Exploring and Utilizing Pattern Imbalance

Shibin Mei, Chenglong Zhao, Shengchao Yuan, Bingbing Ni

2022/05/20

Paper tag: TUE-PM-329

01

Motivation

02

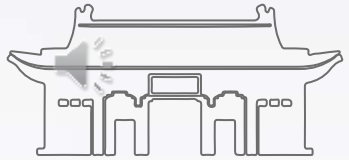
Method

03

Experiments

04

Conclusion



Motivation

How to improve the generalization of DNN models?

Existing problems:

- Encouraging the model to learn domain invariant features
- Avoiding spurious features

Our work:

- Solving domain generalization by exploring the character of the datasets.
- We attribute the domain generalization problem to the mining of hard or minority patterns under imbalanced patterns.
- We define a new concept, seed category, to promote model training by paying full attention to various patterns in the data set.

Method

Preliminary

$$D_e = \{X_i^e, Y_i^e\}_{e=1}^{n^e}$$

$$R^e(\theta) = \mathbb{E}[\mathcal{L}(f_\theta(X^e), Y^e)]$$

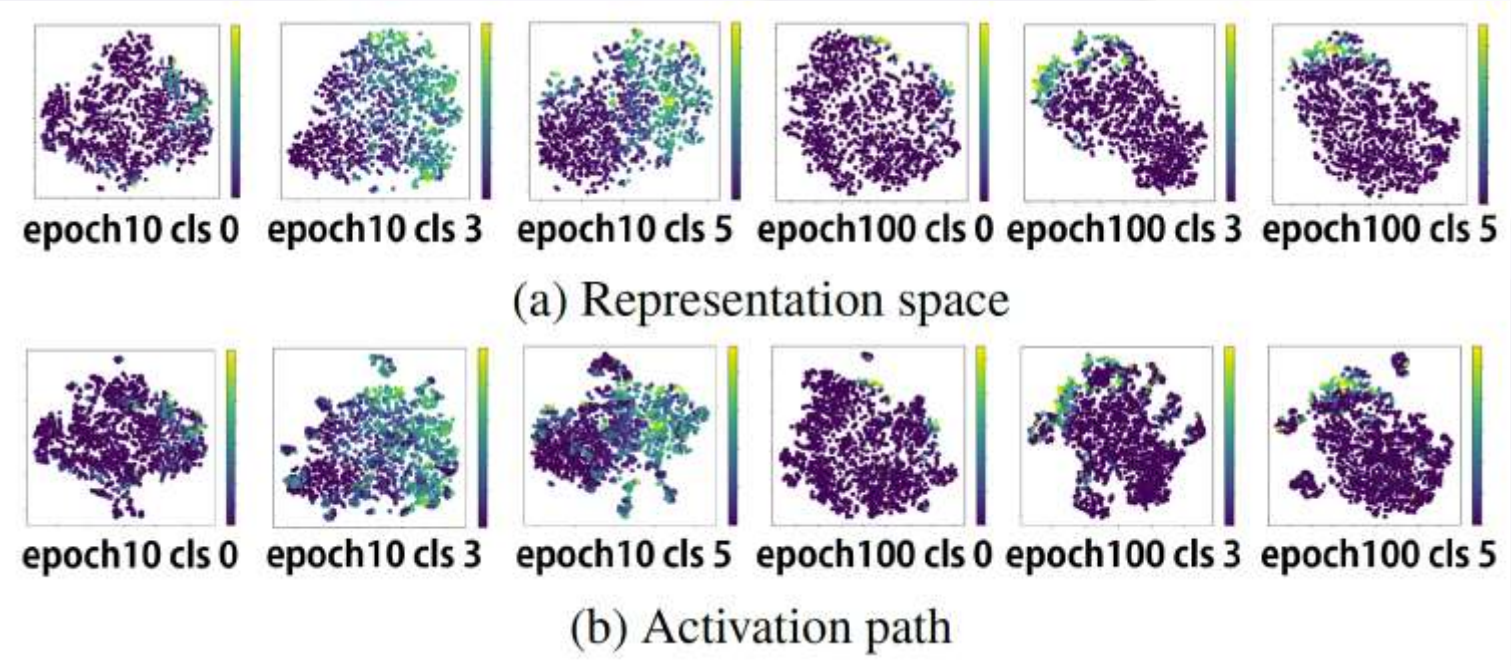
$$\min_{\theta \in \Theta} \max_{e \in \mathcal{E}} R^e(\theta)$$

The objective of the domain generalization problem is minimizing the worst case (worst domain) classification loss.

