Domain Generalization via Nuclear Norm Regularization

Zhenmei Shi*, Yifei Ming*, Ying Fan*, Frederic Sala, Yingyu Liang
University of Wisconsin-Madison

CPAL 2024
Intro - Domain Generalization

Train on multiple training domains, e.g., Sketch + Cartoon + Art.
Test on new/unseen domain, e.g., Photo.

Sketch

Cartoon

Art painting

Photo

Training set

Test set

PACS dataset

Source: Deeper, broader and artier domain generalization. ICCV 2017.
Intro - More Datasets

**DomainNet**
- 6 domains,
- 345 classes,
- 586,575 images

**OfficeHome**
- Bed, Bike, Bottle, Chair, Glasses

**Terra Incognita**
- 6 domains,
- 345 classes,
- 586,575 images

Source: In Search of Lost Domain Generalization. ICLR 2021.
Intro - Domain Labels

With domain labels

Domain generalization using a mixture of multiple latent domains

Without domain labels (ours)

Source: Domain Generalization Using a Mixture of Multiple Latent Domains. AAAI 2020.
Intro - Invariant/Spurious Feature

Waterbirds dataset
Invariant - Birds; Spurious - Background
Source: Avoiding spurious correlations via logit correction, ICLR 2023
Intro - Invariant/Spurious Feature

Waterbirds dataset
Invariant - Birds; Spurious - Background
Source: Avoiding spurious correlations via logit correction, ICLR 2023

Hidden representation data model:
- e : environment (background)
- y : label (bird)
- ze: spurious feature
- zc: invariant feature
- x : input (image)
Empirical Risk Minimization already learn features sufficient for domain generalization:

- **ERM**: train on training domains.
- **Linear**: train on training domains $\Rightarrow$ Linear Probing on unseen domain.
- **End-to-End**: train on training + unseen domain.

Evaluate on the unseen domain.

Source: Domain-adjusted regression or: Erm may already learn features sufficient for out-of-distribution generalization, 2023
Main Issue: features in ERM can be arbitrarily mixed: spurious features are hard to disentangle from invariant features.

Idea: low-dimensional (parsimonious) structures => minimal information retrieved from ERM solution from training domains by controlling the rank => avoid domain overfitting.

Hypothesis: spurious features have lower correlation with labels than invariant features.

Source: Domain-adjusted regression or: Erm may already learn features sufficient for out-of-distribution generalization, 2023
Question:
Can ERM benefit from rank regularization of the extracted feature for better domain generalization?

Answer:
Yes, ERM with Nuclear Norm Regularization (ERM-NU). Nuclear norm is convex envelope to the rank function.
Method - Setup

Hidden representation data model:
- $e$: environment (background)
- $y$: label (bird)
- $z_e$: spurious feature
- $z_c$: invariant feature
- $x$: input (image)

$a$: linear head
$\Phi$: feature extractor, e.g., ResNet 50
Method - Objective Function

$$\min_{a, \Phi} \mathcal{L}(a, \Phi) + \lambda \| \Phi(X) \|_*$$

ERM

NU

NU can select a subset of ERM solutions that extract the smallest possible information for classification => reduce the effect of spurious features for better generalization.
Experiment - Simulation

In-distribution (ID) / training:
- $x_1$ and $y$ has strong correlation.
- $x_2$ and $y$ has weak correlation.

Out-of-distribution (OOD) / unseen:
- $x_1$ and $y$ has the same correlation.
- $x_2$ and $y$ change the correlation.

NU significantly reduces the OOD error rate, while keep small ID error.
## SWAD: Domain Generalization by Seeking Flat Minima

NU is effective.

