Robustness of Few-Shot Learning under Domain Shift

Sebastian Bischoff
sebastian.bischoff@epfl.ch
EPFL/TUM

Sitian Li
sitian.li@epfl.ch
EPFL

Adrian Ziegler
adrian.ziegler@epfl.ch
EPFL/TUM

1 . EXPERIMENTS

The aim of our experiments is to evaluate the performance of popular state-of-the-art few-shot learning methods under domain shift. To be more specific, we use transfer-learning as a baseline and evaluate ProtoNet and RelationNet as popular meta-learning techniques for few-shot learning. Additionally, we also propose our own method extending ProtoNet. We will stick to the terminology for few-shot learning used by Chen et al. [1].

1.1 Dataset

A domain shift is present when the novel classes belong to a different data distribution than the base classes. This is a sensible scenario for few-shot learning, as we want algorithms that are able to leverage knowledge about one domain to learn something about a different domain more efficiently.

In order to have a flexible amount of domain-shift scenarios we propose to use CIFAR-FS as a data set. CIFAR-FS [2] is based on CIFAR-100 [3] and is a well-established dataset for few-shot learning. It includes 100 classes and 600 examples for each class. The images are RGB and have a resolution of 32x32.

We propose to leverage that CIFAR-FS does not only come with class labels but also superclass labels for each class to construct domain-shifted datasets. By taking classes from one superclass as base classes and classes from a different superclass as novel classes we can construct arbitrary combinations of domain-shifted datasets. Furthermore, we can prevent possible errors due to a difference in data formats or quality as both, novel and base classes, come from the same data set. Thus, we make sure that the domain shift is only caused due to different features exhibited by the data e.g. vehicles may populate a different manifold than aquatic mammals.

For the experiments, we construct six different datasets. All datasets have seven classes as base classes, three as validation classes and three as novel classes resulting in three-way n-shot learning. As CIFAR-FS only has five classes per superclass, we grouped two similar superclasses together for the base and validation set to reach the required amount of ten classes. The *plants* superclass consists out of the *flower* and *tree* superclasses. The *land animals* superclass consists out of the *large carnivores* and *large omnivores* superclasses. The *outdoor scenes* superclass consists out of the *large natural outdoor scenes* and *large man made outdoor things* superclasses. All other referred-to superclasses are kept as in the description of the CIFAR-100 data set[1]. We designed a dataset with no domain-shift that has classes mixed from the trees , plants and fruits superclasses. We use this dataset as a baseline in order to be able to compare to a domain shift setting.

1.2 Methods

Baseline. We use a transfer learning method as baseline: A feature extractor is first trained using the base dataset. In the second phase, the feature extractor is fixed and a new classifier is trained on the novel set. We follow the model in Chen et al. [1].

Baseline++. The only difference to the baseline is that the linear layer in the classifier is replaced with a cosine distance metric.

RelationNet. RelationNet trains a feature extractor on the training data and trains a new classifier for the novel set. Refer to Sung et al. [4] for more details. If a domain shift is present, the authors of Chen et al. [1] implement an adaptation method which works by randomly splitting the few training samples in the novel classes into 3 support and 2 query samples to fine-tune the relation module for 100 epochs.

ProtoNet. Prototypical Network by Snell et al. [5] also proposes a metric-based few-shot learning method, which learns a metric

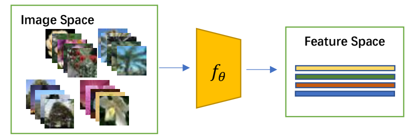

Figure 1: First part of ProtoNet, the feature extractor maps images from pixel domain to feature domain.

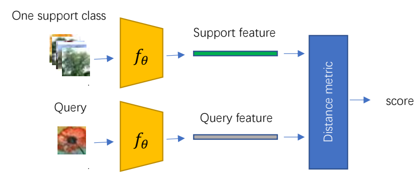

Figure 2: Second part of ProtoNet, given a feature extractor, the average of feature maps for each support set is first computed, and compared with the feature of query images.

