Final-Term Course Report for Statistical Learning and Inference

Shuodian Yu (120033910026), Yifei Shen (120033910111), and Yiming Liu (120033910143)

Abstract. Transductive inference is an effective method to solve the problem of insufficient data in few-shot learning settings. A popular transductive inference technique for methods based on several metrics is to use the mean of the most reliable query examples or the confidence-weighted average of all query samples to update the prototype of each class. However, it should be noted that the model confidence may be unreliable, which may lead to incorrect predictions [6]. Based on the work in [6], instead of taking each support examples in the support set with uniform weight, we consider the different modals of the prototype for each class with the variable queries. For each class, we get a prototype with attention mechanism under different query examples. Moreover, we analysis the parameter sensitivity and discuss some disadvantages in [6], such as the distance function, the scale of original prototype in the calculation of the updated prototype, etc.

1 Introduction

While Deep Learning achieves great performance in various tasks like classification and regression, it relies heavily on datasets and computation resources. When the annotated data is not enough, the performance of a state-of-the-art model may degrade significantly. In many fields, A lack of data is a reality. For example, images of rare wild animals are not as much as normal pet images. Moreover, companies hesitate to spend money on building brand new datasets of emerging categories due to time or other costs.

Few shot learning, namely learning new concepts with very few labeled examples just like human, is of great importance. Contemporary approaches to few-shot learning often decompose training into an auxiliary meta learning phase where transferable knowledge is learned in the form of good initial conditions [3], embeddings [13, 17] or optimization strategies [10].

However, previous transductive or semi-supervised inference approaches are fundamentally limited by the intrinsic unreliability of the labels predicted on the unseen samples. Consequently, in [6], the authors aim to tackle this problem by proposing a novel confidence-based transductive inference scheme for metric-based meta-learning models. Specifically, they first propose to meta-learn the distance metric to assign different confidence scores to each query instance for each class, such that the updated prototypes obtained by confidence-weighted averaging of the queries improve classification of the query samples.

In this report, we introduce a model with confidence score to improve the accuracy and generalization of the model, as is shown in Figure 1. Instead of taking each support examples in the support set with uniform weight, we consider the different modals of the prototype for each class with the variable queries. For each class, we get a prototype with attention mechanism under different query examples. The details will be discussed in Section 4.

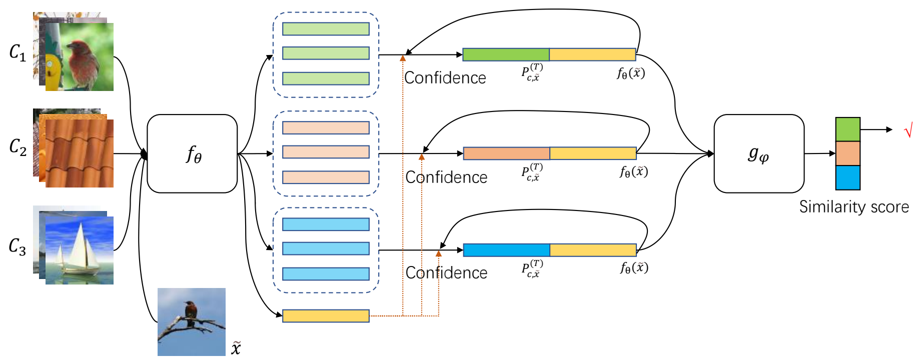

Fig. 1. The overall architecture of the proposed model. The meanings of the variables in the figure is illustrated in the Section 4.

2 Related Work

2.1 Few shot classification and Meta learning

The study of one or few-shot object classification does not rise in recent years. Earlier works try to involve generative models with complex iterative inference strategies [2]. In the sense that a model should extract some transferable knowledge from a set of auxiliary tasks, more and more models achieve success by using meta-learning strategy.

Meta learning is also called learning to learn. It aims to realize a mechanism to learn new concepts and skills fast with a few or zero training examples, just like human can recognize a new thing they never saw before. The task of few shot classification is a case under which meta learning aims to solve. There are three main classes of meta-learning approaches for addressing the few-shot classification problem [16].

Optimization-based or gradient descent based schemes like MAML [3] aim to meta-learn an initial condition that is good for fine-tuning on few-shot problems. The strategy is to pre-train the model's weight into a configuration sensitive to a small change by the fine tuning of later few shot samples. A small change of the initial weights result in a big improvement for the final classification.