### Table: Performance on Real Dataset

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>VLCS</th>
<th>PACS</th>
<th>OfficeHome</th>
<th>Terralnc</th>
<th>DomainNet</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMD* (CVPR 18) [10]</td>
<td>77.5 ± 0.9</td>
<td>84.6 ± 0.5</td>
<td>66.3 ± 0.1</td>
<td>42.2 ± 1.6</td>
<td>23.4 ± 9.5</td>
<td>58.8</td>
</tr>
<tr>
<td>Mixstyle† (ICLR 21) [27]</td>
<td>77.9 ± 0.5</td>
<td>85.2 ± 0.3</td>
<td>60.4 ± 0.3</td>
<td>44.0 ± 0.7</td>
<td>34.0 ± 0.1</td>
<td>60.3</td>
</tr>
<tr>
<td>GroupDRO† (ICLR 19) [28]</td>
<td>76.7 ± 0.6</td>
<td>84.4 ± 0.8</td>
<td>66.0 ± 0.7</td>
<td>43.2 ± 1.1</td>
<td>33.3 ± 0.2</td>
<td>60.7</td>
</tr>
<tr>
<td>IRM† (ArXiv 20) [6]</td>
<td>78.5 ± 0.5</td>
<td>83.5 ± 0.8</td>
<td>64.3 ± 2.2</td>
<td>45.5 ± 0.3</td>
<td>35.5 ± 0.2</td>
<td>61.6</td>
</tr>
<tr>
<td>ARM† (ArXiv 20) [29]</td>
<td>77.6 ± 0.3</td>
<td>85.1 ± 0.4</td>
<td>64.8 ± 0.3</td>
<td>45.5 ± 0.3</td>
<td>35.5 ± 0.2</td>
<td>61.7</td>
</tr>
<tr>
<td>VREx† (ICML 21) [14]</td>
<td>78.3 ± 0.2</td>
<td>84.9 ± 0.6</td>
<td>66.4 ± 0.6</td>
<td>46.4 ± 0.6</td>
<td>33.6 ± 2.9</td>
<td>61.9</td>
</tr>
<tr>
<td>CDANN† (ECCV 18) [8]</td>
<td>77.5 ± 0.1</td>
<td>82.6 ± 0.9</td>
<td>65.8 ± 1.3</td>
<td>45.8 ± 0.3</td>
<td>38.3 ± 0.3</td>
<td>62.0</td>
</tr>
<tr>
<td>AND-mask* (ICLR 20) [30]</td>
<td>78.1 ± 0.9</td>
<td>84.4 ± 0.9</td>
<td>65.6 ± 0.4</td>
<td>44.6 ± 0.3</td>
<td>37.2 ± 0.6</td>
<td>62.0</td>
</tr>
<tr>
<td>DANN† (JMLR 16) [7]</td>
<td>78.6 ± 0.4</td>
<td>83.6 ± 0.4</td>
<td>65.9 ± 0.6</td>
<td>46.7 ± 0.5</td>
<td>38.3 ± 0.1</td>
<td>62.6</td>
</tr>
<tr>
<td>RSC† (ECCV 20) [31]</td>
<td>77.1 ± 0.5</td>
<td>85.2 ± 0.9</td>
<td>65.9 ± 0.6</td>
<td>46.7 ± 0.5</td>
<td>38.3 ± 0.1</td>
<td>62.6</td>
</tr>
<tr>
<td>MTL† (JMLR 21) [32]</td>
<td>77.2 ± 0.4</td>
<td>84.6 ± 0.5</td>
<td>66.4 ± 0.5</td>
<td>45.6 ± 1.2</td>
<td>40.6 ± 0.1</td>
<td>62.9</td>
</tr>
<tr>
<td>Mixup† (ICLR 18) [1]</td>
<td>77.4 ± 0.6</td>
<td>84.6 ± 0.6</td>
<td>68.1 ± 0.3</td>
<td>47.9 ± 0.8</td>
<td>39.2 ± 0.1</td>
<td>63.4</td>
</tr>
<tr>
<td>MLDG† (AAA18) [33]</td>
<td>77.2 ± 0.4</td>
<td>84.9 ± 1.0</td>
<td>66.8 ± 0.6</td>
<td>47.7 ± 0.9</td>
<td>41.2 ± 0.1</td>
<td>63.6</td>
</tr>
<tr>
<td>Fish (ICLR 22) [34]</td>
<td>77.8 ± 0.3</td>
<td>85.5 ± 0.3</td>
<td>68.6 ± 0.4</td>
<td>45.1 ± 1.3</td>
<td>42.7 ± 0.2</td>
<td>63.9</td>
</tr>
<tr>
<td>Fishr* (ICML 22) [35]</td>
<td>77.8 ± 0.1</td>
<td>85.5 ± 0.4</td>
<td>67.8 ± 0.1</td>
<td>47.4 ± 1.6</td>
<td>41.7 ± 0.0</td>
<td>64.0</td>
</tr>
<tr>
<td>SagNet† (CVPR 21) [36]</td>
<td>77.8 ± 0.5</td>
<td>86.3 ± 0.2</td>
<td>68.1 ± 0.1</td>
<td>48.6 ± 1.0</td>
<td>40.3 ± 0.1</td>