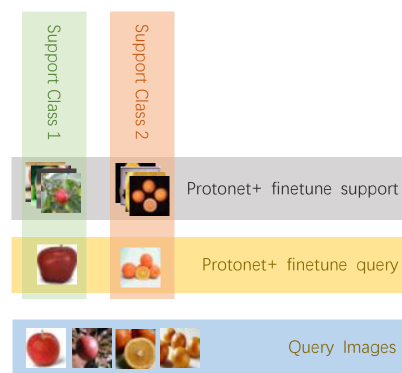

Figure 3: Data splitting for ProtoNet+ fine-tuning

[1]https://www.cs.toronto.edu/ kriz/cifar.html

for image classification. The classifier for ProtoNet is simply an euclidean distance measure combined with a cross-entropy layer. In the meta-testing stage, the support set is fed into the feature extractor to acquire the target feature representation for each support class, then the query image's features are extracted as well for later comparison with the support feature. There is also an adaptation proposed by Chen et al. [1] if an domain shift is present, which replaces the distance metric with a linear softmax classifier.

As shown in Figure 1, the feature extractor f_θ (backbone) first learns to map images from the image domain to the feature domain, such that the the features of images in one class are not too distant from each other. For the testing stage, the feature extractor is fixed, as shown in Figure 2, and the distances between the query image features and the features of each image in the support set are compared. The query image is labeled with the class which has the highest score based on the previously calculated distances. In the case of domain adaptation, the distance metric is abandoned, and a new linear classifier is trained using the support set in the new domain.

ProtoNet+. The common feature of the previously mentioned methods is that the feature extractor is fixed, and will not be updated for the novel set. Inspired by other optimization based methods like MAML [6], which fine-tune the complete network, we expect that the feature extractor can be fine-tuned according to the novel set as well.

If there is a big domain shift, the feature extractor of Protonet may not be able to extract the necessary features for the new domain, since it is trained using the original domain. To deal with this issue, we propose a new adaptation method based on ProtoNet, called **ProtoNet+**. Instead of preventing updates to the feature extractor and only training a new classifier, we keep the distance metric fixed, but update the weights in the feature extractor with support images in the new domain. Our hypothesis is that we are able to fine-tune the features trained on the old domain using only a small portion of the support set. For the test phase, we first split the n-shot support set into two parts, namely the $(n-1)$-shot fine-tune support image and the 1 query image for each set, as shown in Figure 3, and run several epochs to update the feature extractor according to the new features in the novel sets. In the end, we test the accuracy with the query images. We expect ProtoNet+ to show a better performance than ProtoNet and possibly also a better performance than ProtoNet with adaptation.

1.3 Implementation

In the following, we implement and test these methods on different domain-shifts as mentioned in Section 1.1. We implement our experiments based on the code by Chen et al. [1][2] and additionally include CIFAR-FS [2] and Protonet+[3]. We use Pytorch for training. For each dataset, we train for 100 epochs, and test on the novel set with best model based on the validation set. We use Adam optimizer, and train the Conv4 neural network with a GPU from Colab notebook. We test 1-shot, 2-shot and 5-shot for the 3-way classification problem.

[2] https://github.com/wyharveychen/CloserLookFewShot
[3] https://github.com/Baschdl/CloserLookFewShot

Theoretically one could use the trained models from ProtoNet and use them for ProtoNet+ but in our implementation one has to retrain them, resulting in different validation accuracy for ProtoNet and ProtoNet+ in Table 2 and 3.

1.4 Results

We report the validation accuracy in Table 1, 2 and 3 to be able to compare how well the different methods were able to learn the domain-shift problem. The methods are trained on based on the data split as described in Section 1.1. The few-shot accuracy is reported on three classes not included in the training or validation set. Chen et al. [1] proposed adaptation for different methods in the domain shift setting which is reported as few-shot adaptation accuracy.

All the experiments highlight that the performance of all discussed method is significantly impacted by a domain shift. Only the performance on the plant to fruit dataset is close to the no domain shift baseline. The validation accuracies in Table 1, 2 and 3 show that there are some superclasses that are significantly harder to learn than others. For instance the validation accuracies for all methods on the *plant* superclass are always higher than the ones for *land animals*.

In Table 2 and 3 we see that ProtoNet+ outperforms all other ProtoNet configurations in 7 out of 10 domain shifted data sets. Thus, we find that making it possible to fine-tune the extracted features, does indeed help in a domain-shift scenario.

