Ensemble of Averages: Improving Model Selection and Boosting Performance in Domain Generalization

Arpit, D., Wang, H., Zhou, Y., & Xiong, C. Ensemble of Averages: Improving Model Selection and Boosting Performance in Domain Generalization. In Advances in Neural Information Processing Systems.

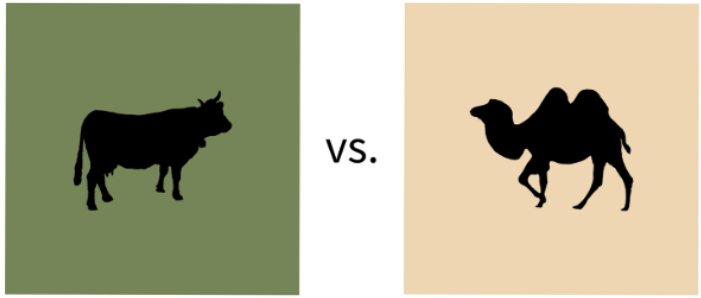
Introduction

- Independent, Identical Distribution (I.I.D) : 어떤 랜덤 확률 변수 집합이 있을 때 각각의 랜덤 확률변수들은 독립적이면서 동일한 분포를 가지는 것을 의미(Ex. CIFAR10의 train-set과 test-set이 나눠져 있지만 그 둘은 동일한 분포)

- Out of Distribution : 학습 데이터의 분포와 검증 데이터의 분포가 다른 경우(Ex. 특정 병원에서 얻은 데이터로 학습한 뒤, 다른 병원에 배포)

- 초록색 배경의 소와 모래색 배경의 낙타로 학습을 시킨 모델이 있을 때 초록색 배경 낙타를 추론 시킬 경우 소로 추론

- 즉, 기존의 학습 방법은 데이터간의 가장 큰 공통 특징(배경색)을 가지고 학습 및 추론을 하기 때문에 학습 데이터와 다른 데이터(OoD)은 잘 추론하지 못함



Domain Generalization

- “Domain generalization via invariant feature representation” 에 처음으로 DG 제안
- Domain adaptation vs Domain generalization
  - domain adaptation은 타겟 도메인은 알지만 label이 없는 도메인으로 일반화하는 것이 목표
  - domain generalization은 하나 이상의 도메인으로부터 타겟 도메인으로 접근하지 않고 domain-agnostic(도메인에 구애받지 않는) 모델을 학습시키는 과제
- Multi source domain generalization은 여러 개의 도메인을 동시에 학습하여 unseen 도메인에서 테스트하는 과제
- Single source domain generalization은 하나의 도메인에서 학습되고 unseen 도메인에서 테스트하는 과제

Empirical Risk Minimization in DG[2]

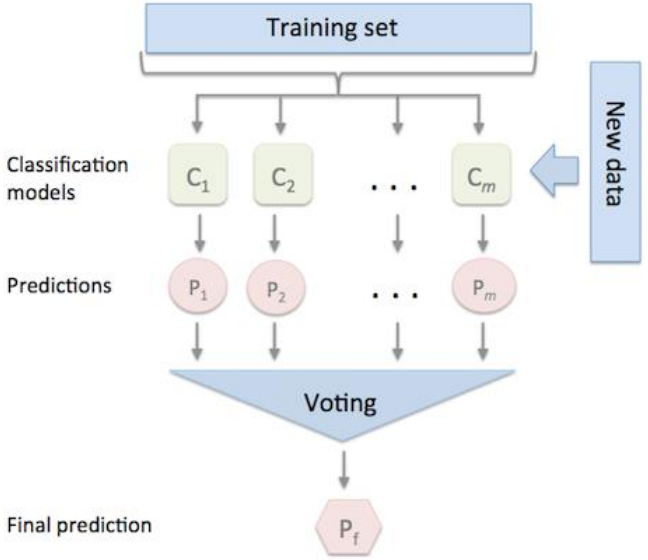
- ERM : training dataset 에서 loss를 최소화하는 방법
  - DG에서 ERM은 source data를 학습하여 target data를 잘 학습하는 방법
- DG에서 초기 ERM은 source 도메인들에 대해 각각 모델 1개씩 생성 => 모델들을 종합해 class의 general한 부분을 얻고자 함
- 하지만 ERM 방식은 domain에 관계 없는 feature를 뽑는 보장이 없기에 개선한 IRM 방식 도입
- IRM[1] : 모델이 특정 도메인에서 잘 수행되도록 최적화하는 대신 모든 도메인에서 동시에 잘 수행하도록 학습
  - IRM loss를 추가하여 regularize => 서로 다른 도메인이더라도 두 데이터 세트에서 동일하게 잘 수행하도록
  - IRM(θ) = max_{P∈𝒮} L_P(θ) / L_P(θ) : the empirical risk , 𝒮 : invariant subsets , perform equally well on all invariant subsets P

