2022 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, AGU. 22–25, 2022, XI'AN, CHINA

JOINT COVARIATE-ALIGNMENT AND CONCEPT-ALIGNMENT: A FRAMEWORK FOR DOMAIN GENERALIZATION

Thuan Nguyen [⋆†] *Boyang Lyu*[†] *Prakash Ishwar* [‡] *Matthias Scheutz* [⋆] *Shuchin Aeron* [†]

[⋆] Department of Computer Science, Tufts University, Medford, MA 02155
[†] Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155
[‡] Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215

ABSTRACT

In this paper, we propose a novel domain generalization (DG) framework based on a new upper bound to the risk on the unseen domain. Particularly, our framework proposes to jointly minimize both the covariate-shift as well as the concept-shift between the seen domains for a better performance on the unseen domain. While the proposed approach can be implemented via an arbitrary combination of covariate-alignment and concept-alignment modules, in this work we use well-established approaches for distributional alignment namely, Maximum Mean Discrepancy (MMD) and covariance Alignment (CORAL), and use an Invariant Risk Minimization (IRM)-based approach for concept alignment. Our numerical results show that the proposed methods perform as well as or better than the state-of-the-art for domain generalization on several data sets.

***Index Terms*—** Domain generalization, domain alignment, out-of-distribution generalization, distribution shift.

1. INTRODUCTION

Domain generalization (DG) has been studied extensively over the past decade as an important practical problem arising in a number of areas such as computer vision, signal processing, and medical imaging [**?**] [1]. Like standard learning settings, DG aims to learn a model from several seen domains (training data) that can generalize well on an unseen domain (test data). However, in contrast to the standard setting where the test data is assumed to come from the same distribution as training data, in DG, the distribution of the test data is different, *i.e.*, there is a presence of what is referred to as a *distribution shift*. This phenomena can be observed in many practical settings [2].

A number of approaches for DG are based on the assumption that there exist domain-invariant features that are transferable and unchanged from domain to domain. Thus, a classifier designed on top of these features will likely generalize well to the unseen domain. In practice, DG methods based on this assumption consist of two steps: (*a*) learning a good representation function that outputs the domain-invariant features, and (*b*) designing a classifier based on these domain-invariant features.

Different definitions of domain-invariant features have been proposed and they lead to different feature selection schemes. In [**?**, 3–9], a feature is defined as domain-invariant if its marginal distribution is unchanged over domains. On the other hand, in [10–17] a feature is considered as domain-invariant if its conditional distribution of the label given the representation feature is unchanged from domain to domain.

The differences in the definitions are rooted in the different modeling assumptions regarding the domains. Particularly, in the *covariate-shift* setting [18], the marginal distribution of data varies across the domains. On the other hand, in the setting of *concept-shift* [18], the conditional distribution of the label (class) conditioned on the data varies from domain to domain[1]. We survey related work in this context in Sec. 2.

In this paper, we revisit the seminal upper bound for the prediction risk in the unseen domain derived in [19] and show the necessity of jointly designing a representation function that minimizes both the covariate-shift and the concept-shift risks. We make the following contributions:

- We derive a novel upper bound for the prediction risk in the unseen domain that consists of two terms, namely, a covariate-alignment term and a concept-alignment term. This theoretical result motivates a domain generalization algorithm that combines both covariate-alignment and concept-alignment algorithms.
- We propose two new algorithms MMD-CEM and CORAL-CEM that combine, respectively, the COR-relation ALignment algorithm (CORAL) [20] and the Maximum Mean Discrepancy algorithm (MMD) [3] with the Conditional Entropy Invariant Risk Minimization algorithm (CEM) [21].

The authors would like to acknowledge funding provided by AFOSR grant # FA9550-18-1-0465. SA would like to acknowledge support by NSF CCF 1553075.

[1] The setting where the conditional distribution of data conditioned on the label (class) changes from domain to domain is also referred to as *concept-shift*. However, we do not consider that scenario in this paper.

