Linear unit-tests for invariance discovery

Benjamin Aubin
Facebook AI Research

Agnieszka Słowik
Facebook AI Research

Martin Arjovsky
INRIA

Leon Bottou
Facebook AI Research

David Lopez-Paz
Facebook AI, Paris, France

Abstract

There is an increasing interest in algorithms to learn invariant correlations across training environments. A big share of the current proposals find theoretical support in the causality literature but, how useful are they in practice? The purpose of this note is to propose six linear low-dimensional problems—"unit tests"—to evaluate out-of-distribution generalization algorithms. Following initial experiments, none of three recently proposed alternatives pass these tests. By providing the code to automatically replicate all the results in this manuscript (`https://github.com/anonymous`), we hope that the development of unit tests becomes a standard stepping stone for researchers in out-of-distribution generalization.

1 Introduction

Machine learning systems crumble when deployed in conditions different to those of training [Szegedy et al., 2013, Rosenfeld et al., 2018, Alcorn et al., 2019]. To address this issue, recent works in causality [Peters et al., 2015, Arjovsky et al., 2019, Parascandolo et al., 2020] propose to learn correlations invariant across multiple training distributions, and to use those correlations as a proxy for out-of-distribution generalization. However, as we will show, these algorithms perform poorly across a catalog of simple and low-dimensional linear problems. Therefore, our contribution is a standardized set of six "unit tests" that researchers can bear in mind when proposing new solutions for out-of-distribution generalization.

Causal learning algorithms such as Invariant Causal Prediction [Peters et al., 2015, ICP] and Invariant Risk Minimization [Arjovsky et al., 2019, IRM] consume several training datasets, each of them possibly produced by the same "structural equation model" operating under a different "valid intervention". A structural equation model is a list of equations describing how variables influence each other to take their values [Pearl, 2009, Peters et al., 2017]. An intervention perturbs one or more of these equations, and it is "valid" as long as it does not modify the conditional expectation of the target variable given its direct causal parents [Arjovsky et al., 2019]. Then, if the interventions producing our training datasets are diverse, invariant correlations should pertain to the fixed causal mechanism of the target variable. This suggests the possibility of learning about the causal structure of data by searching for statistical invariances. In turn, these invariances enable out-of-distribution generalization: if the causal mechanism of the target variable is invariant across diverse training conditions, we may rely on them to perform robust predictions at novel test conditions.

Unfortunately, invariances are often more difficult to capture than spurious correlations. As an example, consider the task of classifying pictures of cows and camels [Arjovsky et al., 2019]. A "green detector" solves this task to a great extent, since almost all pictures of cows contain green pastures, and almost all pictures of camels show beige sandy landscapes. Following the principle of least effort [Geirhos et al., 2020], learning machines cling to the textural "green-cow" spurious correlation, rather than discovering the shapes that make a cow a cow. Predictors absorbing these

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

"distractor", "shortcut", or "bait" correlations fail catastrophically when deployed under novel test conditions.

Although designed to address this very issue, causal learning algorithms often fail to capture causal invariances in data. The purpose of this note is to share six linear problems that illustrate this phenomenon. Each of these problems contains an invariant causal correlation (*inv*) that we would like to learn, as well as a spurious correlation (*spu*) that we would like to discard. In every dataset, empirical risk minimization absorbs the spurious correlation from the training data, failing to generalize to novel conditions. By releasing the code necessary to automatically replicate all the results in this manuscript, we hope that these "unit tests" become a standard guide when developing new out-of-distribution generalization algorithms.

2 Problems

For each problem, we collect datasets $D^e = \{(x_i^e, y_i^e)\}_{i=1}^{n^e}$ for three environments $e \in \mathcal{E} = \{A, B, C\}$. The input feature vector $x^e = (x^e_{\text{inv}}, x^e_{\text{spu}})$ contains features x^e_{inv} that elicit invariant correlations, and features x^e_{spu} that elicit spurious correlations. Our goal is to construct invariant predictors that estimate the target variable y^e by relying on x^e_{inv}, and ignoring x^e_{spu}. To measure the extent to which an algorithm ignores the features x^e_{spu}, we sample a *train* split, a *validation* split, and a *test* split per problem and environment. Both train and validation splits are built by sampling the structural equations outlined below for each problem and environment. The test split is built analogously, but the features x^e_{spu} are shuffled at random across examples. This way, only those predictors ignoring x^e_{spu} will achieve minimal error at the testing split.

