Invariant Risk Minimization Games

Kartik Ahuja [1] **Karthikeyan Shanmugam** [1] **Kush R. Varshney** [1] **Amit Dhurandhar** [1]

Abstract

The standard risk minimization paradigm of machine learning is brittle when operating in environments whose test distributions are different from the training distribution due to spurious correlations. Training on data from many environments and finding *invariant* predictors reduces the effect of spurious features by concentrating models on features that have a causal relationship with the outcome. In this work, we pose such invariant risk minimization as finding the Nash equilibrium of an ensemble game among several environments. By doing so, we develop a simple training algorithm that uses best response dynamics and, in our experiments, yields similar or better empirical accuracy with much lower variance than the challenging bi-level optimization problem of Arjovsky et al. (2019). One key theoretical contribution is showing that the set of Nash equilibria for the proposed game are equivalent to the set of invariant predictors for any finite number of environments, even with nonlinear classifiers and transformations. As a result, our method also retains the generalization guarantees to a large set of environments shown in Arjovsky et al. (2019). The proposed algorithm adds to the collection of successful game-theoretic machine learning algorithms such as generative adversarial networks.

1. Introduction

The annals of machine learning are rife with embarrassing examples of spurious correlations that fail to hold outside a specific training (and identically distributed test) distribution. Beery et al. (2018) trained a convolutional neural network (CNN) to classify camels from cows. The training dataset had one source of bias, i.e., most of the pictures of cows had green pastures, while most pictures of camels were in deserts. The CNN picked up the spurious correlation, i.e., it associated green pastures with cows and failed to classify pictures of cows on sandy beaches correctly. In another case, a neural network used a brake light indicator to continue applying brakes, which was a spurious correlation in the training data (de Haan et al., 2019); the list of such examples goes on.

To address the problem of models inheriting spurious correlations, Arjovsky et al. (2019) show that one can exploit the varying degrees of spurious correlation naturally present in data collected from multiple data sources to learn robust predictors. The authors propose to find a representation Φ such that the optimal classifier given Φ is invariant across training environments. This formulation leads to a challenging bi-level optimization, which the authors relax by fixing a simple linear classifier and learning a representation Φ such that the classifier is "approximately locally optimal" in all the training environments.

In this work, we take a very different approach. We create an *ensemble* of classifiers with each environment controlling one component of the ensemble. Each environment uses the entire ensemble to make predictions. We let all the environments play a *game* where each environment's action is to decide its contribution to the ensemble such that it minimizes its risk. Remarkably, we establish that the set of predictors that solve the *ensemble game* is equal to the set of invariant predictors across the training environments; this result holds for a large class of nonlinear classifiers.

This brings us to the question: how do we solve the game? We use classic best response dynamics (Fudenberg & Levine, 1998), which has a very simple implementation. Each environment periodically takes its turn and moves its classifier in the direction that minimizes the risk specific to its environment. Empirically, we establish that the invariant predictors found by our approach lead to better or comparable performance with much lower standard deviation than Arjovsky et al. (2019) on several different datasets. A nice consequence of our approach is we do not restrict classifiers to be linear, which was emphasized as an important direction for future work by Arjovsky et al. (2019).

Broadly speaking, we believe that the game-theoretic perspective herein can open up a totally new paradigm to address the problem of invariance.

[1]IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY. Correspondence to: Kartik Ahuja <kartik.ahuja@ibm.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

2. Related Work

2.1. Invariance Principles in Causality

The invariant risk minimization formulation of Arjovsky et al. (2019) is the most related work, and is motivated from the theory of causality and causal Bayesian networks (CBNs) (Pearl, 1995). A variable y is caused by a set of non-spurious actual causal factors $x_{\mathrm{Pa}(y)}$ if and only if in all environments where y has not been intervened on, the conditional probability $P(y|x_{\mathrm{Pa}(y)})$ remains invariant. This is called the *modularity condition* (Bareinboim et al., 2012). Related and similar notions are the *independent causal mechanism principle* (Schölkopf et al., 2012; Janzing & Schölkopf, 2010; Janzing et al., 2012) and the *invariant causal prediction principle* (Peters et al., 2016; Heinze-Deml et al., 2018). These principles imply that if all the environments (train and test) are modeled by interventions that do not affect the causal mechanism of target variable y, then a classifier conservatively trained on the transformation that involves the causal factors ($\Phi(x) = x_{\mathrm{Pa}(y)}$) to predict y is robust to unseen interventions.

In general, for finite sets of environments, there may be other invariant predictors. If one has information about the CBN structure, one can find invariant predictors that are maximally predictive using conditional independence tests and other graph-theoretic tools (Magliacane et al., 2018; Subbaswamy et al., 2019).

The above works select subsets of features, primarily using conditional independence tests, that make the optimal classifier trained on the selected features be invariant. Arjovsky et al. (2019) give an optimization-based reformulation of this invariance that facilitates searching over transformations in a continuous space, making their work widely applicable in areas such as computer vision where the causal features are latent (see Figure 6 in Arjovsky et al. (2019)).

2.2. Sample Reweighting, Domain Adaptation, and Robust Optimization

Statistical machine learning has dealt with the distribution shift between the training distribution and test distribution in a number of ways. Conventional approaches are sample weighting, domain adaptation, and robust optimization. Importance weighting or more generally sample weighting attempts to match test and train distributions by reweighting samples (Shimodaira, 2000; Sugiyama et al., 2008; Gretton et al., 2009; Zhao et al., 2018). It typically assumes that the probability of labels given all covariates does not shift, and in more general cases, requires access to test labels. Domain adaptation tries to find a representation Φ whose distribution is invariant across source and target domains (Ajakan et al., 2014; Ben-David et al., 2007; Glorot et al., 2011; Ganin et al., 2016). Domain adaptation is known to have serious limitations even when the marginal distribution of labels shift across environments (Zhao et al., 2019; Johansson et al., 2019). When only training data sources are given, robust optimization techniques find the worst case loss over all possible convex combinations of the training sources (Mohri et al., 2019; Hoffman et al., 2018; Lee & Raginsky, 2018; Duchi et al., 2016). This assumes that the test distribution is within the convex hull of training distributions, which is not true in many settings.

3. Preliminaries

3.1. Game Theory Concepts

We begin with some basic concepts from game theory (Fudenberg & Tirole, 1991) that we will use. Let $\Gamma = (N, \{S_i\}_{i\in N}, \{u_i\}_{i\in N})$ be the tuple representing a standard normal form game, where N is the finite set of players. Player $i \in N$ takes actions from a strategy set S_i. The utility of player i is $u_i : S \to \mathbb{R}$, where we write the joint set $S = \Pi_{i\in N} S_i$. The joint strategy of all the players is given as $s \in S$, the strategy of player i is s_i and the strategy of the rest of players is $s_{-i} = (s_{i'})_{i'\neq i}$. If the set S is finite, then we call the game Γ a *finite game*. If the set S is uncountably infinite, then the game Γ is a *continuous game*.