978-1-6654-8547-0/22//$31.00 ©2022 IEEE

- We compare the proposed algorithms (MMD-CEM, CORAL-CEM) with six competing DG methods. Our methods exhibit similar or better performance compared to the six alternatives on both CS-CMNIST [**?**] and CMNIST [10] datasets.

The remainder of this paper is structured as follows. In Sec. 2, we summarize the recent work on DG dealing with covariate-shift and concept-shift. Sec. 3 formally defines the problem and clarifies notation used. Sec. 4 provides the main theoretical results which motivate the practical algorithms of Sec. 5. Finally, we provide the numerical results in Sec. 6 and conclude in Sec. 7.

2. RELATED WORK

Under the covariate-shift setting, domain alignment methods focus on aligning the marginal distributions of the representation from different seen domains by minimizing distributional divergences or distances [**?**, 3–9]. For instance, in [8, 9], the Wasserstein distance between distributions of the seen domains in representation space is minimized. In [3], Li *et al.* proposed the MMD algorithm to minimize the maximum mean discrepancy distance between seen domain distributions in the representation space. In a similar vein, the CORAL algorithm [20] is based on the idea of matching the mean and covariance of feature distributions from different domains. In [5], the authors use deep neural networks to minimize the difference between the variances of transformed features from seen domains to achieve domain generalization. In [7], the authors aim to learn the domain-invariant features with marginal distributions unchanged from domain to domain together with the domain-specific features to enhance the generalization performance.

Since the conditional distribution of the label given the representation variable is directly related to the classification performance, there are many works that aim to deal with concept-shift, *i.e.*, minimizing the divergence between the conditional distribution of the labels (concepts) given the representation [10–17]. In [10, 16], linear and non-linear IRM algorithms are proposed for learning the invariant features such that their conditional distributions are unchanged across domains, leading to the construction of an optimal classifier for all seen domains. In [21], Nguyen *et al.* developed a conditional entropy minimization principle to remove spurious invariant features from the invariant features learned by the IRM algorithm [10]. In [11], concept-alignment is achieved via minimizing the mutual information of the label given the representation variable for a given domain. In [13], Wang *et al.* handle concept-shift by directly aligning the conditional distribution within each class regardless of domains via the use of Kullback–Leibler divergence.

To the best of our knowledge, there are only two works in the DG setting that simultaneously deal with both the covariate-shift and concept-shift [22, 23][2]. In [22], Nguyen *et al.* proposed an algorithm that is capable of achieving both covariate-alignment and concept-alignment by enforcing the latent representation to be invariant under all transformation functions. Their algorithm, however, requires the use of invertible and differentiable transformation functions together with a generative adversarial network which is known to be hard to train. In [23], motivated by some examples where using concept-alignment alone is inadequate for achieving a good generalization, a covariate-alignment algorithm is heuristically combined with a concept-alignment algorithm to achieve a better out-of-distribution generalization. In contrast to [22, 23], our work appears to be the first work that uses the upper bound for the risk in the unseen domain to theoretically motivate the necessity of achieving both covariate and concept alignment in DG.

3. PROBLEM FORMULATION

Let $\boldsymbol{x} \in \mathbb{X} \subseteq \mathbb{R}^d$ denote the input data and $y \in \mathbb{Y}$ denote the label. The domain generalization task aims to learn a representation function $f : \mathbb{R}^d \to \mathbb{R}^{d'}$, $d' \leq d$, followed by a classifier $g : \mathbb{R}^{d'} \to \mathbb{Y}$ trained using data from seen domains $D^{(s)}$, but generalizes well to data from an unseen domain $D^{(u)}$.

Let $\boldsymbol{z} = f(\boldsymbol{x})$ denote the latent variable induced by input data $\boldsymbol{x}$ under the mapping f. We denote the distributions of data from a seen domain and the unseen domain by $p^{(s)}(\boldsymbol{x})$ and $p^{(u)}(\boldsymbol{x})$, respectively. The distributions of corresponding latent variables in a seen domain and the unseen domain are denoted by $p^{(s)}(\boldsymbol{z})$ and $p^{(u)}(\boldsymbol{z})$, respectively. We use Z to denote the latent random variable and Y to denote the label random variable.

