks was equal to the dimension of the latent, we had shown the success of both the IC-ERM (5) and its approximation ERM-ICA (6), (7). We only analyzed the success of IC-ERM for the setting when task is equal to one and the analysis of the approximation (ERM-ICA) is left to future work. In Theorem 1, 2, we showed that when we have sufficiently many tasks we recover latents with much fewer assumptions. In contrast in Theorem 3, we saw that with more assumptions on the distribution we can guarantee latent recovery with even one task. If we are in the middle, i.e., when the number of tasks are between one and the dimension of the latent, then the above Theorem 3 says that we only require the assumptions (Assumptions 3, 4, 5) to hold for at least one task.

4. Experiments

4.1. Data generation process

Regression. We use the data generation process described in Assumption 1. The components of Z are i.i.d. and follow discrete uniform $\{0, 1\}$ distribution. Each element of the task coefficient matrix Γ is i.i.d. and follows a standard normal distribution. The noise in the label generation is also standard normal. We use a 2-layer invertible MLP to model g and follow the construction used in Zimmermann et al. (2021).[4] We carry out comparisons for three settings, $d = \{16, 24, 50\}$, and vary tasks from $k = \{\frac{d}{2}, \frac{3d}{4}, d\}$. The dataset size used for training and test is 5000 data points, along with a validation set of 1250 data points for hyper parameter tuning.
Classification. We use the data generation process described in Assumption 1. We use the same parameters and dataset splits as regression, except the labels are binary and sampled as follows: $Y \leftarrow \mathsf{Bernoulli}(\sigma(\Gamma Z))$. Also, the noise in the Γ sampling is set to a higher value (10 times that of a standard normal), as otherwise the Bayes optimal accuracy is much smaller.

4.2. Methods, architecture, and other hyperparameters

We compare our method against two natural baselines. **a) ERM.** In this case, we carry out standard ERM (6) and use the representation learned at the layer before the output layer. **b) ERM-PCA.** In this case, we carry out standard ERM (6) and extract the representation learned at the layer before the output layer. We then take the extracted representation and transform it using principal component analysis (PCA); in other words we analyze the representation in the eigenbasis of its covariance matrix. **c) ERM-ICA.** This is the main approach ((6), (7)) that approximates the IC-ERM objective (5). Here we take the representations learnt using ERM (6) and transform them using linear independent component analysis (ICA) (8).

We define mean square error and cross-entropy as our loss objectives for the regression and classification task respectively. In both of the settings, we use a two layer fully connected neural network and train the model using stochastic gradient descent. The architecture and the hyperparameter details for the different settings are provided in Appendix D.

4.3. Metrics

The models are evaluated on the test dataset using the following two metrics.

Label (Y) prediction performance. We use the average R^2 (coefficient of determination) and the average accuracy across tasks in the regression and the classification task respectively. For

4. https://github.com/brendel-group/cl-ica/blob/master/main_mlp.py

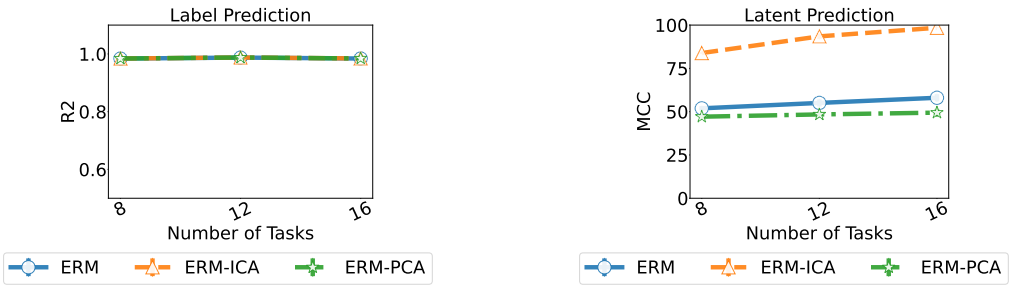

Figure 2: Comparison of label and latent prediction performance (regression, $d = 16$).