Nash equilibrium in pure strategies. A strategy s^* is said to be a pure strategy Nash equilibrium (NE) if it satisfies

$$u_i(s_i^*, s_{-i}^*) \geq u_i(k, s_{-i}^*), \forall k \in S_i, \forall i \in N$$

3.2. Invariant Risk Minimization

We describe the invariant risk minimization (IRM) of Arjovsky et al. (2019). Consider datasets $\{(x_i^e, y_i^e)\}_{i=1}^{n_e}$ from multiple training environments $e \in \mathcal{E}_{tr}$. The feature value $x_i^e \in \mathcal{X}$ and the corresponding labels $y_i^e \in \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{Y} \subseteq \mathbb{R}^k$.[1] Define a predictor $f : \mathcal{X} \to \mathbb{R}^k$.

The goal of IRM is to use these multiple datasets to construct a predictor f that performs well across many unseen environments $\mathcal{E}_{all}$, where $\mathcal{E}_{all} \supseteq \mathcal{E}_{tr}$. Define the risk achieved by f in environment e as $R^e(f) = \mathbb{E}_{X^e,Y^e}\left[\ell(f(X^e), Y^e)\right]$, where ℓ is the loss when $f(X)$ is the predicted value and Y is the corresponding label. To assume that f maps to real values is not restrictive; for instance, in a k-class classification problem, the output of the function f is the score for each class, which can be converted into a hard label by selecting the class with the highest score.

Invariant predictor: We say that a data representation $\Phi : \mathcal{X} \to \mathcal{Z} \subseteq \mathbb{R}^d$ elicits an invariant predictor $w \circ \Phi$ across environments $e \in \mathcal{E}$ if there is a classifier $w : \mathcal{Z} \to \mathbb{R}^k$ that achieves the minimum risk for all the environments

[1] The setup applies to both continuous and categorical data. If any feature or label is categorical, we one-hot encode it.

$w \in \arg\min_{\bar{w} \in \mathcal{H}_w} R^e(\bar{w} \circ \Phi),\ \forall e \in \mathcal{E}$. The set of all the mappings Φ is given as $\mathcal{H}_\Phi$ and the set of all the classifiers is given as $\mathcal{H}_w$. IRM may be phrased as the following constrained optimization problem (Arjovsky et al., 2019):

$$\begin{aligned} &\min_{\Phi \in \mathcal{H}_\Phi, w \in \mathcal{H}_w} \sum_{e \in \mathcal{E}_{tr}} R^e(w \circ \Phi) \\ &\text{s.t. } w \in \arg\min_{\bar{w} \in \mathcal{H}_w} R^e(\bar{w} \circ \Phi),\ \forall e \in \mathcal{E}_{tr}. \end{aligned} \tag{1}$$

If $w \circ \Phi$ satisfies the constraints above, then it is an invariant predictor across the training environments $\mathcal{E}_{tr}$.

Intuition behind IRM optimization in equation (1) We use a simplified version of the model described by Peters et al. (2016). In each environment e, the random variable $X^e = [X_1^e, ..., X_n^e]$ corresponds to the feature vector and Y^e corresponds to the label. The data for each environment is generated by i.i.d. sampling (X^e, Y^e) from the following generative model. Assume a subset $S^* \subset \{1, ..., n\}$ is causal for the label Y^e. For each environment $e \in \mathcal{E}_{all}$, X^e has an arbitrary distribution and

$$Y^e \leftarrow g(X_{S^*}^e) + \tilde{\epsilon}^e \tag{2}$$

where $X_{S^*}^e$ is the vector X^e with indices in S^*, $g : \mathbb{R}^{|S^*|} \to \mathbb{R}$ is a function to describe the conditional expectation and $\epsilon^e \sim F^e$, $\mathbb{E}[\tilde{\epsilon}_e] = 0$, $\tilde{\epsilon}^e \perp X_{S^*}^e$. Let ℓ be the squared error loss function. We fix the representation $\Phi^*(X^e) = X_{S^*}^e$. With Φ^* as the representation, the optimal classifier w among all the functions is $g(X_{S^*}^e)$ (this follows from the generative model). Since the optimal classifier does not vary across environments, $w_*^e \circ \Phi^* = g$ is an invariant predictor across all $\mathcal{E}_{all}$ (assume $g \in \mathcal{H}_w$). Define $X_{S^\circ}^e$ as the remaining feature values that are not causal of Y^e. If Φ does not focus on $\mathcal{S}^*$ and some of the variables in $X_{S^\circ}^e$ are a part of the Markov blanket of Y^e (Pearl, 2014), then the optimal classifier may not be the same across environments and thus invariance won't be achieved. By solving (1) across training environments, IRM hopes to arrive at g, which will generalize well across all the test environments $\mathcal{E}_{all}$.

Define the set of representations and the corresponding classifiers, (Φ, w) that satisfy the constraints in the above optimization problem (1) as $\mathcal{S}^{\mathsf{IV}}$, where IV stands for invariant. Also, separately define the set of invariant predictors $w \circ \Phi$ as $\hat{\mathcal{S}}^{\mathsf{IV}} = \{w \circ \Phi \,|(\Phi, w) \in \mathcal{S}^{\mathsf{IV}}\}$.

Remark. The sets $\mathcal{S}^{\mathsf{IV}}$, $\hat{\mathcal{S}}^{\mathsf{IV}}$ depend on the choice of classifier class $\mathcal{H}_w$ and representation class $\mathcal{H}_\Phi$. We avoid making this dependence explicit until later sections.

Members of $\mathcal{S}^{\mathsf{IV}}$ are equivalently expressed as solutions to

$$R^e(w \circ \Phi) \leq R^e(\bar{w} \circ \Phi),\ \forall \bar{w} \in \mathcal{H}_w,\ \forall e \in \mathcal{E}_{tr}. \tag{3}$$

The main generalization result (Theorem 9) in Arjovsky et al. (2019) states that if representations and classifiers are from the class of linear models, i.e., $\Phi \in \mathcal{H}_\Phi = \mathbb{R}^{n \times n}$ (representation output for input x is Φx) and $w \in \mathcal{H}_w = \mathbb{R}^{n \times 1}$ (classifier output for input z is $w^\mathsf{T} z$), then under certain conditions on the data generation process and training environments $\mathcal{E}_{tr}$, the solution to (3) remains invariant in $\mathcal{E}_{all}$ (we use the same setup in our Theorem 2 as well).

4. Ensemble Invariant Risk Minimization Games

4.1. Game-Theoretic Reformulation

Optimization problem (1) can be quite challenging to solve. We introduce an alternate characterization based on game theory to solve it. We endow each environment with its own classifier $w^e \in \mathcal{H}_w$. We use a simple ensemble to construct an overall classifier $w^{av} : \mathcal{Z} \to \mathbb{R}^k$ defined as $w^{av} = \frac{1}{|\mathcal{E}_{tr}|} \sum_{q=1}^{|\mathcal{E}_{tr}|} w^q$, where for each $z \in \mathcal{Z}$, $w^{av}(z) = \frac{1}{|\mathcal{E}_{tr}|} \sum_{q=1}^{|\mathcal{E}_{tr}|} w^q(z)$. (The av stands for average.) Consider the example of binary classification with two environments $\{e_1, e_2\}$; $w^e = [w_1^e, w_2^e]$ is the classifier of environment e, where each component is the score for each class. We define the component j of the ensemble classifier w^{av} as $w_j^{av} = \frac{w_j^{e_1} + w_j^{e_2}}{2}$. These scores are input to a softmax; the final probability assigned to class j for an input z is $\frac{e^{w_j^{av}(z)}}{e^{w_1^{av}(z)} + e^{w_2^{av}(z)}}$.