An optimal labeling function for a given domain and a given representation function f is the labeling function that minimizes the prediction risk in the domain using the given representation function. For a given representation function f, we denote the optimal labeling functions from representation space to label space in a seen domain and the unseen domain by $l^{(s)}(\boldsymbol{z})$ and $l^{(u)}(\boldsymbol{z})$ respectively. For a given f, the risks of using a classifier g (possibly stochastic) in a seen domain and the unseen domain are defined by:

$$\epsilon^{(s)}(g, l^{(s)}) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p^{(s)}(\boldsymbol{z})} d(g(\boldsymbol{z}), l^{(s)}(\boldsymbol{z})), \tag{1}$$

$$\epsilon^{(u)}(g, l^{(u)}) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p^{(u)}(\boldsymbol{z})} d(g(\boldsymbol{z}), l^{(u)}(\boldsymbol{z})), \tag{2}$$

where $\mathbb{E}[\cdot]$ denotes expectation and $d(\cdot, \cdot)$ is a loss function that captures the mismatch between the classifier g and the

[2] But there are several works that simultaneously handle both covariate-shift and concept-shift under the Domain Adaptation (DA) setting, for example the work in [24, 25]. However, DA is not considered in this paper.

optimal labeling functions $l^{(s)}, l^{(u)}$. In addition, for a given representation function f we define the following quantity:

$$\epsilon^{(s)}(g, l^{(u)}) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p^{(s)}(\boldsymbol{z})} d(g(\boldsymbol{z}), l^{(u)}(\boldsymbol{z})) \tag{3}$$

which is the loss of using classifier g to input data from a seen domain s transformed by f and labeled with the optimum labeling function of the unseen domain for representation f. Since $p^{(u)}(\boldsymbol{z})$ and $l^{(u)}(\boldsymbol{z})$ depend on f, we want to learn a good representation function f together with a classifier g to minimize the prediction risk in the unseen domain $\epsilon^{(u)}(g, l^{(u)})$.

4. MAIN RESULTS

Theorem 1. *If the loss function $d(\cdot,\cdot)$ is non-negative, symmetric and satisfies the triangle inequality, then for a given representation function f:*

$$\begin{aligned}\epsilon^{(u)}(g, l^{(u)}) &\leq \epsilon^{(s)}(g, l^{(s)}) \\ &+ \int_{\boldsymbol{z}} |p^{(u)}(\boldsymbol{z}) - p^{(s)}(\boldsymbol{z})|\, d(g(\boldsymbol{z}), l^{(u)}(\boldsymbol{z}))\, d\boldsymbol{z} \\ &+ \int_{\boldsymbol{z}} p^{(s)}(\boldsymbol{z})\, d(l^{(u)}(\boldsymbol{z}), l^{(s)}(\boldsymbol{z}))\, d\boldsymbol{z}. \end{aligned} \tag{4}$$

Proof. We have:

$$\begin{aligned}\epsilon^{(u)}(g, l^{(u)}) &= \epsilon^{(u)}(g, l^{(u)}) \\ &+ \epsilon^{(s)}(g, l^{(s)}) - \epsilon^{(s)}(g, l^{(s)}) \\ &+ \epsilon^{(s)}(g, l^{(u)}) - \epsilon^{(s)}(g, l^{(u)}) \qquad (5) \\ &\leq \epsilon^{(s)}(g, l^{(s)}) \\ &+ |\epsilon^{(u)}(g, l^{(u)}) - \epsilon^{(s)}(g, l^{(u)})| \\ &+ \epsilon^{(s)}(g, l^{(u)}) - \epsilon^{(s)}(g, l^{(s)}) \qquad (6) \\ &\leq \epsilon^{(s)}(g, l^{(s)}) \\ &+ \int_{\boldsymbol{z}} |p^{(u)}(\boldsymbol{z}) - p^{(s)}(\boldsymbol{z})|\, d(g(\boldsymbol{z}), l^{(u)}(\boldsymbol{z}))\, d\boldsymbol{z} \\ &+ \int_{\boldsymbol{z}} p^{(s)}(\boldsymbol{z})\, d(l^{(u)}(\boldsymbol{z}), l^{(s)}(\boldsymbol{z}))\, d\boldsymbol{z} \qquad (7)\end{aligned}$$