$$\min_{\substack{\Phi:\mathcal{X}\to\mathcal{H}\\ w:\mathcal{H}\to\mathcal{Y}}} \quad \sum_{e\in\mathcal{E}_{\text{tr}}} R^e(w\circ\Phi) \qquad \text{(IRM)}$$
$$\text{subject to} \quad w \in \arg\min_{\bar{w}:\mathcal{H}\to\mathcal{Y}} R^e(\bar{w}\circ\Phi), \ \text{for all } e\in\mathcal{E}_{\text{tr}}.$$

1. Arjovsky, M., Bottou, L., Gulrajani, I., & Lopez-Paz, D. (2019). Invariant risk minimization. arXiv preprint arXiv:1907.02893.
2. Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, *10*(5), 988-999.

Ensemble in DG

- 앙상블 : 여러 개의 분류기를 생성하고 그 예측을 결합함으로써 보다 정확한 예측을 도출하는 기법
- Domain-Specific Neural Networks : 다른 종류의 딥러닝 모델들을 동시에 학습시켜 예측 값을 앙상블 하는 방법
  - 모델의 다양성을 활용해 일반화 성능 향상
  - 전체 모델 대신 일부 layer를 특정 domain에 할당 1개의 모델로 앙상블 혹은 학습한 source domain에 대한 가중치를 기억해 앙상블
- Weight Averaging : 학습하는 동안 다양한 source에 대해 각각 학습한 모델의 weight를 종합하여 단일 모델 형성

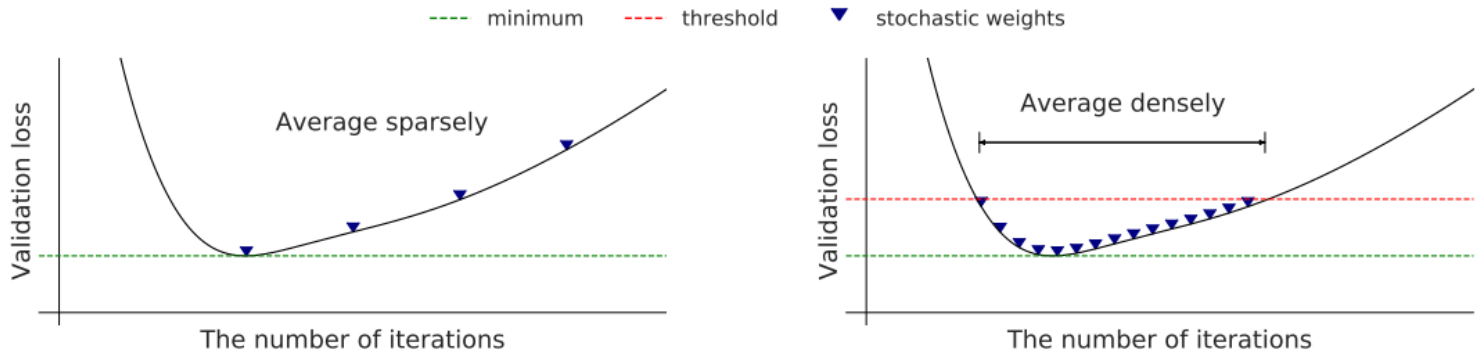


SWAD[*]