Method

Identifying Pattern Imbalance

- Data view
- Model view
- Optimization view



Seed Category

Definition 1 *(Seed) For a given dataset $\mathcal{D}$ and the above concept of adjacent samples, we define seed set $\mathcal{S}$ if and only if for any sample $x \in \mathcal{D} - \mathcal{S}$, there exists at least one $s \in \mathcal{S}$ that satisfies x and s are adjacent samples. We define each sample in the smallest seed set $\mathcal{S}$ as seed.*

Definition 2 *(Seed Category) For any seed s, we define the union of this seed and the samples $\mathcal{B}_s$ subordinate to it as seed category, i.e.,*

$$\mathcal{C}_{seed} = \{s\} \cup \mathcal{B}_s. \tag{4}$$

Dynamic distribution algorithm

Algorithm 1 Dynamic Distribution Based on Seed Category

Require: Original data $\mathcal{D}$, total training steps T, re-distribution cycle t_0

Ensure: Model parameters θ

1: **for** t= 1 to T **do**
2: Conduct re-distribution and update seed category for $\mathcal{D}$ every t_0 steps.
3: Maintain a weight distribution vector q on these seed categories.
4: Sample data from each seed category as a batch.
5: Calculate average losses and update weights for each seed category.
6: Weight normalization.
7: Update model parameters θ according to distribution weight vector q.
8: **end for**

Experiments

Algorithm	ERM	DANN	IRM	GDRO	MLDG	MMD	MTL	ARM	SagNet	VREx	Ours
Train-domain validation set											
A	81.1	81.9	82.7	86.2	81.8	84.9	82.5	82.1	83.0	81.4	82.0
C	81.6	77.8	78.5	80.5	80.0	81.0	79.9	82.9	78.6	81.4	80.3
P	97.0	95.1	96.4	96.2	95.3	95.7	95.5	93.6	95.3	96.0	96.8
S	74.1	75.4	74.3	75.3	69.5	73.3	79.6	76.0	80.7	77.8	80.8
Avg.	83.5	82.6	83.0	84.5	81.6	83.7	84.4	83.6	84.4	84.2	**85.0**
Leave-one-domain-out cross validation											
A	82.7	79.0	82.7	81.3	82.6	81.6	80.0	77.2	80.7	81.8	83.5
C	80.0	76.1	78.5	76.8	79.5	80.8	80.3	82.9	78.1	79.9	79.9
P	95.3	95.1	96.4	94.8	97.7	95.1	96.7	93.1	95.7	95.4	96.2
S	75.3	72.4	74.3	80.3	70.5	71.7	74.6	73.2	63.6	72.8	83.2
Avg.	83.3	78.6	83.0	83.3	82.5	82.3	82.9	81.6	79.6	82.4	**85.7**
Test-domain validation set											
A	80.9	74.0	72.4	72.4	82.6	81.2	84.5	73.5	77.5	81.8	80.2
C	81.2	75.6	77.1	79.0	81.3	81.7	76.1	76.7	78.6	79.7	79.6
P	95.1	91.0	90.9	94.8	94.9	95.1	94.2	94.7	95.7	95.3	94.4
S	78.1	76.1	74.1	72.6	73.2	78.8	74.6	70.6	77.4	76.3	84.3
Avg.	83.8	79.2	78.9	79.7	83.0	84.2	82.3	78.9	82.3	83.3	**84.6**

PACS

Algo.	A	C	P	R	Avg.
ERM	54.7	47.3	72.7	74.0	62.2
DANN	54.3	51.1	73.0	67.4	61.5
IRM	55.4	49.1	68.2	75.0	61.9
GDRO	55.7	52.0	71.4	74.7	63.4
MTL	53.0	47.1	70.5	76.3	61.7
ARM	51.9	46.8	69.8	71.0	59.9
SagNet	53.1	49.0	72.5	73.4	62.0
Ours	57.5	50.4	73.2	74.0	**63.8**

OfficeHome

Algo.	C	L	S	V	Avg.
ERM	98.1	59.0	70.1	74.6	75.4
DANN	98.5	63.0	56.9	74.6	73.2
IRM	95.4	59.3	74.2	76.0	76.2
GDRO	94.9	66.3	69.7	71.3	75.6
MTL	96.1	59.5	70.0	73.0	74.6
ARM	94.6	62.9	74.0	70.3	75.4
SagNet	93.4	60.4	75.1	75.0	76.0
Ours	97.4	66.4	70.1	72.8	**76.7**

VLCS

Experiments

Datasets	ColoredMNIST						PACS					
Methods	Loss			Cluster			Loss			Cluster		
Seeds per domain	2	3	4	2	3	4	2	3	4	2	3	4
Model Selection 1	51.5	51.5	**52.2**	51.6	51.6	51.9	82.9	83.0	**85.0**	82.5	82.6	82.6
Model Selection 2	38.5	41.8	**45.9**	37.3	37.3	37.5	82.8	84.2	**85.7**	83.5	83.5	83.8
Model Selection 3	59.2	57.4	**60.5**	54.8	59.3	53.0	83.1	79.6	**84.6**	82.2	81.8	80.9

Performance vs. seed category calculation methods

Datasets	CMNIST		RMNIST		PACS		OfficeHome		VLCS	
Methods	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours	Baseline	Ours
PCMA(G)	0.186	0.186	0.437	0.437	8.112	8.124	8.113	8.128	8.117	8.125
TTR(min)	6.18	7.14	7.75	9.62	44.15	48.14	66.53	71.19	76.57	81.71

Efficiency of our method compared with baseline. PCMA represents peak CUDA memory allocated and TTR represents the time of a training round.

Conclusion

Contribution

- We identify pattern imbalance generally existed in classification tasks and give a new definition of seed category, that is, the inherent pattern to recognize.
- We further develop a dynamic weight distribution training strategy based on seed category to facilitate out-of-distribution performance.
- Extensive experiments on several domain generalization datasets well demonstrate the effectiveness of the proposed method.

Thank you for listening !

饮 水 思 源　爱 国 荣 校