Finally, let $\mathcal{N}_d(\mu, \sigma^2)$ be the d-dimensional Gaussian distribution with mean vector μ and diagonal covariance matrix with diagonal elements σ^2. We fix $n^e = 10^4$ and $d = 10$.

2.1 `example1`: regression from causes and effects

A linear least-squares regression problem where features contain causes and effects of the target variable, as proposed by Arjovsky et al. [2019]. By virtue of considering only valid interventions, the mapping from the causes to the target variable is invariant across environments. Conversely, the mapping from the target variable to the effects changes across environments. To construct the datasets D^e for every $e \in \mathcal{E}$ and $i = 1, \ldots, n^e$, sample:

$$\begin{aligned} x^e_{\text{inv},i} &\sim \mathcal{N}_{\frac{d}{2}}(0, (\sigma^e)^2), & x_i^e &\leftarrow (x^e_{\text{inv},i}, x^e_{\text{spu},i}),\\ \tilde{y}_i^e &\sim \mathcal{N}_{\frac{d}{2}}(x^e_{\text{inv},i}, (\sigma^e)^2), & y_i^e &\leftarrow \frac{2}{d} \cdot 1^\top \tilde{y}_i^e;\\ x^e_{\text{spu},i} &\sim \mathcal{N}_{\frac{d}{2}}(\tilde{y}_i^e, 1), \end{aligned}$$

where $(\sigma^{e=A})^2 = 0.1$, $(\sigma^{e=B})^2 = 1.5$ and $(\sigma^{e=C})^2 = 2$.

Challenges First, the distribution of the feature x^e_{inv} eliciting the invariant correlation changes across environments. This disallows the use of domain-adversarial methods [Ganin et al., 2016], which seek features with matching distribution across environments. Second, the distribution of the residuals varies across environments, disallowing the use of ICP [Peters et al., 2015]. Third, since the target variable is continuous, conditional domain-adversarial techniques [Li et al., 2018] do not apply easily. This example is solved by IRM, and it is analogous to a linear regression variant of the ColorMNIST task [Arjovsky et al., 2019].

2.2 `example2`: cows versus camels

In the spirit of [Beery et al., 2018, Arjovsky et al., 2019], we add a binary classification problem to imitate the introductory example "most cows appear in grass and most camels appear in sand". Let:

$$\begin{aligned} \mu_{\text{cow}} &\sim \mathcal{N}_{\frac{d}{2}}(0, 1), & \mu_{\text{camel}} &= -\mu_{\text{cow}}, & \nu_{\text{animal}} &= 10^{-2},\\ \mu_{\text{grass}} &\sim \mathcal{N}_{\frac{d}{2}}(0, 1), & \mu_{\text{sand}} &= -\mu_{\text{grass}}, & \nu_{\text{background}} &= 1. \end{aligned}$$

To construct the datasets D^e for every $e \in \mathcal{E}$ and $i = 1, \dots, n^e$, sample:

$$j_i^e \sim \text{Categorical}\left(p^e s^e, (1-p^e)s^e, p^e(1-s^e), (1-p^e)(1-s^e)\right);$$

$$x_{\text{inv},i}^e \sim \begin{cases} (\mathcal{N}_{\frac{d}{2}}(0, 10^{-1}) + \mu_{\text{cow}}) \cdot \nu_{\text{animal}} & \text{if } j_i^e \in \{1, 2\}, \\ (\mathcal{N}_{\frac{d}{2}}(0, 10^{-1}) + \mu_{\text{camel}}) \cdot \nu_{\text{animal}} & \text{if } j_i^e \in \{3, 4\}, \end{cases} \qquad x_i^e \leftarrow (x_{\text{inv},i}^e, x_{\text{spu},i}^e);$$

$$x_{\text{spu},i}^e \sim \begin{cases} (\mathcal{N}_{\frac{d}{2}}(0, 10^{-1}) + \mu_{\text{grass}}) \cdot \nu_{\text{background}} & \text{if } j_i^e \in \{1, 4\}, \\ (\mathcal{N}_{\frac{d}{2}}(0, 10^{-1}) + \mu_{\text{sand}}) \cdot \nu_{\text{background}} & \text{if } j_i^e \in \{2, 3\}, \end{cases} \qquad y_i^e \leftarrow \begin{cases} 0 & \text{if } 1^\top x_{\text{inv},i}^e > 0, \\ 1 & \text{else;} \end{cases}$$

where the environment foreground/background probabilities are $p^{e=A} = 0.95$, $p^{e=B} = 0.97$, $p^{e=C} = 0.99$ and the cow/camel probabilities are $s^{e=A} = 0.3$, $s^{e=B} = 0.5$ and $s^{e=C} = 0.7$.