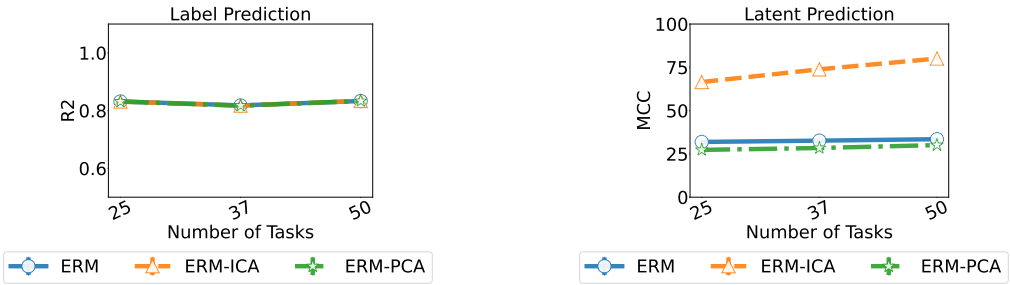

Figure 3: Comparison of label and latent prediction performance (regression, $d = 50$).

the ERM-PCA and ERM-ICA, we take final representations learnt by these methods and train a linear/logistic regression model to predict the label Y for the regression/classification task. We check this metric to ensure that the representations do not lose any information about the label.

Latent (Z) prediction performance. We use mean correlation coefficient (MCC), a standard metric used to measure permutation and scaling based identification (refer (Hyvarinen and Morioka, 2017; Zimmermann et al., 2021) for further details). The metric is computed by first obtaining the correlation matrix ($\rho(Z, \hat{Z})$) between the recovered latents $\hat{Z}$ and the true latents Z. Lets define $|\rho(Z, \hat{Z})|$ as the absolute values of the correlation matrix. Then we find a matching (assign each row to a column in $|\rho(Z, \hat{Z})|$) such that the average absolute correlation is maximized and return the optimal average absolute correlation.

4.4. Results

Regression. Figure 2 and 3 show a comparison of the performance of the three approaches across $d = 16$ and $d = 50$ for varying number of tasks. The results for the case of $d = 24$ are in the Appendix E.1 (due to space limits). In both of the cases we find that the method ERM-ICA is significantly better than the other approaches in terms of guaranteeing permutation and scaling based identification. All the three approaches have similar label prediction performance. We observe a similar trend for the case of $d = 24$ shown in the Appendix E.1.

Classification. Figure 4 and 5 show a comparison of the performance of the three approaches across $d = 16$ and $d = 24$ for varying number of tasks. In both of the cases we find that the method ERM-ICA is better than the other approaches in terms of guaranteeing permutation and scaling based identification, except in the case of 24 data dimensions with a total of 12 tasks. All the three approaches have similar label prediction performance. For classification, unlike regression the improvements are smaller and we do not see improvement for $d = 50$ (comparisons for case of

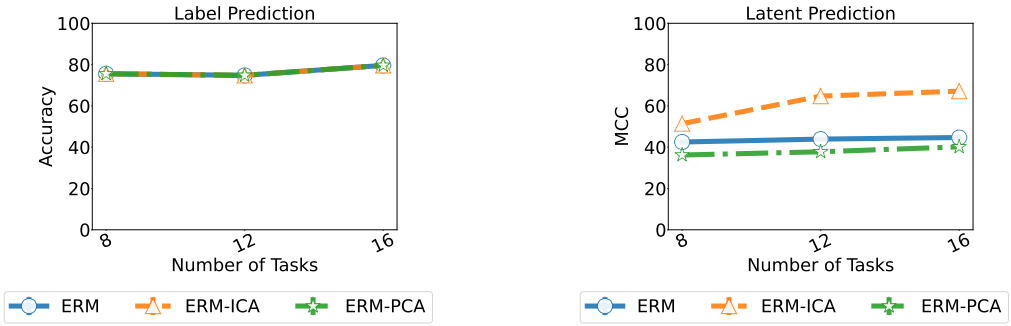

Figure 4: Comparison of label and latent prediction performance (classification, $d = 16$)

