Episodic Training for Domain Generalization

Seungmin Lee (profile2697@gmail.com; 2013-11420), Dept. of Computer Science and Engineering, Seoul National University

1. Problem Setting

This paper attempts to solve *Domain Generalization* (DG). In DG, we assume that there are n source domains $\mathcal{D} = [\mathcal{D}_1, \dots, \mathcal{D}_n]$ where $\mathcal{D}_i$ indicates i-th source domain which contains data-label pairs $\{x_i^j, y_i^j\}$. Using source domains, we try to learn a model that generalizes well to unseen target domain D_t. This paper deals with two types of DGs, one *homogeneous DG* and the other *heterogeneous DG*. The homogeneous DG assumes all domains, including target domain, share the same label space. Meanwhile, the heterogeneous DG assumes the domains can have different label space. The authors used a model consists of a feature extractor θ and a classifier ψ.

2. Intuition

Like meta-learning, the authors tried to design an episode that can closely simulate the testing condition. First, they exposed a feature extractor to poorly calibrated classifiers and then trained the feature extractor to work well on those classifiers. We can interpret this training procedure as training the feature extractor to encode more general information that can be used for all classifiers. Besides, they trained a classifier with poorly calibrated feature extractors. This procedure can be understood as training the classifier to be robust to noise generated by feature extractors.

3. Methods

The proposed method uses little bit complicating training procedures but basically uses a weighted sum of losses described in this section.

Vanilla Aggregation Method Aggregation Method is simple yet effective baseline. We just aggregate data-label pairs from all source domains and train a model using those pairs. The loss can be written as follows:

$$L_{agg} = \operatorname*{argmin}_{\theta,\psi} \mathbb{E}_{\mathcal{D}_i \sim \mathcal{D}} \left[\mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_i} \left[\ell(y_i, \psi(\theta(\mathbf{x}_i))) \right] \right] \tag{1}$$

Domain-Specific Models For creating poorly calibrated feature extractors and classifiers, the authors trained domain-specific models trained only on their domain using Eq. 2,

$$L_{ds} = \operatorname*{argmin}_{\theta_i,\psi_i} \mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_i} \left[\ell(y_i, \psi_i(\theta_i(\mathbf{x}_i)) \right] \tag{2}$$

where ℓ is cross entropy loss.

Episodic Training for Feature Extractor To get a robust feature extractor, the authors trained a feature extractor to work well with other domains' classifiers. If the classifiers are vulnerable to texture or style, the feature extractor should learn not to extract those kinds of noise. This procedure makes the feature extractor to encode general and discriminative features. This procedure can be written as follows:

$$L_{epif} = \operatorname*{argmin}_{\theta} \mathbb{E}_{i,j \sim [1,n], i \neq j} \left[\mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_i} \left[\ell(y_i, \overline{\psi}_j(\theta(\mathbf{x}_i))) \right] \right] \tag{3}$$

where $\overline{\psi}_j$ means a fixed classifier of j-th domain. Fixing weights is crucial because it prevents weights to be polluted by other domains.

Episodic Training for Classifier Similar to **Episodic Training for Feature Extractor**, we can train a robust classifier using other domains' feature extractors. If the feature extractors encode noisy information like style or texture, the classifier learns not to use those kinds of information. This training makes the classifier robust to noise from feature extractors. This procedure using following loss:

$$L_{epic} = \operatorname*{argmin}_{\psi} \mathbb{E}_{i,j \sim [1,n], i \neq j} \left[\mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_i} \left[\ell(y_i, \psi(\overline{\theta}_j(\mathbf{x}_i))) \right] \right] \tag{4}$$

where $\overline{\theta}_j$ is fixed feature extractor.

Episodic Training by Random Classifier The episodic training procedures described above are limited to homogeneous DG. For heterogeneous DG, the authors proposed episodic training using a random classifier. The use of random classifier is an extreme version of a poorly calibrated classifier. This procedure uses a loss described below:

$$L_{epir} = \operatorname*{argmin}_{\theta} \mathbb{E}_{\mathcal{D}_i \sim \mathcal{D}} \left[\mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_i} \left[\ell(y_i, \overline{\psi}_r(\theta(\mathbf{x}_i))) \right] \right] \tag{5}$$

where $\overline{\psi}_r$ is a random classifier with fixed weights.

4. Personal Memo

There must be a better way to create a poorly calibrated sets of feature extractors and classifiers. Despite ablation study was given, it is hard to understand why the random classifier is helpful to DG. Additionally, I wonder why the difference between coefficients of L_{epir} and L_{epif} is up to 20 times.