Challenges Achieving zero *population* error while using only x_{inv}^e requires learning large weights. This is difficult when using gradient descent methods, or most forms of regularization. As the dimension of the features space grows, the probability of achieving zero *training* error using only x_{spu}^e increases rapidly. This means that invariance penalties based on training error, such as IRM, may accept solutions using spurious features.

2.3 `example3`: small invariant margin

A linear version of the spiral binary classification problem proposed by [Parascandolo et al., 2020], where the first two dimensions offer an invariant, small-margin linear decision boundary. The rest of the dimensions offers a changing, large-margin linear decision boundary. Let $\gamma = (0.2, 0.2)$, and $\mu^e \sim \mathcal{N}_{d-2}(0, 0.1)$, for all environments.

To construct the datasets D^e for every $e \in \mathcal{E}$ and $i = 1, \dots, n^e$, sample:

$$y_i^e \sim \text{Bernoulli}\left(\frac{1}{2}\right),$$

$$x_{\text{inv},i}^e \sim \begin{cases} \mathcal{N}_2(+\gamma, 10^{-1}) & \text{if } y_i^e = 0, \\ \mathcal{N}_2(-\gamma, 10^{-1}) & \text{if } y_i^e = 1; \end{cases} \qquad x_i^e \leftarrow (x_{\text{inv},i}^e, x_{\text{spu},i}^e).$$

$$x_{\text{spu},i}^e \sim \begin{cases} \mathcal{N}_{d-2}(+\mu^e, 10^{-1}) & \text{if } y_i^e = 0, \\ \mathcal{N}_{d-2}(-\mu^e, 10^{-1}) & \text{if } y_i^e = 1; \end{cases}$$

Challenges We can solve this problem to zero *population* error with high probability using x_{spu}^e alone. Things complicate further, as solving this task using x_{inv}^e forcefully incurs a small amount of *population* error. Therefore, learning algorithms should learn to sacrifice training error to realize that x_{inv}^e leads to the same *maximum margin* classifier across environments (even though the varying margin based on x_{spu}^e is larger!). While the predictor based on x_{inv}^e is the optimal in terms of worst-case out-of-distribution generalization, it is not a causal predictor of the target variable.

2.4 Scrambled variations

We define three additional problems: `example1s`, `example2s`, and `example3s`. These are "scrambled" variations of the three problems described above, respectively. Scrambled variations build observed datasets $D^e = \{(S^\top x_i^e, y_i^e)\}_{i=1}^{n^e}$, where $S \in \mathbb{R}^{d \times d}$ is a random rotation matrix fixed for all environments $e \in \mathcal{E}$. Observing a scrambled version of the variables appearing in the structural equation models requires algorithms to learn a (linear) feature representation under which the desired invariance should be elicited. Because of this reason, we do not compare to brute-force feature selection methods, such as ICP [Peters et al., 2015].

3 Experiments

We provide an initial set of experiments evaluating the following algorithms on our six problems:

- Empirical Risk Minimization [Vapnik, 1998, ERM] minimizes the error on the union of all the training splits. We leverage the cross-validated estimators of scikit-learn package [Pedregosa et al., 2011].