where (6) follows from the reorganization of (5) and the fact that $a \leq |a|$, $\forall a$ and (7) follows from the definitions of $\epsilon^{(u)}(g, l^{(u)})$, $\epsilon^{(s)}(g, l^{(s)})$, $\epsilon^{(s)}(g, l^{(u)})$ in (1), (2), (3), the symmetry of $d(\cdot,\cdot)$, and the triangle inequality $d(g(\boldsymbol{z}), l^{(u)}(\boldsymbol{z})) - d(g(\boldsymbol{z}), l^{(s)}(\boldsymbol{z})) \leq d(l^{(u)}(\boldsymbol{z}), l^{(s)}(\boldsymbol{z}))$. □

The bound in (4) characterizes the risk of using a classifier trained on a seen domain but applied to the unseen domain. The bound contains three terms: *(a)* the first term captures the risk induced by the classifier on a seen domain for a given representation function f, *(b)* the second term measures the discrepancy between the marginal distributions of seen and unseen domains in the latent space corresponding to representation function f, and *(c)* the third term quantifies the mismatch between optimal classifiers for seen and unseen domains for the given f.

Our bound shares some similarities with the bound proposed by Ben-David *et al.* [19]. Particularly, for any representation function f, the following bound holds [19]:

$$\epsilon^{(u)}(g, l^{(u)}) \leq \epsilon^{(s)}(g, l^{(s)}) + d_{\mathcal{H}}\Big(p^{(u)}(\boldsymbol{z}), p^{(s)}(\boldsymbol{z})\Big) + \lambda \tag{8}$$

where $d_{\mathcal{H}}$ is $\mathcal{H}$-divergence [26] and:

$$\lambda = \inf_{g} \Big(\epsilon^{(s)}(g, l^{(s)}) + \epsilon^{(u)}(g, l^{(u)})\Big). \tag{9}$$

In [19], the above bound was developed for the DA setting. We have adapted it to the DG setting by identifying "source" domains in DA with "seen" domains in DG and the "target" domain in DA with the "unseen" domain in DG. While the first two terms of (4) and (8) are quite similar, the main difference comes from the way that the third term is optimized in practice. The practical DA algorithm proposed in [19] ignores optimizing the third term λ of the upper bound in (8) since the second term of the infimum in (9) cannot be estimated during training. Moreover, the first term of the infimum in (9) is already captured in the first term of the upper bound in (8). In the DG setting too, since the unseen domain samples are not available during training, one usually optimizes the bound only over several seen domains as in [**?**, 1, 21]. In contrast, the third term in (4) measures the mismatch of optimal classifiers between domains and, therefore, can be practically optimized via designing a representation function f followed by a classifier g that is optimal for all (seen) domains. As will be shown later, finding an optimal classifier over all domains encourages the use of concept-alignment algorithms. Indeed, designing a classifier that is optimal over all domains is the principle behind the recently proposed IRM algorithm [10].

It is also worth comparing our bound with the bound proposed by Lyu *et al.* [9]. Indeed, Lemma 1 in [9] extends the work in [19] to establish a new upper bound on the prediction risk in the unseen domain that is a function of both the representation mapping f and the classifier g. While the first two terms of the bound in [9] are quite similar to our first two terms, their third term is a constant which is completely independent of both f and g. Thus, the bound in [9] does not encourage the use of concept-alignment algorithms. In practice, the proposed algorithms in [19] and [9] only aim to achieve covariate-alignment, *i.e.*, minimizing the discrepancy of marginal distributions between domains.