- SWA : 시간 축으로 모델을 Ensemble & 모델의 weight를 시간 축으로 여러 개 저장하는 대신 모델의 weight를 시간 축으로 누적(running average)시킨다는 점

$$w_{\text{SWA}} \leftarrow \frac{w_{\text{SWA}} \cdot n_{\text{models}} + w}{n_{\text{models}} + 1},$$

- SWAD : SWA 방식에서 threshold 미만의 지점들만 모델의 weight를 시간 축으로 누적(running average)



(a) SWA (b) SWAD (proposed)

Cha, J., Chun, S., Lee, K., Cho, H. C., Park, S., Lee, Y., & Park, S. (2021). Swad: Domain generalization by seeking flat minima. Advances in Neural Information Processing Systems, 34, 22405-22418.

Ensemble of Averages

- 해당 방식은 SWAD 방식에 대해서 모티브를 얻음
- 본 논문은 ERM에서 SWAD 방식처럼 moving average model을 사용해 앙상블 하여 성능 개선 수행
- Why DG work ? : Model averaging이 Tikhonov regularization(Ridge regression)와 유사하게 정규화 수행
- 본 논문은 다음과 같은 장점을 가짐
  1. Hyperparameter-free : iteration마다 생기는 weight들을 이동 평균 하는 것이니 추가적인 파라미터가 필요 없다.
  2. Computationally efficient : threshold를 비교하는 계산이 없으니 cost 적음
  3. EoA : DG 성능적으로 뛰어남(by 실험)
  4. Theoretical explanation

Method

- 각 도메인에 대해 모델을 학습하기 위해 source domain dataset을 random으로 k개의 fold로 나눔(도메인이 섞이지 않게)

- K개의 fold dataset에서 fold마다 각 모델을 학습
  - 이때, 특정 횟수까지 그대로 학습되고 특정 횟수(t_0)가 넘으면 다음 식을 이용해 모델 파라미터(θ) 업데이트
  - 학습을 끝까지 진행하면 validation performance가 가장 좋은 θ_t 생성

$$\hat{\theta}_t = \begin{cases} \theta_t, & \text{if } t \leq t_0 \\ \frac{t-t_0}{t-t_0+1} \cdot \hat{\theta}_{t-1} + \frac{1}{t-t_0+1} \cdot \theta_t, & \text{otherwise} \end{cases}$$

- Test 단계는 Moving average를 통해 학습 시킨 모델들을 최종적으로 앙상블 진행
  - E : 앙상블 하는 모델 개수, k : class 개수

$$\hat{y} = \arg\max_k Softmax(\frac{1}{E}\sum_{i=1}^{E} f(\mathbf{x}; \hat{\theta}_i))_k$$

Method

- 각 도메인에 대해 모델을 학습하기 위해 source domain dataset을 random으로 k개의 fold로 나눔(도메인이 섞이지 않게)

- K개의 fold dataset에서 fold마다 각 모델을 학습
  - 이때, 특정 횟수까지 그대로 학습되고 특정 횟수(t_0)가 넘으면 다음 식을 이용해 모델 파라미터(θ) 업데이트
  - 학습을 끝까지 진행하면 validation performance가 가장 좋은 θ_t 생성

$$\hat{\theta}_t = \begin{cases} \theta_t, & \text{if } t \leq t_0 \\ \frac{t-t_0}{t-t_0+1} \cdot \hat{\theta}_{t-1} + \frac{1}{t-t_0+1} \cdot \theta_t, & \text{otherwise} \end{cases}$$

- Test 단계는 Moving average를 통해 학습 시킨 모델들을 최종적으로 앙상블 진행
  - E : 앙상블 하는 모델 개수, k : class 개수

$$\hat{y} = \arg\max_k Softmax(\frac{1}{E}\sum_{i=1}^{E} f(\mathbf{x}; \hat{\theta}_i))_k$$

Experiments

- 각 subset에 대해 다양한 Resnet을 사용해서 다른 DG 모델들과 비교 결과 좋은 성능을 보임

Table 10: Performance benchmarking on 5 datasets of the DomainBed benchmark using two different pre-trained models. SWAD is the previous SOTA. Note that ensembles do not have confidence interval because an ensemble uses all the models to make a prediction. Gray background shows our proposal. *Our runs* implies we ran experiments, but we did not propose it.