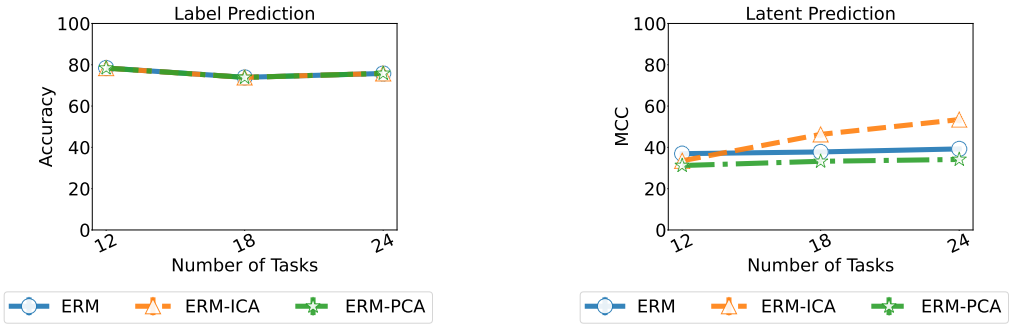

Figure 5: Comparison of label and latent prediction performance (classification, $d = 24$)

$d = 50$ are in the Appendix E.2). We see a worse performance in the classification setting because the ERM model does not learn the Bayes optimal predictor unlike regression.

Discussion. We have shown that ERM-ICA achieves a significant improvement in latent recovery with much fewer tasks (up to $\frac{d}{2}$). However, in our theory we proved that under certain assumptions solving equation (10) can achieve identification even with one task. Note that we only approximate equation (10) with ERM-ICA and if we build better approximations of the ideal approach in equation (10), then we can witness gains with even fewer tasks. We believe that building such approximations is a fruitful future work.

5. Conclusion

In this work, we analyzed the problem of disentanglement in a natural setting, where latent factors cause the labels, a setting not well studied in the ICA literature. We show that if ERM is constrained to learn independent representations, then we can have latent recovery from learnt representations even when the number of tasks is small. We propose a simple two step approximate procedure (ERM-ICA) to solve the constrained ERM problem, and show that it turns out to be quite effective in a variety of experiments. Our analysis highlights the importance of learning independent representations and motivates the development of further approaches to achieve the same in practice.

References

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. *arXiv preprint arXiv:2007.01754*, 2020.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

George Darmois. General analysis of stochastic bonds: special study of linear factorial analysis. *Journal of the International Statistical Institute*, pages 2–8, 1953.

Mike Davies. Identifiability issues in noisy ica. *IEEE Signal processing letters*, 11(5):470–473, 2004.

Willliam Feller. *An introduction to probability theory and its applications, vol 2*. John Wiley & Sons, 2008.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Aapo Hyvarinen. Gaussian moments for noisy independent component analysis. *IEEE signal processing letters*, 6(6):145–147, 1999.

Aapo Hyvarinen and Hiroshi Morioka. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in Neural Information Processing Systems*, 29:3765–3773, 2016.

Aapo Hyvarinen and Hiroshi Morioka. Nonlinear ica of temporally dependent stationary sources. In *Artificial Intelligence and Statistics*, pages 460–469. PMLR, 2017.

Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR, 2019.

Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Bernhard Schölkopf, Michael C Mozer, Chris Pal, and Yoshua Bengio. Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075*, 2019.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020a.

Ilyes Khemakhem, Ricardo Pio Monti, Diederik P Kingma, and Aapo Hyvärinen. Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica. *arXiv preprint arXiv:2002.11537*, 2020b.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pages 4114–4124. PMLR, 2019.

Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. Weakly-supervised disentanglement without compromises. In *International Conference on Machine Learning*, pages 6348–6359. PMLR, 2020.

Boris Mityagin. The zero set of a real analytic function. *arXiv preprint arXiv:1512.07276*, 2015.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Maxime Peyrard, Sarvjeet Singh Ghotra, Martin Josifoski, Vidhan Agarwal, Barun Patra, Dean Carignan, Emre Kiciman, and Robert West. Invariant language modeling. *arXiv preprint arXiv:2110.08413*, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Geoffrey Roeder, Luke Metz, and Diederik P Kingma. On linear identifiability of learned representations. *arXiv preprint arXiv:2007.00810*, 2020.

Bernhard Schölkopf. Causality for machine learning. *arXiv preprint arXiv:1911.10500*, 2019.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. Weakly supervised disentanglement with guarantees. *arXiv preprint arXiv:1910.09772*, 2019.

Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838, 1992.

Julius von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *arXiv preprint arXiv:2106.04619*, 2021.

Yixin Wang and Michael I Jordan. Desiderata for representation learning: A causal perspective. *arXiv preprint arXiv:2109.03795*, 2021.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

Roland S Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. *arXiv preprint arXiv:2102.08850*, 2021.