<td>64.2</td>
</tr>
<tr>
<td>SelfReg (ICCV 21) [37]</td>
<td>77.8 ± 0.9</td>
<td>89.6 ± 0.3</td>
<td>67.9 ± 0.7</td>
<td>47.6 ± 1.0</td>
<td>41.5 ± 0.2</td>
<td>64.5</td>
</tr>
<tr>
<td>CORAL† (ECCV 16) [9]</td>
<td>78.8 ± 0.6</td>
<td>86.2 ± 0.3</td>
<td>68.7 ± 0.3</td>
<td>47.6 ± 1.0</td>
<td>41.5 ± 0.1</td>
<td>64.5</td>
</tr>
<tr>
<td>SAM† (ICLR 21) [38]</td>
<td>79.4 ± 0.1</td>
<td>85.8 ± 0.2</td>
<td>69.6 ± 0.1</td>
<td>43.3 ± 0.7</td>
<td>44.3 ± 0.0</td>
<td>64.5</td>
</tr>
<tr>
<td>mSDSI (NeurIPS 21) [39]</td>
<td>79.0 ± 0.3</td>
<td>86.2 ± 0.2</td>
<td>69.2 ± 0.4</td>
<td>48.1 ± 1.4</td>
<td>42.8 ± 0.1</td>
<td>65.1</td>
</tr>
<tr>
<td>MIRO (ECCV 22) [40]</td>
<td>79.0 ± 0.0</td>
<td>85.4 ± 0.4</td>
<td>70.5 ± 0.4</td>
<td>50.4 ± 1.1</td>
<td>44.3 ± 0.2</td>
<td>65.9</td>
</tr>
<tr>
<td>ERM† [41]</td>
<td>77.5 ± 0.4</td>
<td>85.5 ± 0.2</td>
<td>66.5 ± 0.3</td>
<td>46.1 ± 1.8</td>
<td>40.9 ± 0.1</td>
<td>63.3</td>
</tr>
<tr>
<td>ERM-NU (ours)</td>
<td>78.3 ± 0.3</td>
<td>85.6 ± 0.1</td>
<td>68.1 ± 0.1</td>
<td>49.6 ± 0.6</td>
<td>43.4 ± 0.1</td>
<td>65.0</td>
</tr>
<tr>
<td>SWAD† (NeurlPS 21) [24]</td>
<td>79.1 ± 0.1</td>
<td>88.1 ± 0.1</td>
<td>70.6 ± 0.2</td>
<td>50.0 ± 0.3</td>
<td>46.5 ± 0.1</td>
<td>66.9</td>
</tr>
<tr>
<td>SWAD-NU (ours)</td>
<td>79.8 ± 0.2</td>
<td>88.5 ± 0.2</td>
<td>71.3 ± 0.3</td>
<td>52.2 ± 0.3</td>
<td>47.1 ± 0.1</td>
<td>67.8</td>
</tr>
</tbody>
</table>
NU is broadly applicable.

```
import torch
import torch.nn.functional as F

class NUModel(nn.Module):
    def forward(self, x, y):
        f = self.feature(x)  # get feature embedding
        loss = F.cross_entropy(self.classifier(f), y)  # get classification loss
        _, s, _ = torch.svd(f)  # singular value decomposition
        loss += self.lambda_ * torch.sum(s)  # add nuclear norm regularization
        return loss
```
Theoretical Analysis

Theorem (Informal; Linear data and linear model)
- The optimal solution for the ERM-NU has high OOD test accuracy.
- The optimal solution for the ERM with/without weight decay has low OOD test accuracy (like random guessing).

Proof Intuition:
1. ERM will encode all features correlated with labels, even when the correlation is weak (logistic or cross-entropy loss).
2. Larger correlation with label => stronger feature encoding.
3. When OOD has different spurious feature distributions => ERM fails (random guessing).
4. However, ERM-NU will only encode features that have a large correlation with labels (invariant features) => high OOD test accuracy.
Take Home Message

Nuclear Norm Regularization is an
1. effective
2. broadly applicable
3. easy to implement
method for domain generalization.

Q&A
Thanks!

CPAL 2024