Algorithm	PACS	VLCS	OfficeHome	TerraIncognita	DomainNet	Avg.
ResNet-50 (25M Parameters, pre-trained on ImageNet)						
ERM (our runs)	84.4 ± 0.8	77.1 ± 0.5	66.6 ± 0.2	48.3 ± 0.2	43.6 ± 0.1	64.0
Ensemble (our runs)	87.6	78.5	70.8	49.2	**47.7**	66.8
ERM [18]	85.7 ± 0.5	77.4 ± 0.3	67.5 ± 0.5	47.2 ± 0.4	41.2 ± 0.2	63.8
IRM [2]	84.4 ± 1.1	78.1 ± 0.0	66.6 ± 1.0	47.9 ± 0.7	35.7 ± 1.9	62.5
Group DRO [38]	84.1 ± 0.4	77.2 ± 0.6	66.9 ± 0.3	47.0 ± 0.3	33.7 ± 0.2	61.8
Mixup [47, 46]	84.3 ± 0.5	77.7 ± 0.4	69.0 ± 0.1	48.9 ± 0.8	39.6 ± 0.1	63.9
MLDG [28]	84.8 ± 0.6	77.1 ± 0.4	68.2 ± 0.1	46.1 ± 0.8	41.8 ± 0.4	63.6
CORAL [41]	86.0 ± 0.2	77.7 ± 0.5	68.6 ± 0.4	46.4 ± 0.8	41.8 ± 0.2	64.1
MMD [30]	85.0 ± 0.2	76.7 ± 0.9	67.7 ± 0.1	49.3 ± 1.4	39.4 ± 0.8	63.6
DANN [16]	84.6 ± 1.1	78.7 ± 0.3	65.4 ± 0.6	48.4 ± 0.5	38.4 ± 0.0	63.1
C-DANN [31]	82.8 ± 1.5	78.2 ± 0.4	65.6 ± 0.5	47.6 ± 0.8	38.9 ± 0.1	62.6
Fish [39]	85.5 ± 0.3	77.8 ± 0.3	68.6 ± 0.4	45.1 ± 1.3	42.7 ± 0.2	63.9
Fishr [37]	85.5 ± 0.4	77.8 ± 0.1	67.8 ± 0.1	47.4 ± 1.6	41.7 ± 0.0	65.7
SWAD [8]	88.1 ± 0.4	**79.1 ± 0.4**	70.6 ± 0.3	50.0 ± 0.4	46.5 ± 0.2	66.9
MIRO [9]	85.4 ± 0.4	79.0 ± 0.	70.5 ± 0.4	50.4 ± 1.1	44.3 ± 0.2	65.9
SMA (ours)	87.5 ± 0.2	78.2 ± 0.2	70.6 ± 0.1	50.3 ± 0.5	46 ± 0.1	66.5
EoA (ours)	**88.6**	**79.1**	**72.5**	**52.3**	47.4	**68.0**
ResNeXt-50 32x4d [48] (25M Parameters, Pre-trained 1B Images)						
ERM (our runs)	88.9 ± 0.3	79.0 ± 0.1	70.9 ± 0.5	51.4 ± 1.2	48.1 ± 0.2	67.7
Ensemble (our runs)	91.2	80.3	77.8	53.5	52.8	71.1
SMA (ours)	92.7 ± 0.3	79.7 ± 0.3	78.6 ± 0.1	53.3 ± 0.1	53.5 ± 0.1	71.6
EoA (ours)	**93.2**	**80.4**	**80.2**	**55.2**	**54.6**	**72.7**
RegNetY-16GF [40] (81M Parameters, Pre-trained on 3.6B Images)						
ERM (our runs)	92.0 ± 0.4	78.6 ± 0.6	73.8± 0.5	55.6 ± 0.9	53.2 ± 0.2	70.6
Ensemble (our runs)	95.1	80.6	80.5	59.5	57.8	74.7
ERM [9]	89.6 ± 0.4	78.6 ± 0.3	71.9 ± 0.6	51.4 ± 1.8	48.5 ± 0.6	68.0
SWAD [9]	94.7 ± 0.2	79.7 ± 0.2	80.0 ± 0.1	57.9 ± 0.7	53.6 ± 0.6	73.2
MIRO [9]	**97.4 ± 0.2**	79.9 ± 0.6	80.4 ± 0.2	58.9 ± 1.3	53.8 ± 0.1	74.1
SMA (ours)	95.5± 0.0	80.7 ± 0.1	82.0 ± 0.0	59.7 ± 0.0	60.0 ± 0.0	75.6
EoA (ours)	95.8	**81.1**	**83.9**	**61.1**	**60.9**	**76.6**

