In Search of Lost Domain Generalization

Elnikov Vladislav

О чем данная статья

- Данная статья рассказывает, как обучать модель на известных доменах, чтобы та справлялась с работой на других доменах.
- Рекомендации как сравнивать данные модели.
- Получать модели более устойчивые к изменению данных.
- Рекомендации по использованию данных моделей и для их дальнейшего изучения.
- Сравнение существующих алгоритмов.

Почему это важно

train

test

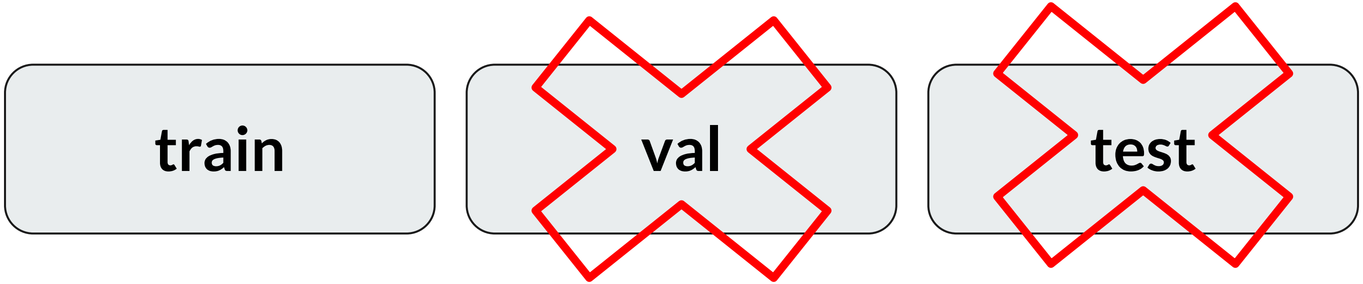
Формальная постановка задачи

Learning setups. L^d and U^d denote the labeled and unlabeled distributions from domain d.

Setup	**Training inputs**	**Test inputs**
Generative learning	U^1	$\emptyset$
Unsupervised learning	U^1	U^1
Supervised learning	L^1	U^1
Semi-supervised learning	L^1, U^1	U^1
Multitask learning	$L^1, \ldots, L^{d_{\mathrm{tr}}}$	$U^1, \ldots, U^{d_{\mathrm{tr}}}$
Continual (or lifelong) learning	$L^1, \ldots, L^{\infty}$	$U^1, \ldots, U^{\infty}$
Domain adaptation	$L^1, \ldots, L^{d_{\mathrm{tr}}}, U^{d_{\mathrm{tr}}+1}$	$U^{d_{\mathrm{tr}}+1}$
Transfer learning	$U^1, \ldots, U^{d_{\mathrm{tr}}}, L^{d_{\mathrm{tr}}+1}$	$U^{d_{\mathrm{tr}}+1}$
Domain generalization	$L^1, \ldots, L^{d_{\mathrm{tr}}}$	$U^{d_{\mathrm{tr}}+1}$

Отличие от обычного обучения

- Непонятно как подбирать гиперпараметры
- Тяжело сравнить работу двух моделей



Рекомендации от авторов

Recommendation 1 *A domain generalization algorithm should be responsible for specifying a model selection method.*

Recommendation 2 *Researchers should disclaim any oracle-selection results as such and specify policies to limit access to the test domain.*

Гитхаб от авторов

https://github.com/facebookresearch/DomainBed

- Дает датасеты для изучения новых моделей
- OpenSource
- Метрики
- Модели

GitHub

Running unit tests

DomainBed includes some unit tests and end-to-end tests. While not exhaustive, but they are a good sanity-check. To run the tests:

```
python -m unittest discover
```

By default, this only runs tests which don't depend on a dataset directory. To run those tests as well:

```
DATA_DIR=/my/datasets/path python -m unittest discover
```

README.md

- `OracleSelectionMethod`: A random subset from the data of the test domain.