In contrast to Chen et al. [1], Baseline++ doesn't perform consistently better than the Baseline as can be see in Table 1, 2 and 3. Baseline++ aims at reducing intra-class variations compared to the baseline. However, our experiments suggest that this isn't a good additional objective in the domain shift case as the overall performance does even decrease in comparison to Baseline.

Furthermore, we see that adapting ProtoNet doesn't work well. It performs on average similar to ProtoNet without adaptation, as shown in Table 1, 2 and 3. These observations are also consist with the experiment results shown in Figure 5 of Chen et al. [1] for a domain shift from CUB to mini-ImageNet. There could be two reasons for that. One reason is that the metric measure is dropped and replaced with a linear classifier, which could destroy the original behavior of ProtoNet. The other possible reason is that the old feature extractor fails extract relevant features for the new domain.

The adaptation of RelationNet seems to work better than the adaptation of ProtoNet. Compared to ProtoNet, the relation module of RelationNet is learnable. So in the adaptation phase, the relation module is kept and updated by novel set, which means the structure of the network is not destroyed.

1.5 Conclusion

In this work, we focused on the influence of domain shift on different few-shot learning methods. We proposed six datasets based on CIFAR-FS to test the adaptability of several meta-learning methods to domain shifts using transfer learning methods as baselines. Based on existing methods, we also proposed our own domain-shift adaptation strategy, ProtoNet+. With our experiments, we compared the performance of these methods, and showed that our method can outperform existing methods in certain domain shift scenarios.

		CIFARFS No domain shift	plant -> fruit	Land animals -> fish	Land animals ->insects	Land animals -> aquatic mammals	Outdoor scenes and object ->vehicles
Baseline	validation acc.	-	-	-	-	-	-
	few-shot acc.	53.9% (0.8%)	44.4% (0.7%)	38.8% (0.6%)	41.4% (0.7%)	45.2% (0.7%)	39.7% (0.6%)
Baseline++	validation acc.	-	-	-	-	-	-
	few-shot acc.	65.4% (1.0%)	38.0% (0.7%)	36.4% (0.6%)	38.8% (0.7%)	37.4% (0.6%)	37.0% (0.7%)
relationnet	validation acc.	64.0% (2.2%)	65.7% (2.4%)	43.4% (1.8%)	45.3% (2.0%)	51.5% (1.9%)	55.4% (2.3%)
	few-shot acc.	72.8% (1.3%)	41.1% (0.7%)	36.8% (0.7%)	37.3% (0.7%)	36.7% (0.6%)	34.5% (0.6%)
	few-shot (adapt) acc.	not applic.	not applic.	not applic.	not applic.	not applic.	not applic.
Protonet	validation acc.	64.9% (2.1%)	74.0% (2.3%)	43.4% (1.9%)	43.7% (1.6%)	54.0% (1.9%)	56.6% (2.4%)
	few-shot acc.	69.2% (0.9%)	38.6% (0.6%)	39.1% (0.7%)	42.9% (0.7%)	41.0% (0.6%)	38.5% (0.6%)
	few-shot (adapt) acc.	69.1% (0.9%)	39.6% (0.7%)	38.4% (0.7%)	42.4% (0.7%)	39.4% (0.6%)	38.6% (0.6%)
Protonet+, ours		not applic.	not applic.	not applic.	not applic.	not applic.	not applic.

Table 1: 1-shot: Test accuracy on the test set after fine-tuning on the validation set and training on the base set for 100 epochs, and the best and the second best results for each domain shift are highlighted.