Experiments

- 앙상블 하는 모델의 개수에 관계 없이 이동 평균으로 weight를 업데이트 할 경우 항상 좋은 모습을 보임

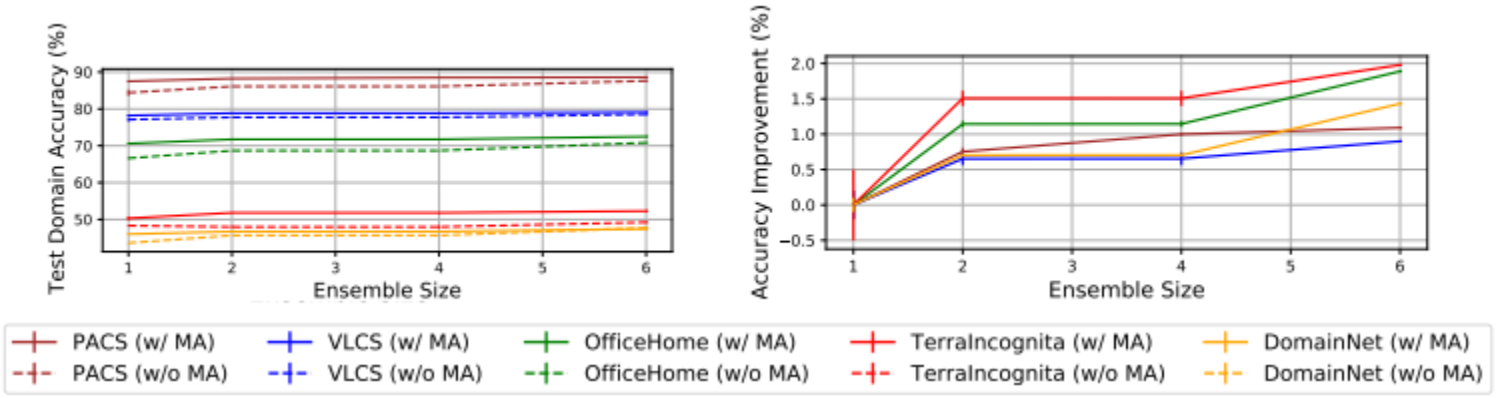


Figure 3: **Left**: Effect of ensemble size (number of models in an ensemble) on out-domain performance (mean and standard error) for models with and without moving average (MA) parameters for ResNet-50 pre-trained on ImageNet. **Right**: Using the performance of ensemble of size 1 (shown in the left plot) as reference, right plot shows the percentage point improvement for ensembles of size > 1. The plots show that i) ensemble of averages (solid lines in left plot) are consistently better than ensemble of models without averaging (dashed lines in left plot); ii) ensemble of averages consistently improves performance over averaged models (ensemble of size 1 in right plot).

Conclusion

- 본 논문은 다음과 같은 결론과 한계점을 가지고 있음
  - Domain Generalization Limitations
    - 딥러닝 모델은 source 도메인에 대해 학습하고 이를 domain alignment 하는 방식이 아니라는 방식이 존재함
    - 하지만 source domain에 대해 variance를 줄임으로써 성능적으로 개선할 수 있었음
  - Functional Diversity
    - 해당 방식은 기존의 모델을 보다 더 개선하는 방법이지 근본적인 방법은 아님(DLC 느낌)
  - Scalability
    - 도메인 개수가 엄청나게 많을 경우 활용이 불가 (각 도메인마다 모델을 만들 수 없기에!)
    - 특정 몇 개의 모델에 랜덤으로 domain subset을 학습시켜 업데이트 하는 확률론적 방법도 고안 중