We require all the environments to use this ensemble w^{av}. We want to solve the following new optimization problem.

$$\begin{aligned} &\min_{\Phi \in \mathcal{H}_\Phi, w^{av} \in \mathcal{H}_w} \sum_{e \in \mathcal{E}_{tr}} R^e(w^{av} \circ \Phi) \\ &\text{s.t. } w^e \in \arg\min_{\bar{w}^e \in \mathcal{H}_w} R^e\left(\frac{1}{|\mathcal{E}_{tr}|}\left[\bar{w}^e + \sum_{q \neq e} w^q\right] \circ \Phi\right),\ \forall e \in \mathcal{E}_{tr} \end{aligned}$$

We can equivalently restate the above as:

$$\begin{aligned} &\min_{\Phi \in \mathcal{H}_\Phi, w^{av}} \sum_{e \in \mathcal{E}_{tr}} R^e(w^{av} \circ \Phi) \\ &\text{s.t. } R^e\left(\frac{1}{|\mathcal{E}_{tr}|}\left[w^e + \sum_{q \neq e} w^q\right] \circ \Phi\right) \\ &\leq R^e\left(\frac{1}{|\mathcal{E}_{tr}|}\left[\bar{w}^e + \sum_{q \neq e} w^q\right] \circ \Phi\right)\ \forall \bar{w}^e \in \mathcal{H}_w\ \forall e \in \mathcal{E}_{tr} \end{aligned} \tag{4}$$

What are the advantages of this formulation (4)?

- Using the ensemble automatically enforces that the

same classifier is used across the environments.
- Each environment is free to select the classifier w^e from the entire set $\mathcal{H}_w$, unlike in (1), where all environments' choices are required to be the same.
- The constraints in (4) are equivalent to the set of pure NE of a game that we define next.

The game is played between $|\mathcal{E}_{tr}|$ players, with each player corresponding to an environment e. The set of actions of the environment e are $w^e \in \mathcal{H}_w$. At the start of the game, a representation Φ is selected from the set $\mathcal{H}_\Phi$, which is observed by all the environments. The utility function for an environment e is defined as $u_e[w^e, w^{-e}, \Phi] = -R^e(w^{av}, \Phi)$, where $w^{-e} = \{w^q\}_{q \neq e}$ is the set of choices of all environments but e. We call this game Ensemble Invariant Risk Minimization (EIRM) and express it as a tuple

$$\Gamma^{\mathsf{EIRM}} = \Big(\mathcal{E}_{tr}, \mathcal{H}_\Phi, \{\mathcal{H}_w\}_{q=1}^{|\mathcal{E}_{tr}|}, \{u_e\}_{e \in \mathcal{E}_{tr}}\Big).$$

We represent a pure NE as a tuple $\Big(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}\Big)$. Since each pure NE depends on Φ, we include it as a part of the tuple.[2] We define the set of pure NE as $\mathcal{S}^{\mathsf{EIRM}}$. We construct a set of all the ensemble predictors constructed from NE as[3]

$$\hat{\mathcal{S}}^{\mathsf{EIRM}} = \Big\{ \Big[\frac{1}{|\mathcal{E}_{tr}|} \sum_{q=1}^{|\mathcal{E}_{tr}|} w^q \Big] \circ \Phi \mid (\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_t|}) \in \mathcal{S}^{\mathsf{EIRM}} \Big\}.$$

Members of $\mathcal{S}^{\mathsf{EIRM}}$ are equivalently expressed as the solutions to

$$u_e[w^e, w^{-e}, \Phi] \geq u_e[\bar{w}^e, w^{-e}, \Phi], \ \forall w^e \in \mathcal{H}_w, \forall e \in \mathcal{E}_{tr}. \tag{5}$$

If we replace $u_e[w^e, w^{-e}, \Phi]$ with $-R^e(w^{av}, \Phi)$, we obtain the inequalities in (4). So far we have defined the game and given its relationship to the problem in (4).

Overview of the Results In Sections 4.2 and 4.3 we will discuss the main theoretical results of this work. Here we give a brief preview of them.

- In Theorem 1 and Corollary 1 we establish equivalence between the predictors obtained using NE of EIRM game $\mathcal{S}^{\mathsf{EIRM}}$ and invariant predictors $\mathcal{S}^{\mathsf{IV}}$. We establish this equivalence for a large class of representations and classifiers, where both can be nonlinear.
- In Theorem 2, we borrow the generalization result from (Arjovsky et al., 2019) and show that same generalization guarantees continue to hold for our setting. Following (Arjovsky et al., 2019), we assume both classifiers and representations are linear.
- In Theorem 3, we discuss the role of representation and how in some cases we can reduce the computational expense that one may incur in searching for the representations. We establish this result for a large class of classifiers and invertible representations, where both can be nonlinear.
- In Theorem 4, we discuss the existence of both the Nash equilibria of EIRM game and the invariant predictors. We restrict the classifiers to be linear but representations may be nonlinear. In the supplement, we extend the result to nonlinear classifiers.

[2] We can also express each environment's action as a mapping from $\pi : \mathcal{H}_\Phi \to \mathcal{H}_w$ but we don't to avoid complicated notation.

[3] We don't double count compositions leading to the same predictor.

4.2. Equivalence Between NE and Invariant Predictors

What is the relationship between the predictors obtained from NE $\hat{\mathcal{S}}^{\mathsf{EIRM}}$ and invariant predictors $\hat{\mathcal{S}}^{\mathsf{IV}}$?

Remarkably, these two sets are the same under very mild conditions. Before we show this result, we establish a stronger result and this result will follow from it.

We use the set $\mathcal{S}^{\mathsf{EIRM}}$ to construct a new set. To each tuple $\Big(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}\Big) \in \mathcal{S}^{\mathsf{EIRM}}$ augment the ensemble classifier $w^{av} = \frac{1}{|\mathcal{E}_{tr}|} \sum_{q=1}^{|\mathcal{E}_{tr}|} w^q$ to get $\Big(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}, w^{av}\Big)$. We call the set of these new tuples $\tilde{\mathcal{S}}^{\mathsf{EIRM}}$.

We use the set $\mathcal{S}^{\mathsf{IV}}$ to construct a new set. Consider an element $(\Phi, w) \in \mathcal{S}^{\mathsf{IV}}$. We define a decomposition for w in terms of the environment-specific classifiers as follows: $w = \frac{1}{|\mathcal{E}_{tr}|} \sum_{q=1}^{|\mathcal{E}_{tr}|} w^q$, where $w^q \in \mathcal{H}_w$. $w^q = w, \forall q \in \mathcal{E}_{tr}$ is one trivial decomposition. We use each such decomposition and augment the tuple to obtain $\Big(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}, w\Big)$. We call this set of new tuples $\tilde{\mathcal{S}}^{\mathsf{IV}}$.