Next, to explicitly show that Theorem 1 motivates the combination of both covariate-alignment and concept-alignment algorithms, we define the following two terms:

$$\Delta_1 = \max_{\boldsymbol{z}} |p^{(s)}(\boldsymbol{z}) - p^{(u)}(\boldsymbol{z})|, \tag{10}$$

and:

$$\Delta_2 = \max_{\boldsymbol{z}} d(l^{(u)}(\boldsymbol{z}), l^{(s)}(\boldsymbol{z})). \tag{11}$$

The first quantity Δ_1 captures the maximum mismatch of marginal distributions between domains in latent space. Thus,

minimizing Δ_1 supports achieving covariate-alignment. On the other hand, the second quantity Δ_2 measures the maximum mismatch of optimal classifiers between domains. Since stochastic classifiers are considered, enforcing a small mismatch between classifiers leads to a small discrepancy of conditional distributions $p(Y|Z)$ between domains. In other words, minimizing Δ_2 over representation functions supports achieving concept-alignment.

Corollary 1. *Under the settings of Theorem 1, for a given representation function f:*

$$\begin{aligned}\epsilon^{(u)}(g, l^{(u)}) &\leq \epsilon^{(s)}(g, l^{(s)}) \\ &+ \Delta_1 \int_{\boldsymbol{z}} d(g(\boldsymbol{z}), l^{(u)}(\boldsymbol{z}))\, d\boldsymbol{z} \\ &+ \Delta_2. \end{aligned} \tag{12}$$

Proof. The proof directly follows by using Theorem 1, the definitions in (10), (11) and the fact that $\int_{\boldsymbol{z}} p^{(s)}(\boldsymbol{z}) d\boldsymbol{z} = 1$. □

If the representation space is compact and the distance $d(\cdot, \cdot)$ is bounded then $\int_{\boldsymbol{z}} d(g(\boldsymbol{z}), l^{(u)}(\boldsymbol{z}))\, d\boldsymbol{z}$ is finite. Thus, minimizing the risk on seen domain $\epsilon^{(s)}(g, l^{(s)})$ together with the covariate-alignment term Δ_1 and the concept-alignment term Δ_2 encourages the trained classifier g to generalize well on the unseen domain.

5. PRACTICAL APPROACH

Motivated by the theoretical results in Sec. 4, we propose a framework that combines covariate-alignment algorithms together with concept-alignment algorithms for minimizing risk in the unseen domain. There are myriad ways of combining a given covariate-alignment algorithm with a given concept-alignment algorithm. For simplicity, we focus on minimizing the sum of the empirical prediction risk in the seen domains and a nonnegative weighted combination of the cost functions for covariate-alignment and concept-alignment algorithms. We consider two well-studied algorithms for covariate-alignment, namely CORAL [20] and MMD [3], and one well-studied concept-alignment algorithm, namely CEM [21]. We propose two DG algorithms named CORAL-CEM and MMD-CEM whose objective functions are defined as follows:

$$R_{\text{CORAL-CEM}}(f, g) := R(f, g) + \alpha L_{\text{CORAL}}(f) + \beta L_{\text{CEM}}(f, g), \tag{13}$$

$$R_{\text{MMD-CEM}}(f, g) := R(f, g) + \alpha L_{\text{MMD}}(f) + \beta L_{\text{CEM}}(f, g), \tag{14}$$

where the first term is the summation of the empirical risk from all seen domains, *i.e.*, summation of $\epsilon^{(s)}(g, l^{(s)})$ over all seen domains, the second term is the covariate-alignment term which is implemented via either the CORAL algorithm [20] or the MMD algorithm [3], the third term is the concept-alignment term which is implemented via the CEM algorithm [21], and α, β are two positive hyper-parameters that control the trade-off between these loss terms.

6. EXPERIMENTS

6.1. Datasets

In this section, we examine our proposed methods on two datasets CMNIST [10] and CS-CMNIST [**?**]. To illustrate how CMNIST and CS-CMNIST datasets are generated, we use e to denote domain index, $e = 1, 2, 3$, X_g^e to denote the gray image in domain e, Y_g^e to denote the corresponding label of X_g^e, X^e to denote the colored image in domain e, and Y^e to denote the corresponding label of X^e. Each domain is specified by a bias parameter θ^e changing from domain to domain. We use C^e to denote the color index, i.e., the color assigned to X_g^e to produce X^e. Let $\oplus$ denote the XOR operator, and Bern(p) denote the Bernoulli distribution with parameter p. Of these three domains, two are selected as seen domains while the rest is unseen domain.

