Class-wise Feature Distribution Matching Regularization for Domain Generalization

Seungmin Lee
Seoul National University
profile2697@gmail.com

Abstract

Domain generalization (DG) aims to learn a model that generalizes well to an unseen domain (target domain), which has a different distribution than known domains (source domain). Many of the previous works try to learn domain-invariant features. These methods adopt a loss that tries to match the whole distributions of the source domains. However, these works are sub-optimal because they rarely utilize task-specific information, such as class labels. For concerning the information, we propose a simple but effective regularizing method called Class-wise Feature Distribution Matching (FDM). The proposed method attempts to induce the network to produce similar features when the labels are the same. By doing this, a model is expected to learn more task-specific and robust features than the previous works.

1. Introduction

Deep learning has been remarkably successful in many areas []. However, many studies find out deep learning methods are hard to generalize when they encounter an unseen domain, which has a different data distribution than domains used for training []. This problem called *domain shift* []. For alleviating the domain shift problem, many studies have been carried out on different assumptions. Domain Adaptation (DA) assumes there are two domains []. The first is a fully-labeled source domain, and the other is a sparsely labeled or totally unlabeled target domain. Otherwise, domain generalization (DG) assumes there are some fully-labeled source domains, but the target domain is totally unavailable. DG is a challenge but important research area because generalization to other domains is crucial to make a safe artificial intelligence. Moreover, it is helpful to understand how deep neural networks see the world.

Existing DG studies can be classified into several categories depending on their strategies. Some methods proposed novel model architectures that are robust to domain shift []. Others suggested learning algorithms aim to induce a model to fit in a more robust minimum []. The others adopted losses to learn domain-invariant features by matching feature distributions of the source domains []. Although these domain-invariant feature learning methods work well, the methods are sub-optimal because they do not utilize task-specific information such as class labels explicitly.

Our approach also aims to learn domain-invariant features, but the proposed method explicitly adopts the task-specific information. More specifically, we add a simple consistency loss that induces the model to produce similar features when the labels are the same. By doing this, the model is expected to learn features that are adequately semantic and constant across domains.

To demonstrate the proposed method works, we will conduct experiments on the standard DG benchmarks such as PACS [] or VLCS []. Specifically, we will show the consistency loss is helpful to improve the performance of the baseline, which simply aggregates all the source domains and uses it as a training set. Then, we will show that the proposed method is comparable to or better than many previous works.

2. Related Works

2.1. Multi-Domain Learning and Multi-Source Domain Adaptation

The primary purpose of Multi-Domain Learning (*MDL*) is to learn a single model that can compactly represent all domains with a smaller number of parameters []. For this purpose, Bilen *et al.* [] adopts shared model parameters except for batch normalization parameters and instance normalization parameters. Rebuffi *et al.* [] transforms the standard residual network architecture to share a significant amount of parameters between different domains.

Multi-Source Domain Adaptation (*MSDA*) also uses a set of domains, but it additionally utilizes the images of an unlabeled target domain []. The main focus of MSDA is to train a model that works well on the target domain without labels of it. Even though many studies have been

conducted on single-source domain adaptation, there are a limited number of researches on MSDA []. Chang *et al.* [] proposes a domain-specific batch normalization with shared weights parameters and extends their method to MSDA. Peng *et al.* [] suggests reducing moment distances between different domains. The moment distance measures the difference between feature distributions of two domains without concerning the task at hand.

MDL and MSDA are closely related to DG since DG also utilizes many sources in many cases. However, DG is different from MDL in that the primary focus of the DG is to learn semantic and domain-invariant features, not to learn compact representations. Moreover, DG is more challenging than MSDA because DG can not utilize the target domain in training.

2.2. Domain Generalization

Even though existing DG methods basically aim to learn domain-invariant features, these can be classified into several groups based on their strategies. The first group proposes a novel architecture []. The methods basically separate domain-specific parameters and domain-agnostic parameters. After that, they only extract and utilize the domain-agnostic parameters for the unseen domain. The second group of methods suggests optimization algorithms that adopt meta-learning or episodic learning []. For example, MLDG [] constructs an episode by splitting the source domains into training domains and test domains in each iteration. The final group of methods uses losses that aims to learn domain-invariant features []. The losses such as MMD [] just try to match the feature distributions of all available source domains without concerning the task at hand. Therefore, methods that adopted MMD show suboptimal performances []. We propose a regularizing method using a simple consistency loss that explicitly utilizes the task-specific information. By using this simple loss, we expect that the model can learn domain-invariant but semantic features.

3. Proposed Method

3.1. Problem Setting and Notation

In DG, we assume that there are n source domains $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n\}$ where $\mathcal{D}_i$ indicates i-th source domain which contains sample-label pairs $\{x_i^j, y_i^j\}$. Using the source domains, we try to learn a model that generalizes well to unseen target domain $\mathcal{D}_t$. We use a model h consists of a feature extractor g and a classifier f.

3.2. Deep-All Method

The Deep-All Method is a simple but effective baseline. In this method, we just aggregate all examples of source domains and train the model using the aggregated samples. If we work on a classification task, we can use cross-entropy loss as follows:

$$L_{all} = \underset{g,f}{\operatorname{argmin}} \, \mathbb{E}_{\mathcal{D}_s \sim \mathcal{D}} \left[\mathbb{E}_{\mathbf{x}_i, y_i \sim \mathcal{D}_s} \left[\mathbf{y}_i{}^T \log h(\mathbf{x}_i) \right] \right] \quad (1)$$

where $\mathbf{y}_i$ is a one-hot vector representation of y_i

3.3. Class-wise Feature Distribution Matching Regularization

We propose a consistency loss called Class-wise Feature Distribution Matching Regularization (*FDM*), which tries to make a model generate similar features when the labels are the same. The FDM loss is calculated as follows: For each iteration, the feature extractor produces features of the source domains. After that, we calculate the averages of the features by classes for reducing the variance. Lastly, the FDM loss is calculated as a consistency loss between the averages and the features that share the same label. The FDM loss is defined as follows:

$$L_{fdm} = \underset{f}{\operatorname{argmin}} \, \mathbb{E}_{\mathcal{D}_s \sim \mathcal{D}} \left[\mathbb{E}_{\mathbf{x}_i, y_i \sim \mathcal{D}_s} \left[\mathbb{D}_{KL}(f(\mathbf{x}_i) || \mathbf{m}_{y_i}) \right] \right] \quad (2)$$

where $\mathbb{D}_{KL}(\cdot||\cdot)$ represents Kullback–Leibler divergence and $\mathbf{m}_{y_i}$ is the average feature of class y_i. For simplicity, $\mathbf{m}_{y_i}$ is calculated for each batch.

Finally, overall loss function is defined as a weighted sum of L_{add} and L_{fdm} as follows:

$$L = L_{all} + \lambda L_{fdm} \quad (3)$$

where λ is a hyperparameter that controlls the magnitude between the two losses.

4. Experiments

5. Future Works

6. Conclusion

References

[1] FirstName Alpher. Frobnication. *Journal of Foo*, 12(1):234–778, 2002.

[2] FirstName Alpher and FirstName Fotheringham-Smythe. Frobnication revisited. *Journal of Foo*, 13(1):234–778, 2003.

[3] FirstName Alpher, FirstName Fotheringham-Smythe, and FirstName Gamow. Can a machine frobnicate? *Journal of Foo*, 14(1):234–778, 2004.

[4] Authors. The frobnicatable foo filter, 2014. Face and Gesture submission ID 324. Supplied as additional material `fg324.pdf`.

[5] Authors. Frobnication tutorial, 2014. Supplied as additional material `tr.pdf`.