Both the sets $\tilde{\mathcal{S}}^{\mathsf{IV}}$ and $\tilde{\mathcal{S}}^{\mathsf{EIRM}}$ consist of tuples of representation, set of environment specific classifiers, and the ensemble classifier. We ask an even more interesting question than the one above. Is the set of representations, environment specific classifiers, and the ensembles found by playing EIRM (5) or solving IRM (3) the same? If these two sets are equal, then equality between $\hat{\mathcal{S}}^{\mathsf{EIRM}}$ and $\hat{\mathcal{S}}^{\mathsf{IV}}$ follows trivially.

We state the only assumption we need.

Assumption 1. ***Affine closure:*** *The class of functions $\mathcal{H}_w$ is closed under the following operations.*

- *Finite sum: If $w_1 \in \mathcal{H}_w$ and $w_2 \in \mathcal{H}_w$, then $w_1 + w_2 \in \mathcal{H}_w$, where for every $z \in \mathcal{Z}$, $(w_1 + w_2)(z) = w_1(z) + w_2(z)$*
- *Scalar multiplication: For any $c \in \mathbb{R}$ and $w \in \mathcal{H}_w$, $cw \in \mathcal{H}_w$, where for every $z \in \mathcal{Z}$, $(cw)(z) = c \times w(z)$*

The addition of the functions and scalar multiplication are defined in a standard pointwise manner. Therefore, the class $\mathcal{H}_w$ also forms a vector space.

Examples of functions that satisfy affine closure. Linear classifiers, kernel based classifiers (Hofmann et al., 2008) (functions in RKHS space), ensemble models with arbitrary number of weak learners (Freund et al., 1999), functions in L^p space (Ash & Doléans-Dade, 2000), ReLU networks with arbitrary depth. We provide the justification for each of these functions in the supplement. We now state the main result.

Theorem 1. *If Assumption 1 holds, then* $\tilde{\mathcal{S}}^{\mathsf{IV}} = \tilde{\mathcal{S}}^{\mathsf{EIRM}}$

The proofs of all the results are in the supplement.

Corollary 1. *If Assumption 1 holds, then* $\hat{\mathcal{S}}^{\mathsf{IV}} = \hat{\mathcal{S}}^{\mathsf{EIRM}}$

Significance of Theorem 1 and Corollary 1

i. **Computational** This equivalence permits computational tools from game theory to find NE of the EIRM game and the invariant predictors. (See Algorithm 1)

ii. **Theoretical** This equivalence permits to use game theory to analyze the solutions of the EIRM game and understand the invariant predictors. (See Theorem 3)

iii. **Generalization** In Theorem 9 (Arjovsky et al., 2019), it was shown for linear classifiers and linear representations that the invariant predictors generalize to a large set of unseen environments under certain conditions. Since our equivalence (Theorem 1) holds for linear classifiers (but is even broader), the same generalization holds for predictors obtained from EIRM game. Although the result follows straightaway from the equivalence in Theorem 1, we still state it formally next for completeness.

We describe the generative model for the next theorem. For environment e, $Y^e \leftarrow Z_1^{e\mathsf{T}}\gamma + \epsilon^e$, where ϵ^e is independent of Z_1^e, $Z_1^e \in \mathbb{R}^c$ and $\gamma \in \mathbb{R}^c$. We observe X^e, which is a scrambled version of Z_1^e and Z_2^e, where Z_2^e can be correlated with both Z_1^e and ϵ^e. We use Assumption 8 from (Arjovsky et al., 2019). We restate it below.

Assumption 2. *A set of environments $\mathcal{E}_{tr}$ lie in the linear general position of degree r if $|\mathcal{E}_{tr}| \geq n - r + \frac{n}{r}$ for some $r \in \mathbb{N}$ and for all non-zero $x \in \mathbb{R}^n$*

$$\dim\Big(\mathsf{span}\Big(\big\{\mathbb{E}_{X^e}[X^eX^{e\mathsf{T}}]x - \mathbb{E}_{X^e,\epsilon^e}[X^e\epsilon^e]\big\}_{e\in\mathcal{E}_{tr}}\Big)\Big) > n - r$$

In the next theorem, we consider linear representations, i.e., $\mathcal{H}_\Phi = \mathbb{R}^{n\times n}$ and linear classifiers, i.e., $\mathcal{H}_w = \mathbb{R}^{n\times 1}$ and $w \circ \Phi = w^{\mathsf{T}}\Phi$.

Theorem 2. *For each environment $e \in \mathcal{E}_{all}$ we assume*

$$\begin{aligned} &Y^e \leftarrow Z_1^{e\mathsf{T}}\gamma + \epsilon^e, \quad Z_1^e \perp \epsilon^e, \quad \mathbb{E}[\epsilon^e] = 0 \\ &X^e \leftarrow S(Z_1^e, Z_2^e) \end{aligned} \tag{6}$$

Here $\gamma \in \mathbb{R}^c$, $Z_1^e \in \mathbb{R}^c$, $Z_2^e \in \mathbb{R}^q$, $S \in \mathbb{R}^{n\times(c+q)}$. Assume that Z_1 is invertible component of S, i.e., $\exists\, \tilde{S} \in \mathbb{R}^{c\times n}$ such that $\tilde{S}(S(z_1, z_2)) = z_1$ for all $z_1 \in \mathbb{R}^c$ and $z_2 \in \mathbb{R}^q$. Let $\Phi \in \mathbb{R}^{n\times n}$ have rank r. If at least $n - r + \frac{n}{r}$ training environments $\mathcal{E}_{tr} \subseteq \mathcal{E}_{all}$ lie in linear general position of degree r, then any predictor obtained from EIRM *game over the training environments in $\hat{\mathcal{S}}^{\mathsf{EIRM}}$ is invariant across all the testing environments $\mathcal{E}_{all}$.*

The above theorem establishes generalization guarantees for predictors obtained from the NE of the EIRM game and the proof follows from Theorem 1 above and proof of Theorem 9 in (Arjovsky et al., 2019).

Role of representation Φ. We investigate the scenario when we fix Φ to the identity mapping; this will motivate one of our approaches. Define the set $\hat{\mathcal{S}}^{\mathsf{EIRM}}(\Phi)$ as the set of ensemble predictors arrived at by playing the EIRM game using a fixed representation representation Φ.[4] Similarly, we define a set $\hat{\mathcal{S}}^{\mathsf{IV}}(\Phi)$ as the set of invariant predictors derived using the representation Φ. From Theorem 1, it follows that $\hat{\mathcal{S}}^{\mathsf{EIRM}}(\Phi) = \hat{\mathcal{S}}^{\mathsf{IV}}(\Phi)$. We modify some of the earlier notations for results to follow. The set of predictors that result from the EIRM game $\hat{\mathcal{S}}^{\mathsf{EIRM}}$ and the sets of invariant predictors $\hat{\mathcal{S}}^{\mathsf{IV}}$ are defined for a family of maps Φ with co-domain $\mathcal{Z}$. We make the co-domain $\mathcal{Z}$ explicit in the notation. We write $\hat{\mathcal{S}}^{\mathsf{EIRM}}_{\mathcal{Z}}$ for $\hat{\mathcal{S}}^{\mathsf{EIRM}}$ and $\hat{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$ for $\hat{\mathcal{S}}^{\mathsf{IV}}$.