Colored MNIST (CMNIST) [10] is a variant of the gray MNIST handwritten digit dataset. The task is to classify a colored digit (in red or blue color) into two classes: the digit is strictly less than five or the digit is greater or equal five. There are two seen domains each contains 25,000 images and one unseen domain contains 20,000 images. By adding the color (spurious feature) to the digit, the label is more correlated with the color than with the digit, thus, any algorithm simply aims to minimize the training error will tend to discover the color rather than the shape of the digit and fail at the testing phase.

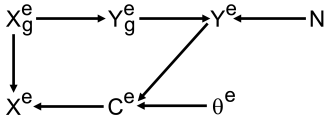

Fig. 1. Graphical model for CMNIST.

The graphical model for CMNIST dataset is illustrated in Fig. 1 that contains the following steps:

1. $Y_g^e \leftarrow L(X_g^e)$: from 10 digits, we construct a binary classification problem by labeling $Y_g^e = 0$ if the digit is less than or equal to four and labeling $Y_g^e = 1$, otherwise.

2. $Y^e \leftarrow Y_g^e \oplus N$, $N \sim \text{Bern}(0.25)$: noise is added to the label Y^e of the colored image.

3. $C^e \leftarrow Y^e \oplus \theta^e$, $\theta^e \sim \text{Bern}(p^e)$: the index color C^e is selected with a domain bias parameter θ^e. For $e = 1, 2, 3$, $p^e = 0.1, 0.2, 0.9$, respectively.

4. $X^e \leftarrow T(X_g^e, C^e)$: the gray image X_g^e is colored with the index color C^e to produce the colored image X^e.

Covariate-Shift-CMNIST (CS-CMNIST) [**?**] is a synthetic dataset derived from CMNIST. This dataset contains three domains (two training domains and one test domain) having 20,000 images each. There are ten classes in CS-CMNIST where each class corresponds to a digit from one to nine. Each

Datasets	ERM [27]	IRM [10]	IB-ERM [12]	IB-IRM [12]	MMD-IRM [23]	CEM [21]	MMD-CEM (our)	CORAL-CEM (our)
CMNIST [10]	51.2 ± 0.1	51.4 ± 0.1	51.2 ± 0.3	52.1 ± 0.2	51.4 ± 0.1	**52.5 ± 0.3**	52.0 ± 0.2	**52.5 ± 0.1**
CS-CMNIST [?]	60.3 ± 1.2	62.5 ± 1.1	71.5 ± 0.7	71.9 ± 0.7	77.2 ± 0.9	86.1 ± 0.8	**90.7 ± 0.9**	89.9 ± 0.6

Table 1. Average accuracy of the compared methods.

class is associated with one color that is strongly correlated with the digits in the two seen (training) domains but is weakly correlated with the digits in the unseen (test) domain.

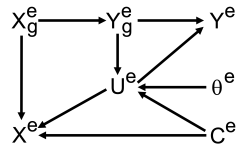

Fig. 2. Graphical model for CS-CMNIST.

The graphical model for CS-CMNIST dataset is illustrated in Fig. 2 that contains the following steps:

1. $Y_g^e \leftarrow L(X_g^e)$: the label of gray image is a function of gray image. There are 10 classes with labels from 0 to 9.
2. $C^e \leftarrow \text{Uniform}(\{0, \ldots, 9\})$: 10 colors are picked up randomly (from color 0 to color 9).
3. $U^e \leftarrow \text{Bern}\Big((C^e \oplus Y_g^e)\theta^e + (1-(C^e \oplus Y_g^e))(1-\theta^e)\Big)$: a pair of image and its color (X_g^e, C^e) is selected with probability θ^e if $Y_g^e = C^e$, else if $Y_g^e \neq C^e$, select (X_g^e, C^e) with probability $1-\theta^e$. For $e = 1, 2, 3$, the bias parameter $\theta^e = 0.1, 0.2, 0.9$, respectively.
4. $X^e \leftarrow T(X_g^e, C^e)|U^e = 1$, $Y^e \leftarrow Y_g^e|U^e = 1$: the gray image X_g^e is colored with color C^e to produce colored image X^e. The colored image is labeled by Y_g^e.