Quick start

Download the datasets:

```
python3 -m domainbed.scripts.download \
       --data_dir=./domainbed/data
```

Train a model:

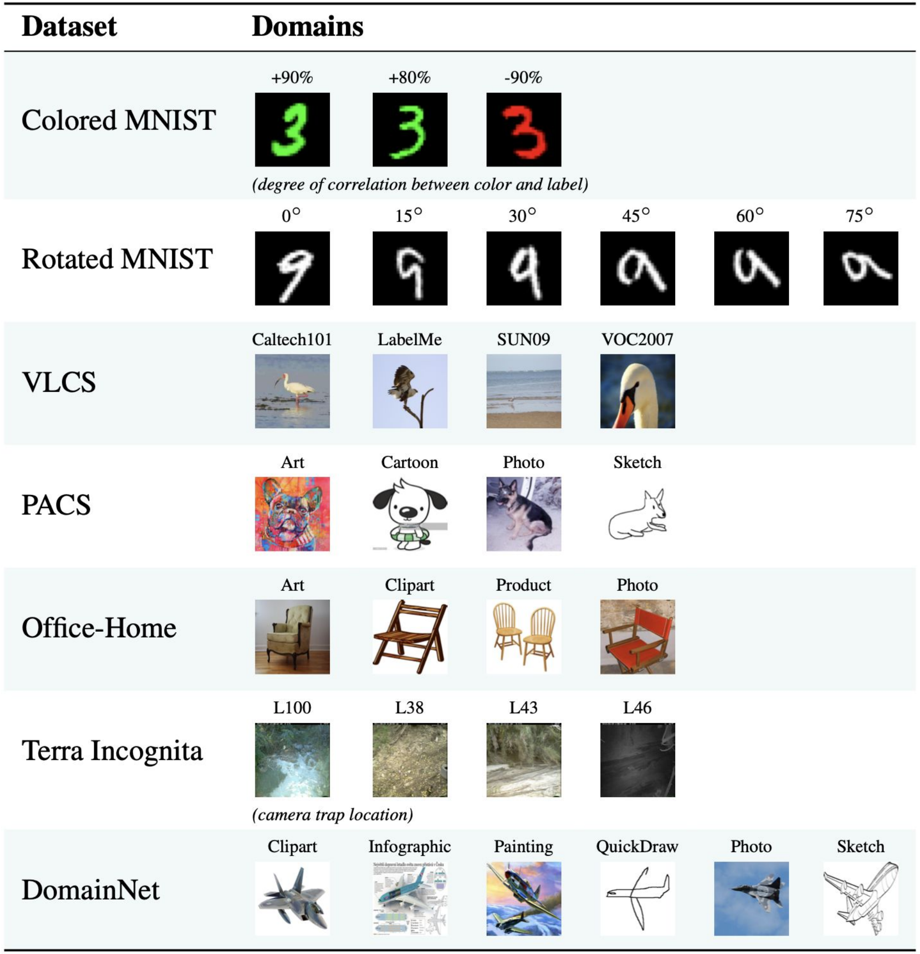
```
python3 -m domainbed.scripts.train\
       --data_dir=./domainbed/data/MNIST/\
       --algorithm IGA\
       --dataset ColoredMNIST\
       --test_env 2
```

Launch a sweep:

```
python -m domainbed.scripts.sweep launch\
       --data_dir=/my/datasets/path\
       --output_dir=/my/sweep/output/path\
       --command_launcher MyLauncher
```

Here, `MyLauncher` is your cluster's command launcher, as implemented in `command_launchers.py`. At the time of writing, the entire sweep trains tens of thousands of models (all algorithms x all datasets x 3 independent trials x 20 random hyper-parameter choices). You can pass arguments to make the sweep smaller:

Датасеты от авторов статьи



Алгоритмы, которые сравнивали

- Empirical Risk Minimization (**ERM**, Vapnik [1998]) minimizes the sum of errors across domains and examples.
- Group Distributionally Robust Optimization (**DRO**, Sagawa et al. [2019]) performs ERM while increasing the importance of domains with larger errors.
- Inter-domain Mixup (**Mixup**, Xu et al. [2019], Yan et al. [2020], Wang et al. [2020]) performs ERM on linear interpolations of examples from random pairs of domains and their labels.
- Meta-Learning for Domain Generalization (**MLDG**, Li et al. [2018a]) leverages MAML [Finn et al., 2017] to meta-learn how to generalize across domains.
- Different variants of the popular algorithm of Ganin et al. [2016] to learn features $\phi(X^d)$ with distributions matching across domains:
  - Domain-Adversarial Neural Networks (**DANN**, Ganin et al. [2016]) employ an adversarial network to match feature distributions.
  - Class-conditional DANN (**C-DANN**, Li et al. [2018d]) is a variant of DANN matching the conditional distributions $P(\phi(X^d)|Y^d = y)$ across domains, for all labels y.
  - **CORAL** [Sun and Saenko, 2016] matches the mean and covariance of feature distributions.
  - **MMD** [Li et al., 2018b] matches the MMD [Gretton et al., 2012] of feature distributions.
- Invariant Risk Minimization (**IRM** [Arjovsky et al., 2019]) learns a feature representation $\phi(X^d)$ such that the optimal linear classifier on top of that representation matches across domains.

Выбор модели (гиперпараметров)

- **Training-domain validation set:**
  Обучаемся на всех известных доменах и их же используем как валидацию
- **Leave-one-domain-out cross-validation:**
  Выбираем один домен и обучаемся без него. Смотрим на ошибку на оставшемся домене.
- **Test-Domain Validation Set:**
  Когда у нас можно посмотреть тестовую выборку. Мы обращаемся максимум раз в 20 итераций к информации теста, чтобы узнать на нем качество.

Эксперименты

Model selection method: training domain validation set

Algorithm	CMNIST	RMNIST	VLCS	PACS	Office-Home	TerraInc	DomainNet	Avg
ERM	52.0 ± 0.1	98.0 ± 0.0	77.4 ± 0.3	85.7 ± 0.5	67.5 ± 0.5	47.2 ± 0.4	41.2 ± 0.2	67.0
IRM	51.8 ± 0.1	97.9 ± 0.0	78.1 ± 0.0	84.4 ± 1.1	66.6 ± 1.0	47.9 ± 0.7	35.7 ± 1.9	66.0
DRO	52.0 ± 0.1	98.1 ± 0.0	77.2 ± 0.6	84.1 ± 0.4	66.9 ± 0.3	47.0 ± 0.3	33.7 ± 0.2	65.5
Mixup	51.9 ± 0.1	98.1 ± 0.0	77.7 ± 0.4	84.3 ± 0.5	69.0 ± 0.1	48.9 ± 0.8	39.6 ± 0.1	67.1
MLDG	51.6 ± 0.1	98.0 ± 0.0	77.1 ± 0.4	84.8 ± 0.6	68.2 ± 0.1	46.1 ± 0.8	41.8 ± 0.4	66.8
CORAL	51.7 ± 0.1	98.1 ± 0.1	77.7 ± 0.5	86.0 ± 0.2	68.6 ± 0.4	46.4 ± 0.8	41.8 ± 0.2	67.2
MMD	51.8 ± 0.1	98.1 ± 0.0	76.7 ± 0.9	85.0 ± 0.2	67.7 ± 0.1	49.3 ± 1.4	39.4 ± 0.8	66.8
DANN	51.5 ± 0.3	97.9 ± 0.1	78.7 ± 0.3	84.6 ± 1.1	65.4 ± 0.6	48.4 ± 0.5	38.4 ± 0.0	66.4
C-DANN	51.9 ± 0.1	98.0 ± 0.0	78.2 ± 0.4	82.8 ± 1.5	65.6 ± 0.5	47.6 ± 0.8	38.9 ± 0.1	66.1