Assumption 3. $\Phi \in \mathcal{H}_\Phi$ *satisfies the following*

- *Bijective:* $\exists\ \Phi^{-1} : \mathcal{Z} \to \mathcal{X}$ *such that* $\forall x \in \mathcal{X}$, $\Big(\Phi^{-1} \circ \Phi\Big)(x) = x$, *and* $\forall z \in \mathcal{Z}\ \Big(\Phi \circ \Phi^{-1}\Big)(z) = z$. *Both $\mathcal{X}$ and $\mathcal{Z}$ are subsets of* $\mathbb{R}^n$
- Φ *is differentiable and Lipschitz continuous.*

$L^p(\mathcal{Z})$: set of functions $f : \mathcal{Z} \to \mathbb{R}$ s.t. $\int_{\mathcal{Z}} |f|^p d\mu < \infty$

Assumption 4. $\mathcal{H}_w = L^p(\mathcal{Z})$.

Define a subset $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}} \subseteq \hat{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$ consisting of invariant predictors that are in $L^p(\mathcal{X})$, i.e., $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}} = \{u \mid u \in \hat{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}} \text{ and } u \in L^p(\mathcal{X})\}$. Let $\Phi = \mathsf{I}$, where $\mathsf{I} : \mathcal{X} \to \mathcal{X}$ is the identity mapping. Following the above notation, the set of invariant predictors and the set of ensemble predictors obtained from NE are $\hat{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{X}}(\mathsf{I})$ and $\hat{\mathcal{S}}^{\mathsf{EIRM}}_{\mathcal{X}}(\mathsf{I})$ respectively.

Theorem 3. *If Assumptions 3 and 4 are satisfied and $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}}$ is non-empty, then* $\bar{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{Z}} = \hat{\mathcal{S}}^{\mathsf{IV}}_{\mathcal{X}}(\mathsf{I}) = \hat{\mathcal{S}}^{\mathsf{EIRM}}_{\mathcal{X}}(\mathsf{I})$

Significance of Theorem 3. If we fix the representation to identity and play the EIRM game, then it is sufficient to recover all the invariant predictors (with bounded L^p norm) that can be obtained using all the representations $\Phi \in \mathcal{H}_\Phi$. Therefore, we can simply fix $\Phi = \mathsf{I}$ and use game-theoretic algorithms for learning equilibria.

[4] $\cup_\Phi \hat{\mathcal{S}}^{\mathsf{EIRM}}(\Phi) = \hat{\mathcal{S}}^{\mathsf{EIRM}}$

4.3. Existence of NE of Γ^{EIRM} and Invariant Predictors

In this section, we argue that there are many settings when both invariant predictors and the NE exist. Recall that in the example described in Section 3.2 based on the generative model in equation (2), we already showed existence of invariant predictor as we constructed one. In the same example if we also assume that $\mathcal{H}_w$ is affine closed, then the NE also exist (From Theorem 1). The above claims for existence require us to make assumptions on the data generation process. Next, we discuss the existence with no assumptions on the data generation process.

Assumption 5.
- *$\mathcal{H}_w$ is a class of linear models, i.e. $w \in \mathcal{H}_w \subseteq \mathbb{R}^{d\times 1}$ and classifier output for input z is $w^\mathsf{T} z$. $\mathcal{H}_w$ is a closed, bounded and convex. The interior of $\mathcal{H}_w$ is non-empty.*
- *The loss function $\ell(w^\mathsf{T} z, Y)$, where $Y \in \mathbb{R}$ is the label, is convex and continuous in w. For e.g., if loss is cross-entropy for binary classification or loss is mean squared error for regression, then this assumption is automatically satisfied.*

Theorem 4. *If Assumption 5 is satisfied, then a pure strategy Nash equilibrium of the game Γ^{EIRM} exists, i.e., $\mathcal{S}^{\text{EIRM}}$ is not empty. Suppose there exists a $\left(\Phi, \{w^q\}_{q=1}^{|\mathcal{E}_{tr}|}\right) \in \mathcal{S}^{\text{EIRM}}$ such that$\forall q \in \mathcal{E}_{tr}$ w^q is in the interior of $\mathcal{H}_w$, then the corresponding ensemble predictor $\frac{1}{|\mathcal{E}^{tr}|}\sum_{q=1}^{|\mathcal{E}_{tr}|} w^q \circ \Phi$ is invariant across all the training environments $\mathcal{E}_{tr}$.*

The family $\mathcal{H}_w$ of bounded linear functions does not satisfy affine closure, which is why existence of NE does not immediately imply the existence of invariant predictor (from Theorem 1). However, if the solution is in the interior of $\mathcal{H}_w$, then it is the globally optimal solution among all the linear functions, which in fact actually satisfy affine closure. As a result, in this case the invariant predictor also exists.

Significance of Theorem 4 Our approach is based on finding the NE. Therefore, it is important to understand when the solutions are guaranteed to exist. In the above theorem, we proved the result for linear classifiers only, but there were no assumptions made on the representation class. In the supplement, we discuss extensions to nonlinear classifiers. Following the sufficient condition for existence of invariant predictors, understanding what conditions cause the NEs to be in the interior or on the boundary of $\mathcal{H}_w$ can help further the theory of invariant prediction.

4.4. Algorithms for Finding NE of Γ^{EIRM}

There are different strategies in the literature to compute the equilibrium, such as best response dynamics (BRD) and fictitious play (Fudenberg & Levine, 1998), but none of these strategies are guaranteed to arrive at equilibria in continuous games except for special classes of games (Hofbauer & Sorin, 2006; Barron et al., 2010; Mertikopoulos & Zhou, 2019; Bervoets et al., 2016; Daskalakis et al., 2017). BRD is one the most popular methods given its intuitive and natural structure. The training of GANs also follows an approximate BRD (Goodfellow et al., 2014). BRD is not known to converge to equilibrium in GANs. Instead a modification of it proposed recently, Hsieh et al. (2018) achieves mixed NE. Our game Γ^{EIRM} is a non-zero sum game with continuous actions unlike GANs. Since there are no known techniques that are guaranteed to compute the equilibrium (pure or mixed) for these games, we adopt the classic BRD approach.

In our first approach, we use a fixed representation Φ. Recall in Theorem 3, we showed how just fixing Φ to identity can be a very effective approach. Hence, we can fix Φ to be identity mapping or we can select Φ as some other mapping such as approximation of the map for Gaussian kernel (Rahimi & Recht, 2008). Once we fix Φ, the environments play according to best response dynamics as follows.