From the graphical models in Fig. 1 and 2, it is worth noting that while concept-shift is common in both CMNIST and CS-CMNIST, CS-CMNIST also suffers from covariate-shift (the third steps in its data generating process).

6.2. Compared Methods

We compare our proposed algorithms CORAL-CEM and MMD-CEM to ERM [27], IRM [10], IB-ERM [12], IB-IRM [12], MMD-IRM [23], and CEM [21]. Since the code for MMD-IRM algorithm was not released by its authors, we implemented this algorithm according to the description in [23].

6.3. Implementation

For the CS-CMNIST dataset, we follow the learning model in [12] that contains three convolutional layers with the corresponding feature dimensions of 256, 128, and 64. We use a stochastic gradient descent optimizer for training, the batch size is set to 128, the learning rate is 10^{-1} and decays after 600 steps while the total number of steps is $2,000$. 25 trials corresponding to 25 pairs of hyper-parameters (α, β) are uniformly selected from $[10^{-1}, 10^4]$.

For the CMNIST dataset, we follow the setting in [28] and use MNIST-ConvNet with four convolutional layers as the learning model. The details of MNIST-ConvNet can be found in Table 7 of [28]. 10 trials corresponding to 10 pairs of hyper-parameters (α, β) are uniformly selected in $[10^{-1}, 10^4]$. For each trial, the learning rate is randomly selected in $[10^{-4.5}, 10^{-3.5}]$ while the batch size is randomly selected in $[2^3, 2^9]$.

We use the training-domain validation set tuning procedure [28] for model selection, *i.e.*, selecting the model with the highest validation accuracy on the validation set sampled from seen domain data. We repeat our entire experiment five times for CS-CMNIST dataset and three times for CMNIST dataset. Finally, the average accuracy and its standard deviation are reported. Due to the limited space, the details of our implementation together with the source code can be found at this link[3].

6.4. Results and Discussion

Table 1 shows the average accuracy of the compared methods. As seen, our proposed algorithms outperform the rest of the tested methods with a margin of nearly 4% for CS-CMNIST dataset. However, when evaluating on a more challenging CMNIST dataset, the proposed algorithms only achieve comparable or slightly better performance than other tested methods. The substantial gain on CS-CMNIST can be explained by the way the data is generated. Indeed, by construction, CS-CMNIST suffers from both covariate-shift and concept-shift, therefore, it is necessary for achieving both covariate and concept alignment for better DG on CS-CMNIST. On the other hand, because CMNIST is designed in a way such that there exists a strong spurious correlation between colors and labels, no algorithm works on this dataset. This observation is also confirmed by the numerical results in [28].

7. CONCLUSIONS

In this paper, by revisiting the upper bound of generalization error of the unseen domain, we motivate a domain generalization framework that combines covariate-alignment algorithms together with concept-alignment algorithms to simultaneously handle both the covariate distribution shift and the concept distribution shift. Our numerical results show a gain on generalization performance which confirms our theoretical results.

[3]https://github.com/thuan2412/Joint-covariate-alignment-and-concept-alignment-for-domain-generalization

8. REFERENCES

[1] K. Zhou, Z. Liu, Y. Qiao, T. Xiang, and C. C. Loy, "Domain generalization: A survey," *arXiv preprint arXiv:2103.02503*, 2021.

[2] R. Taori, A. Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt, "Measuring robustness to natural distribution shifts in image classification," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18 583–18 599, 2020.

[3] H. Li, S. J. Pan, S. Wang, and A. C. Kot, "Domain generalization with adversarial feature learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5400–5409.

[4] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1414–1430, 2016.