		CIFARFS No domain shift	plant -> fruit	Land animals -> fish	Land animals ->insects	Land animals -> aquatic mammals	Outdoor scenes and object ->vehicles
Baseline	validation acc.	-	-	-	-	-	-
	few-shot acc.	65.0% (0.8%)	64.9% (0.7%)	42.2% (0.6%)	46.4% (0.7%)	49.4% (0.6%)	42.0% (0.6%)
Baseline++	validation acc.	-	-	-	-	-	-
	few-shot acc.	74.8% (0.8%)	71.8% (0.8%)	40.5% (0.6%)	38.8% (0.6%)	37.8% (0.6%)	37.8% (0.6%)
relationnet	validation acc.	65.2% (1.7%)	67.9% (1.8%)	47.9% (1.6%)	45.6% (1.9%)	60.5% (1.1%)	62.4% (2.2%)
	few-shot acc.	78.7% (0.7%)	75.3% (0.8%)	37.9% (0.7%)	38.4% (0.7%)	39.5% (0.7%)	36.6% (0.6%)
	few-shot (adapt) acc.	not applic.	not applic.	not applic.	not applic.	not applic.	not applic.
Protonet	validation acc.	70.4% (1.5%)	73.8% (1.4%)	45.3% (1.6%)	44.6% (1.6%)	62.3% (1.5%)	62.0% (1.9%)
	few-shot acc.	75.0% (0.7%)	74.3% (0.8%)	41.9% (0.7%)	45.7% (0.8%)	44.5% (0.7%)	41.5% (0.7%)
	few-shot (adapt) acc.	73.3% (0.7%)	72.4% (0.7%)	40.4% (0.6%)	43.4% (0.7%)	44.7% (0.7%)	40.6% (0.6%)
Protonet+, ours	validation acc.	69.9% (1.7%)	77.3% (1.4%)	45.6% (1.6%)	46.5% (1.7%)	61.7% (1.2%)	63.1% (2.0%)
	few-shot acc.	73.6% (0.7%)	44.5% (0.7%)	42.3% (0.7%)	46.6% (0.7%)	46.5% (0.7%)	45.1% (0.8%)

Table 2: 2-shot: Test accuracy on the test set after fine-tuning on the validation set and training on the base set for 100 epochs, and the best and the second best results for each domain shift are highlighted.

		CIFARFS No domain shift	plant -> fruit	Land animals -> fish	Land animals ->insects	Land animals -> aquatic mammals	Outdoor scenes and object ->vehicles
Baseline	validation acc.	-	-	-	-	-	-
	few-shot acc.	77.1% (0.6%)	76.0% (0.6%)	49.5% (0.7%)	51.7% (0.7%)	54.8% (0.6%)	50.4% (0.7%)
Baseline++	validation acc.	-	-	-	-	-	-
	few-shot acc.	78.8% (0.5%)	76.4% (0.5%)	41.9% (0.6%)	43.4% (0.6%)	43.6% (0.7%)	39.3% (0.6%)
Relationnet	validation acc.	74.8% (1.4%)	73.1% (1.5%)	49.4% (1.8%)	50.9% (1.4%)	63.6% (1.3%)	68.0% (0.6%)
	few-shot acc.	81.3% (0.5%)	78.0% (0.5%)	40.2% (0.7%)	44.8% (0.7%)	41.2% (0.6%)	41.4% (0.7%)
	few-shot (adapt) acc.	81.8% (0.5%)	78.4% (0.5%)	41.0% (0.7%)	45.3% (0.7%)	42.3% (0.6%)	40.8% (0.7%)
Protonet	validation acc.	77.4% (1.1%)	78.1% (1.0%)	49.8% (1.5%)	50.0% (1.6%)	66.8% (1.5%)	68.9% (1.6%)
	few-shot acc.	82.6% (0.5%)	82.3% (0.5%)	47.6% (0.7%)	51.0% (0.7%)	55.9% (0.6%)	47.0% (0.7%)
	few-shot (adapt) acc.	81.9% (0.5%)	81.5% (0.5%)	47.6% (0.7%)	50.9% (0.7%)	55.1% (0.6%)	47.4% (1.8%)
Protonet+, ours	validation acc.	79.4% (1.2%)	82.4% (1.2%)	50.1% (1.7%)	51.5% (1.5%)	67.9% (1.46%)	68.0% (1.6%)
	few-shot (adapt) acc.	79.0% (0.5%)	56.8% (0.7%)	48.3% (0.7%)	51.4% (0.6%)	51.7% (0.6%)	48.9% (0.7%)

Table 3: 5-shot: Test accuracy on the test set after fine-tuning on the validation set and training on the base set for 100 epochs, and the best and the second best results for each domain shift are highlighted.

REFERENCES

[1] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

[2] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

[3] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[4] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to Compare: Relation Network for Few-Shot Learning. *arXiv:1711.06025 [cs]*, March 2018. URL http://arxiv.org/abs/1711.06025. arXiv: 1711.06025.

[5] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical Networks for Few-shot Learning. *arXiv:1703.05175 [cs, stat]*, June 2017. URL http://arxiv.org/abs/1703.05175. arXiv: 1703.05175.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.