- Each environment takes its turn (in a periodic manner with each environment going once) and minimizes its respective objective.
- Repeat this procedure until a certain criterion is achieved, e.g., maximum number of epochs or desired value of training accuracy.

The above approach does not give much room to optimize Φ. We go back to the formulation in (4) and use the upper level optimization objective as a way to guide search for Φ. In this new approach, Φ is updated by the representation learner periodically using the objective in (4) and between two updates of Φ the environments play according to best response dynamics as described above.

We now make assumptions on $\mathcal{H}_w$ and $\mathcal{H}_\Phi$ and give a detailed algorithm (see Algorithm 1) that we use in experiments. We assume that w^e is parametrized by family of neural networks $\theta_w \in \Theta_w$ and Φ is parametrized by family of neural networks $\theta_\Phi \in \Theta_\Phi$. In the Algorithm 1, one of the variables Fixed - Phi (for our first approach) or Variable-Phi is set to true, and then accordingly Φ remains fixed or is updated periodically. In the Algorithm, K is a hyperparameter that dictates how many updates of each environment occur before updating Φ. In Figure 1, we also show an illustration of the best response training when there are two environments and one representation learner.

5. Experiments

5.1. Benchmarks

The most important benchmark for comparison is Arjovsky et al. (2019), which we refer to as IRM in the comparisons. We use the architecture described in their work (details in the supplement). We also compare with

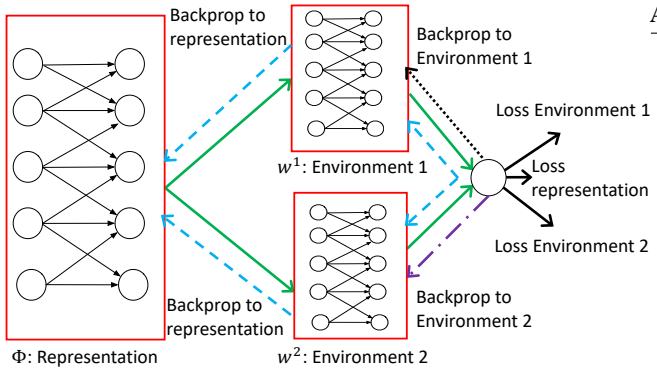

Figure 1. Illustration of best response training with 2 environments and representation learner. Dotted lines for backpropagation and solid lines for forward pass.

Algorithm 1 Best Response Training

Input: Data for each environment and combined data
Initialize: Randomly initialize $\{w^e_{cur}\}_{e=1}^{|\mathcal{E}_{tr}|}$ and Φ_{cur} from $\mathcal{H}_w$ and $\mathcal{H}_\Phi$ respectively
while iter $\leq$ iter$_{\max}$ **do**
  if Fixed-Phi **then**
    $\Phi_{\text{cur}} = \mathsf{I}$
  end if
  if Variable-Phi **then**
    $\Phi_{\text{cur}} = \text{SGD}\Big[\sum_e R^e(w^{av}_{\text{cur}} \circ \Phi_{\text{cur}})\Big]$, SGD[.]: step update using stochastic gradient descent
  end if
  for $p \in \{1, ..K\}$ **do**
    for $e \in \{1, .., |\mathcal{E}_{tr}|\}$ **do**
      $w^e_{\text{cur}} = \text{SGD}\Big[R^e(w^{av}_{\text{cur}} \circ \Phi_{\text{cur}})\Big]$
      $w^{av}_{\text{cur}} = \frac{1}{|\mathcal{E}_{tr}|}\sum_e w^e_{\text{cur}}$
    end for
    iter = iter + 1
  end for
end while

- Variants of empirical risk minimization: ERM on entire training data (ERM), ERM on each environment separately (ERM e refers to ERM trained on environment e), and ERM on data with no spurious correlations.
- Robust min-max training: In this method, we minimize the maximum loss across the multiple environments.

We have two approaches for EIRM games: one that uses a Φ fixed to the identity and the other that uses a variable Φ, which we refer to as the F-IRM and V-IRM game, respectively. The details on architectures, hyperparameters, and optimizers used are in the supplement. The source-code is available at https://github.com/IBM/IRM-games.

5.2. Datasets

Colored MNIST dataset. In Arjovsky et al. (2019), the comparisons were done on a colored digits MNIST dataset. We create the same dataset for our experiments. The task is to classify whether the digit is less than 5 (not including 5) or more than 5. There are three environments (two training containing 30,000 points each, one test containing 10,000 points) We add noise to the preliminary label ($\tilde{y} = 0$ if digit is between 0-4 and $\tilde{y} = 1$ if the digit is between 5-9) by flipping it with 25 percent probability to construct the final labels. We sample the color id z by flipping the final labels with probability p_e, where p_e is 0.2 in the first environment, 0.1 in the second environment, and 0.9 in the third environment. The third environment is the testing environment. We color the digit red if $z = 1$ or green if $z = 0$.

In addition to colored MNIST digits, we also create two other datasets that are inspired from Colored MNIST: Colored Fashion MNIST and Colored Desprites. In these datasets as well the color is spuriously correlated with the label. We also create another dataset: Structured Noise Fashion MNIST. In this dataset, instead of coloring the images to establish spurious correlations, we create small patches of noise at specific locations in the image, where the locations are correlated with the labels (detailed description of the datasets is in the supplement). In all the comparisons, we averaged the performance of the different approaches over ten runs.

5.3. Comparisons

Colored MNIST (Table 1) Standard ERM based approaches, and robust training based approach achieve between 10-15 percent accuracy on the testing set. F-IRM game achieves 59.9 ± 2.7 percent testing accuracy. This implies that the model is not using spurious correlation unlike the ERM based approaches, and robust training based approach, that is present in the color of the digit. F-IRM has a comparable mean and a much lower standard deviation than IRM, which achieves 62.75 ± 9.5 percent. ERM grayscale is ERM on uncolored data, which is why it is better than all. In each Table, we include the optimal performance that is achievable (75 percent train and test).

Colored Fashion MNIST (Table 2) We observe that the V-IRM game performs the best both in terms of the mean and the standard deviation achieving 70.2 ± 1.5 percent.

Colored Desprites (Table 3) We observe that V-IRM game achieves 50.0 ± 0.2 percent while IRM achieves 51.8 ± 6 percent.

Structured Noise Fashion MNIST (Table 4) We observe that F-IRM achieves 62.0 ± 2.0 percent and is comparable with IRM that achievs 63.9 ± 10.9 percent; again observe

Table 1. Colored MNIST: Comparison of methods in terms of training, testing accuracy (mean ± std deviation).