[5] S. Erfani, M. Baktashmotlagh, M. Moshtaghi, X. Nguyen, C. Leckie, J. Bailey, and R. Kotagiri, "Robust domain generalisation by enforcing distribution invariance," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*. AAAI Press, 2016, pp. 1455–1461.

[6] X. Jin, C. Lan, W. Zeng, and Z. Chen, "Feature alignment and restoration for domain generalization and adaptation," *arXiv preprint arXiv:2006.12009*, 2020.

[7] M.-H. Bui, T. Tran, A. Tran, and D. Phung, "Exploiting domain-specific features to enhance domain generalization," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[8] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[9] B. Lyu, T. Nguyen, P. Ishwar, M. Scheutz, and S. Aeron, "Barycentric-alignment and invertibility for domain generalization," *arXiv preprint arXiv:2109.01902*, 2021.

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

[11] B. Li, Y. Shen, Y. Wang, W. Zhu, C. J. Reed, J. Zhang, D. Li, K. Keutzer, and H. Zhao, "Invariant information bottleneck for domain generalization," *arXiv preprint arXiv:2106.06333*, 2021.

[12] K. Ahuja, E. Caballero, D. Zhang, Y. Bengio, I. Mitliagkas, and I. Rish, "Invariance principle meets information bottleneck for out-of-distribution generalization," *arXiv preprint arXiv:2106.06607*, 2021.

[13] Z. Wang, M. Loog, and J. van Gemert, "Respecting domain relations: Hypothesis invariance for domain generalization," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 9756–9763.

[14] S. Zhao, M. Gong, T. Liu, H. Fu, and D. Tao, "Domain generalization via entropy regularization," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 096–16 107, 2020.

[15] O. E. Salaudeen and O. O. Koyejo, "Exploiting causal chains for domain generalization," in *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

[16] C. Lu, Y. Wu, J. M. Hernández-Lobato, and B. Schölkopf, "Nonlinear invariant risk minimization: A causal approach," *arXiv preprint arXiv:2102.12353*, 2021.

[17] K. Ahuja, K. Shanmugam, K. Varshney, and A. Dhurandhar, "Invariant risk minimization games," in *International Conference on Machine Learning*. PMLR, 2020, pp. 145–155.

[18] W. M. Kouw and M. Loog, "An introduction to domain adaptation and transfer learning," *arXiv preprint arXiv:1812.11806*, 2018.

[19] S. Ben-David, J. Blitzer, K. Crammer, F. Pereira *et al.*, "Analysis of representations for domain adaptation," *Advances in neural information processing systems*, vol. 19, p. 137, 2007.

[20] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *European conference on computer vision*. Springer, 2016, pp. 443–450.

[21] T. Nguyen, B. Lyu, P. Ishwar, M. Scheutz, and S. Aeron, "Conditional entropy minimization principle for learning domain invariant representation features," *arXiv preprint arXiv:2201.10460*, 2022.

[22] A. T. Nguyen, T. Tran, Y. Gal, and A. G. Baydin, "Domain invariant representation learning with domain density transformations," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[23] R. Guo, P. Zhang, H. Liu, and E. Kiciman, "Out-of-distribution prediction with invariant risk minimization: The limitation and an effective fix," *arXiv preprint arXiv:2101.07732*, 2021.

[24] W. Yang, T. Ling, C. Yang, L. Wang, Y. Shi, L. Zhou, and M. Yang, "Class distribution alignment for adversarial domain adaptation," *arXiv preprint arXiv:2004.09403*, 2020.

[25] J. Wen, N. Zheng, J. Yuan, Z. Gong, and C. Chen, "Bayesian uncertainty matching for unsupervised domain adaptation," *arXiv preprint arXiv:1906.09693*, 2019.

[26] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *VLDB*, vol. 4. Toronto, Canada, 2004, pp. 180–191.

[27] V. N. Vapnik, "An overview of statistical learning theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.

[28] I. Gulrajani and D. Lopez-Paz, "In search of lost domain generalization," *arXiv preprint arXiv:2007.01434*, 2020.