Model selection method: Leave-one-domain-out cross-validation

Algorithm	CMNIST	RMNIST	VLCS	PACS	Office-Home	TerraInc	DomainNet	Avg
ERM	34.2 ± 1.2	98.0 ± 0.0	76.8 ± 1.0	83.3 ± 0.6	67.3 ± 0.3	46.2 ± 0.2	40.8 ± 0.2	63.8
IRM	36.3 ± 0.4	97.7 ± 0.1	77.2 ± 0.3	82.9 ± 0.6	66.7 ± 0.7	44.0 ± 0.7	35.3 ± 1.5	62.9
DRO	32.2 ± 3.7	97.9 ± 0.1	77.5 ± 0.1	83.1 ± 0.6	67.1 ± 0.3	42.5 ± 0.2	32.8 ± 0.2	61.8
Mixup	31.2 ± 2.1	98.1 ± 0.1	78.6 ± 0.2	83.7 ± 0.9	68.2 ± 0.3	46.1 ± 1.6	39.4 ± 0.3	63.6
MLDG	36.9 ± 0.2	98.0 ± 0.1	77.1 ± 0.6	82.4 ± 0.7	67.6 ± 0.3	45.8 ± 1.2	42.1 ± 0.1	64.2
CORAL	29.9 ± 2.5	98.1 ± 0.1	77.0 ± 0.5	83.6 ± 0.6	68.6 ± 0.2	48.1 ± 1.3	41.9 ± 0.2	63.9
MMD	42.6 ± 3.0	98.1 ± 0.1	76.7 ± 0.9	82.8 ± 0.3	67.1 ± 0.5	46.3 ± 0.5	39.3 ± 0.9	64.7
DANN	29.0 ± 7.7	89.1 ± 5.5	77.7 ± 0.3	84.0 ± 0.5	65.5 ± 0.1	45.7 ± 0.8	37.5 ± 0.2	61.2
C-DANN	31.1 ± 8.5	96.3 ± 1.0	74.0 ± 1.0	81.7 ± 1.4	64.7 ± 0.4	40.6 ± 1.8	38.7 ± 0.2	61.1

Model selection method: Test-domain validation set *(oracle)*

Algorithm	CMNIST	RMNIST	VLCS	PACS	Office-Home	TerraInc	DomainNet	Avg
ERM	58.5 ± 0.3	98.1 ± 0.1	77.8 ± 0.3	87.1 ± 0.3	67.1 ± 0.5	52.7 ± 0.2	41.6 ± 0.1	68.9
IRM	70.2 ± 0.2	97.9 ± 0.0	77.1 ± 0.2	84.6 ± 0.5	67.2 ± 0.8	50.9 ± 0.4	36.0 ± 1.6	69.2
DRO	61.2 ± 0.6	98.1 ± 0.0	77.4 ± 0.6	87.2 ± 0.4	67.7 ± 0.4	53.1 ± 0.5	34.0 ± 0.1	68.4
Mixup	58.4 ± 0.2	98.0 ± 0.0	78.7 ± 0.4	86.4 ± 0.2	68.5 ± 0.5	52.9 ± 0.3	40.3 ± 0.3	69.0
MLDG	58.4 ± 0.2	98.0 ± 0.1	77.8 ± 0.4	86.8 ± 0.2	67.4 ± 0.2	52.4 ± 0.3	42.5 ± 0.1	69.1
CORAL	57.6 ± 0.5	98.2 ± 0.0	77.8 ± 0.1	86.9 ± 0.2	68.6 ± 0.4	52.6 ± 0.6	42.1 ± 0.1	69.1
MMD	63.4 ± 0.7	97.9 ± 0.1	78.0 ± 0.4	87.1 ± 0.5	67.0 ± 0.2	52.7 ± 0.2	39.8 ± 0.7	69.4
DANN	58.3 ± 0.2	97.9 ± 0.0	80.1 ± 0.6	85.4 ± 0.7	65.6 ± 0.3	51.6 ± 0.6	38.3 ± 0.1	68.2
C-DANN	62.0 ± 1.1	97.8 ± 0.1	80.2 ± 0.1	85.7 ± 0.3	65.6 ± 0.3	51.0 ± 1.0	38.9 ± 0.1	68.7

Выводы о статье

- \+ Рассказывает о проблеме сравненния работы моделей
- \+ Предлагает решения для научного сообщества и среду для решения, хороший гитхаб
- \+ Показывает, что базовый алгоритм хорошо себя показывает при равных условиях
- \- Меньше внимания уделенно алгоритму, не предложено чего-то нового