ALGORITHM	TRAIN ACCURACY	TEST ACCURACY
ERM	84.88 ± 0.16	10.45 ± 0.66
ERM 1	84.84 ± 0.21	10.86 ± 0.52
ERM 2	84.95 ± 0.20	10.05 ± 0.23
ROBUST MIN MAX	84.25 ± 0.43	15.24 ± 2.45
F-IRM GAME	**63.37 ± 1.14**	**59.91 ± 2.69**
V-IRM GAME	**63.97 ± 1.03**	**49.06 ± 3.43**
IRM	**59.27 ± 4.39**	**62.75 ± 9.59**
ERM GRAYSCALE	71.81 ± 0.47	71.36± 0.65
OPTIMAL	75	75

Table 2. Colored Fashion MNIST: Comparison of methods in terms of training, testing accuracy (mean ± std deviation).

ALGORITHM	TRAIN ACCURACY	TEST ACCURACY
ERM	83.17 ± 1.01	22.46 ± 0.68
ERM 1	81.33 ± 1.35	33.34 ± 8.85
ERM 2	84.39 ± 1.89	13.16 ± 0.82
ROBUST MIN MAX	82.81 ± 0.11	29.22 ± 8.56
F-IRM GAME	**62.31 ± 2.35**	**69.25 ± 5.82**
V-IRM GAME	**68.96 ± 0.95**	**70.19 ± 1.47**
IRM	**75.01 ± 0.25**	**55.25 ± 12.42**
ERM GRAYSCALE	74.79 ± 0.37	74.67± 0.48
OPTIMAL	75	75

Table 3. Colored Desprites: Comparison of methods in terms of training, testing accuracy (mean ± std deviation).

ALGORITHM	TRAIN ACCURACY	TEST ACCURACY
ERM	85.01 ± 0.03	9.97 ± 0.05
ERM 1	81.33 ± 1.35	33.34 ± 8.85
ERM 2	84.39 ± 1.89	13.16 ± 0.82
ROBUST MIN MAX	84.94 ± 0.09	10.28 ± 0.33
F-IRM GAME	**53.36 ± 1.40**	**48.61 ± 3.06**
V-IRM GAME	**56.31 ± 4.94**	**50.04 ± 0.15**
IRM	**52.67 ± 2.40**	**51.82 ± 5.95**
ERM GRAYSCALE	67.67 ± 0.58	66.97± 0.69
OPTIMAL	75	75

Table 4. Structured Noise Fashion MNIST: Comparison of methods in terms of training, testing accuracy (mean ± std deviation).

ALGORITHM	TRAIN ACCURACY	TEST ACCURACY
ERM	83.49 ± 1.22	20.13 ± 8.06
ERM 1	81.80 ± 1.50	30.94 ± 1.01
ERM 2	84.66 ± 0.40	11.98 ± 0.23
ROBUST MIN MAX	82.78 ± 1.32	25.59 ± 9.14
F-IRM GAME	**51.54 ± 2.96**	**62.03 ± 2.02**
V-IRM GAME	**47.70 ± 1.69**	**61.46 ± 0.53**
IRM	**52.57 ± 9.95**	**63.92 ± 10.95**
ERM NO NOISE	74.79 ± 0.37	74.67± 0.48
OPTIMAL	75	75

that we have a lower standard deviation.

5.4. Analyzing the Experiments

In this section, we use plots of F-IRM game played on Colored Fashion MNIST (plots for both F-IRM and V-IRM on all other datasets are similar and hence convey the same message. We provide them in the supplement). In Figure 2, we show the accuracy of the ensemble model on the entire data and the two environments separately. In the initial stages, the training accuracy increases and eventually it starts to oscillate. Best response dynamics can often oscillate (Herings & Predtetchinski, 2017; Fudenberg & Levine, 1998; Barron et al., 2010). Next, we demistify these oscillations, explain their importance, and discuss how we terminate the procedure.

5.4.1. EXPLAINING THE MECHANISM OF OSCILLATIONS

The oscillation has two states. In the first state, the ensemble model performs well 88 % accuracy. In the second state, the accuracy dips to 75 %. In Figure 3, we plot the correlation between the ensemble model and the color. When the oscillations appear in training accuracy in Figure 2, the correlation also start to oscillate in Figure 3. In the first state when the model performs well, the model is heavily correlated (negative correlation) with the color. In the second state, the model performs worse, observe that the model now has much less correlation (close to zero) with the color. We ask two questions: (i) Why do the oscillations persist in the training accuracy plot (Figure 2) and correlation plot (Figure 3)?, and (ii) How do the oscillations emerge?

Why do the oscillations persist? In our experiments there are two environments, the labels are binary, and we want to maximize the log-likelihood. Let s_j be the score vector from environment j's classifier, p be the softmax of s and $\tilde{y}$ be the one hot encoded vector of labels. The gradient of the log-likelihood w.r.t. the scores given by each model for a certain instance x (see derivation in the supplement) is:

$$\frac{\partial \log(p_y)}{\partial s_j} = \bar{y} - p = \tilde{e}. \tag{7}$$

where $\tilde{e}$ is the error vector. The error $\tilde{e}$ is determined by the both the models (both models impact p), it backpropagates and impacts individual weights. We argue next that the examples over which error occur are very different in the two states and that is the reason for oscillations.

Consider the step when the correlation (absolute value) between the ensemble model and color is high. In this step, it

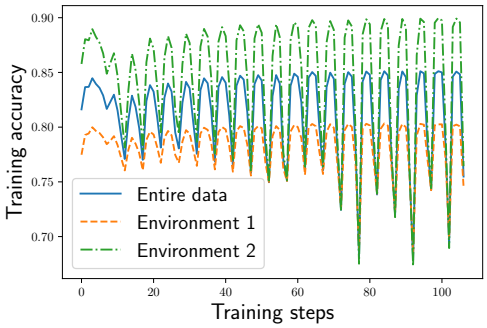

Figure 2. F-IRM, Colored Fashion MNIST: Comparing accuracy of ensemble.

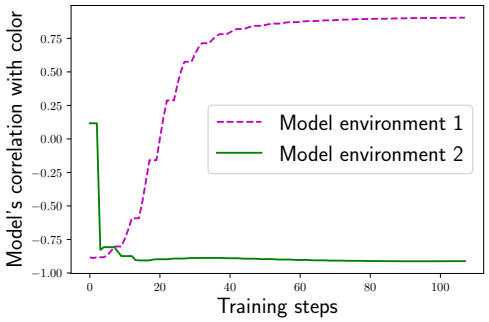

Figure 4. F-IRM, Colored Fashion MNIST: Correlation of the individual models with the color.