	ANDMask	**ERM**	**IGA**	**IRMv1**	**Oracle**
`example1.A`	0.18 ± 0.00	2.36 ± 0.04	3.69 ± 0.16	0.19 ± 0.03	0.05 ± 0.00
`example1.B`	12.45 ± 0.21	29.16 ± 0.73	36.62 ± 1.33	12.50 ± 0.53	11.04 ± 0.09
`example1.C`	22.07 ± 0.35	50.14 ± 0.58	62.32 ± 0.97	22.14 ± 0.59	20.07 ± 0.34
`example1s.A`	0.18 ± 0.05	2.36 ± 0.05	5.06 ± 0.08	0.18 ± 0.02	0.05 ± 0.00
`example1s.B`	20.92 ± 2.91	29.03 ± 0.47	49.47 ± 0.61	12.67 ± 1.00	11.28 ± 0.13
`example1s.C`	37.47 ± 4.90	50.36 ± 0.42	84.95 ± 1.04	22.54 ± 1.66	20.23 ± 0.33
`example2.A`	0.44 ± 0.02	0.34 ± 0.15	0.44 ± 0.02	0.44 ± 0.02	0.12 ± 0.14
`example2.B`	0.50 ± 0.01	0.37 ± 0.16	0.50 ± 0.01	0.49 ± 0.01	0.12 ± 0.14
`example2.C`	0.44 ± 0.02	0.33 ± 0.15	0.44 ± 0.02	0.43 ± 0.03	0.13 ± 0.14
`example2s.A`	0.45 ± 0.02	0.40 ± 0.07	0.45 ± 0.02	0.45 ± 0.02	0.13 ± 0.15
`example2s.B`	0.50 ± 0.00	0.44 ± 0.10	0.50 ± 0.00	0.50 ± 0.01	0.13 ± 0.15
`example2s.C`	0.44 ± 0.03	0.39 ± 0.07	0.44 ± 0.03	0.44 ± 0.02	0.13 ± 0.14
`example3.A`	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.01	0.08 ± 0.00
`example3.B`	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.08 ± 0.00
`example3.C`	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.08 ± 0.00
`example3s.A`	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.50 ± 0.00	0.08 ± 0.00
`example3s.B`	0.50 ± 0.01	0.50 ± 0.01	0.50 ± 0.01	0.50 ± 0.01	0.08 ± 0.00
`example3s.C`	0.50 ± 0.01	0.50 ± 0.01	0.50 ± 0.01	0.50 ± 0.01	0.08 ± 0.00

Table 1: Errors for all algorithms, datasets, and environments (A, B, C) on the test splits. Example 1 measures error using the squared loss; all others use the 0-1 loss.

- Invariant Risk Minimization [Arjovsky et al., 2019, IRMv1] finds a representation of the features such that the optimal classifier, on top of that representation, is the identity function for all environments.
- Inner-environmental Gradient Alignment [Koyama and Yamaguchi, 2020, IGA] minimizes the error on the training splits while reducing the variance of the gradient of the loss at each environment.
- AND-mask [Parascandolo et al., 2020] minimizes the error on the training splits by updating the model on those directions where the sign of the gradient of the loss is the same for most environments.
- Oracle is a version of ERM where all data splits contain randomized $x^e_{\texttt{spu}}$, and therefore are trivial to ignore. The purpose of this method is to understand the achievable upper bound in performance in our problems.

For each algorithm, we run a random hyper-parameter search of 20 trials. We trained each algorithm and hyper-parameter trial on the train splits of all environments, for 10^4 full-batch Adam [Kingma and Ba, 2015] updates. We choose the hyper-parameters trial that minimizes the error on the validation splits of all environments. Finally, we report the error of these selected models on the test splits. To provide error bars, this entire process, including data sampling, is repeated 50 times. We refer the reader to our code to learn about the hyper-parameter search distributions for each algorithm.

Results Table 1 shows that no method is able to achieve a performance close to the Oracle on any of the proposed problems. The only exception is IRMv1 on `example1`. This illustrates that current causal learning algorithms are unable to capture invariances, even in low-dimensional linear problems.

4 Outlook

We propose a battery of "unit-tests" to evaluate the out-of-distribution capabilities of machine learning algorithms. While admittedly synthetic, our collection of problems attempts to cover a wide range of challenging distributional discrepancies that may arise across training and testing conditions. By extending this set of problems, we can learn about the strength and shortcomings of new algorithms in a transparent and standardized manner.

References

Michael A Alcorn, Qi Li, Zhitao Gong, Chengfei Wang, Long Mai, Wei-Shinn Ku, and Anh Nguyen. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. *CVPR*, 2019.

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. *ECCV*, 2018.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *arXiv preprint arXiv:2004.07780*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

Masanori Koyama and Shoichiro Yamaguchi. Out-of-distribution generalization with maximal invariant predictor. *arXiv preprint arXiv:2008.01883*, 2020.

Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. *ECCV*, 2018.

Giambattista Parascandolo, Alexander Neitz, Antonio Orvieto, Luigi Gresele, and Bernhard Schölkopf. Learning explanations that are hard to vary. *arXiv preprint arXiv:2009.00329*, 2020.

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

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *arXiv preprint arXiv:1501.01332*, 2015.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference*. 2017.

Amir Rosenfeld, Richard Zemel, and John K Tsotsos. The elephant in the room. *arXiv preprint arXiv:1808.03305*, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Vladimir Vapnik. Statistical learning theory wiley. *New York*, 1998.