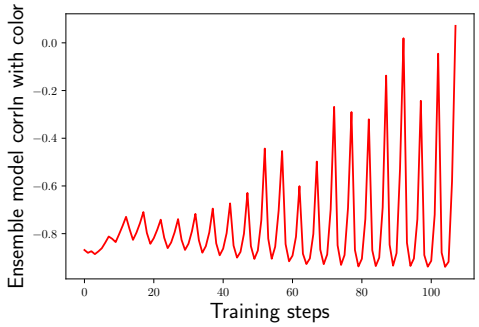

Figure 3. F-IRM, Colored Fashion MNIST: Correlation of the ensemble model with color.

is the turn of Model 1 to train. Observe that the accuracy of the model is high because the ensemble model is exploiting the spurious correlations with the color. We approximate this mathematically. The score from Model j for Label 1 is $s_j^1 - s_j^0 \approx \beta_j^t \phi_j^{nc}(x) + \gamma_j \phi_j^c(x)$, where ϕ_j^{nc} are the features that are not correlated with the color, ϕ_j^c is the indicator of the color. From Figure 4, γ_1 and γ_2 should have opposite signs, i.e. positive and negative respectively. In the current step, γ_2 dominates γ_1, which is why the ensemble model has a heavy negative correlation. The errors (7) that backpropagate come from the examples for which exploiting spurious correlation with color does not work, i.e., the color is not indicative of the digit. During this step Model 1 is trained, backpropagation will change the weights such that γ_1 increases. As a result, the ensemble model's correlation with the color decreases (as we see in Figure 3). In the next step, it is the turn of Model 2 to train. Model 2's environment has more examples than environment 1 where exploiting the color can help improve its accuracy. As a result, error from these examples backpropagate and γ_2 decreases. This brings the ensemble model back to being negatively correlated with colors and also the training accuracy back to where it was approximately. This cycle of push and pull between the models continues.

How do these cycles emerge? The oscillations are weak at the beginning of the training. In the beginning, when Model 2 trains, the impact of the errors (from examples where spurious correlations can be exploited) on changing the weights are much stronger than when Model 1 trains, as the number of examples that benefit from spurious correlations is much larger in comparison. As the training proceeds, this impact decreases as many examples are classified correctly by using spurious correlations while the weights continue to accumulate for Model 1, thus giving rise to oscillations.

How to terminate? We terminate training when the oscillations are stable and when the ensemble model is in the lower accuracy state, which corresponds to the state with lower correlation with color. To ensure the oscillations are stable, we do not terminate until a certain number of steps have been completed (in our experiments we set this duration to be number of steps= (training data size)/(batch size)). To capture the model in a state of lower correlation with color, we set a threshold on accuracy (we decide the threshold by observing the accuracy plot); we terminate only when the training accuracy falls below this threshold.

6. Conclusion

We developed a new framework based on game-theoretic tools to learn invariant predictors. We work with data from multiple environments. In our framework, we set up an ensemble game; we construct an ensemble of classifiers with each environment controlling one portion of the ensemble. Remarkably, the set of solutions to this game is exactly the same as the set of invariant predictors across training environments. The proposed framework performs comparably to the existing framework of Arjovsky et al. (2019) and also exhibits lower variance. We hope this framework opens new ways to address other problems pertaining to invariance in causal inference using tools from game theory.

References

Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., and Marchand, M. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Ash, R. B. and Doléans-Dade, C. A. *Probability and Measure Theory*. Academic Press, San Diego, California, 2000.

Bareinboim, E., Brito, C., and Pearl, J. Local characterizations of causal Bayesian networks. In *Graph Structures for Knowledge Representation and Reasoning*, pp. 1–17. Springer, 2012.

Barron, E., Goebel, R., and Jensen, R. Best response dynamics for continuous games. *Proceedings of the American Mathematical Society*, 138(3):1069–1083, 2010.

Beery, S., Van Horn, G., and Perona, P. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision*, pp. 456–473, 2018.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems*, pp. 137–144, 2007.

Bervoets, S., Bravo, M., and Faure, M. Learning and convergence to Nash in games with continuous action sets. Technical report, Working paper, 2016.

Daskalakis, C., Ilyas, A., Syrgkanis, V., and Zeng, H. Training gans with optimism. *arXiv preprint arXiv:1711.00141*, 2017.

de Haan, P., Jayaraman, D., and Levine, S. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems*, pp. 11693–11704, 2019.

Duchi, J., Glynn, P., and Namkoong, H. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425*, 2016.

Freund, Y., Schapire, R., and Abe, N. A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612, 1999.

Fudenberg, D. and Levine, D. K. *The Theory of Learning in Games*. MIT Press, Cambridge, Massachusetts, 1998.

Fudenberg, D. and Tirole, J. *Game Theory*. MIT Press, Cambridge, Massachusetts, 1991.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Glorot, X., Bordes, A., and Bengio, Y. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the International Conference on Machine Learning*, pp. 513–520, 2011.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. Covariate shift by kernel mean matching. In *Dataset Shift in Machine Learning*. 2009.

Heinze-Deml, C., Peters, J., and Meinshausen, N. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2), 2018.

Herings, P. J.-J. and Predtetchinski, A. Best-response cycles in perfect information games. *Mathematics of Operations Research*, 42(2):427–433, 2017.

Hofbauer, J. and Sorin, S. Best response dynamics for continuous zero-sum games. *Discrete and Continuous Dynamical Systems Series B*, 6(1):215, 2006.

Hoffman, J., Mohri, M., and Zhang, N. Algorithms and theory for multiple-source adaptation. In *Advances in Neural Information Processing Systems*, pp. 8246–8256, 2018.

Hofmann, T., Schölkopf, B., and Smola, A. J. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, 2008.

Hsieh, Y.-P., Liu, C., and Cevher, V. Finding mixed nash equilibria of generative adversarial networks. *arXiv preprint arXiv:1811.02002*, 2018.

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

Janzing, D., Mooij, J., Zhang, K., Lemeire, J., Zscheischler, J., Daniušis, P., Steudel, B., and Schölkopf, B. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 182:1–31, 2012.

Johansson, F. D., Ranganath, R., and Sontag, D. Support and invertibility in domain-invariant representations. *arXiv preprint arXiv:1903.03448*, 2019.

Lee, J. and Raginsky, M. Minimax statistical learning with Wasserstein distances. In *Advances in Neural Information Processing Systems*, pp. 2687–2696, 2018.

Magliacane, S., van Ommen, T., Claassen, T., Bongers, S., Versteeg, P., and Mooij, J. M. Domain adaptation by using causal inference to predict invariant conditional distributions. In *Advances in Neural Information Processing Systems*, pp. 10846–10856, 2018.

Mertikopoulos, P. and Zhou, Z. Learning in games with continuous action sets and unknown payoff functions. *Mathematical Programming*, 173(1-2):465–507, 2019.

Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. *arXiv preprint arXiv:1902.00146*, 2019.

Pearl, J. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

Pearl, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

Peters, J., Bühlmann, P., and Meinshausen, N. 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.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pp. 1177–1184, 2008.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. On causal and anticausal learning. *arXiv preprint arXiv:1206.6471*, 2012.

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

Subbaswamy, A., Chen, B., and Saria, S. Should I include this edge in my prediction? Analyzing the stability-performance tradeoff. *arXiv preprint arXiv:1905.11374*, 2019.

Sugiyama, M., Suzuki, T., Nakajima, S., Kashima, H., von Bünau, P., and Kawanabe, M. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

Zhao, H., Combes, R. T. d., Zhang, K., and Gordon, G. J. On learning invariant representation for domain adaptation. *arXiv preprint arXiv:1901.09453*, 2019.

Zhao, S., Fard, M. M., Narasimhan, H., and Gupta, M. Metric-optimized example weights. *arXiv preprint arXiv:1805.10582*, 2018.