RNN memory based approaches [17] leverage Recurrent Neural Networks with memories. It iterates over examples of given problem and accumulates the knowledge required to solve that problem in its hidden activations, or external memory. New examples can be classified by comparing them to historic information stored in the memory. So 'learning' a single target problem can occur in unrolling the RNN, while learning-to-learn means training the weights of the RNN by learning many distinct problems.

However, these approaches either suffer from the need to fine-tune on the target problem, or the complexity of recurrent networks and the issues involved in ensuring the adequacy of memory [15]. In contrast, our approach is mainly inspired by Metric Learning Approaches, which solves target problems in an entirely feed-forward manner with no model updates required.

2.2 Embedding and Metric Learning Approaches

Metric-based few-shot classification methods make the prediction based on the similarity between the query image and support examples. While the prior approaches entail some complexity when learning the target few-shot problem, metric-based methods have attracted considerable attention due to their simplicity and effectiveness.

In general, Metric-based few-shot classification approaches learn a similarity space in which learning is efficient for few-shot examples. These methods consists of a feature encoder to extract features from both the labeled and unlabeled images and a metric function that takes image features as input and predict the category of unlabeled images.

The learned embedding function works like a projection function that take every image into a feature similarity space. The metric function then works as a distance function in this space. By comparing the distance in this learned similarity space, we can categorize the test samples of few shot classes. In this space, images are easy to recognise using simple nearest neighbor or linear classifiers [13, 17]. In this case, the meta-learned transferable knowledge are the projection functions and the target problem is a simple feed-forward computation.

Prototypical Networks [13] firstly build a prototype representation of each class in the embedding space. As an extension of Prototypical Networks, IMF [1] constructs infinite mixture prototypes by self-adaptation. RelationNet [15] adopts a distance metric network to learn pointwise relations in both support and query samples.

2.3 Transductive learning

Since few-shot classification is intrinsically challenging, we may assume that we can access other unlabeled query examples, which is called transductive learning [18]. Here we name a few recent works. TPN [9] constructs a nearest-neighbor graph and propagate labels to pseudo-label the unlabeled query examples. EGNN [5] similarly constructs a nearest-neighbor graph, but utilizes both edge and node features in the update steps. On the other hand, Hou et al. [4] tries to update class prototypes by picking top-k confident queries with their own criteria. Our approach also updates class prototypes for each transduction step, but makes use of all the query examples instead of a small subset of k examples.

2.4 Semi-supervised learning

In the few-shot classification, semi-supervised learning can access additional large amount of unlabeled data. Ren et al. [11] proposed several variants of soft k-means method in prototypical networks [13], where soft label is predicted confidence of unlabeled sample. Li et al. [8] proposed the self-training method with pseudo labeling module based on gradient descent approaches [3][14]. Basically, if an unlabeled query set is used for few-shot classification instead of an additional unlabeled set, it becomes transductive learning, and vice versa. Our approach has connection to soft k-means method of Ren et al.[11], but we predict the confidence with input-adaptive distance metric and use meta-learned confidence under various perturbations.

3 MCT Method

3.1 Preliminaries

In the conventional C-way N-shot classification, we first sample C classes randomly from the entire set of classes, and then sample N and M examples from each class for the support set and query set, respectively. We define this sampling distribution as $p(\tau)$. As a result, we have a support set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{C\times N}$ and query set $\mathcal{Q} = \{(\tilde{\mathbf{x}}_i, \tilde{y}_i)\}_{i=1}^{C\times M}$, where $y, \tilde{y} \in \{1, \ldots, C\}$ are the class labels.

3.2 Transductive Inference with Soft k-means

MCT [6] is the transductive inference method using the confidence scores of query examples computed by soft k -means algorithm. Suppose that we are given an episode consisting of support set $\mathcal{S}$ and query set $\mathcal{Q}$. We also define $\mathcal{S}_c$ as the set of support examples in class c and $\mathcal{Q}_x = \{\tilde{\mathrm{x}}_1, \ldots, \tilde{\mathrm{x}}_{C\times M}\}$ as the set of all query instances. Starting from prototypical networks, we first compute the initial prototype $P_c^{(0)} = \frac{1}{|\mathcal{S}_c|}\sum_{x\in\mathcal{S}_c} f_\theta(\mathbf{x})$ for each class $c = 1, \ldots, C$. Then, for each step $t = 1, \ldots, T$, and for each query example $\tilde{\mathrm{x}} \in Q_x$, we compute its confidence score, which denote the probability of it belonging to each class c, as follows:

$$q_c^{(t-1)}(\tilde{\mathrm{x}}) = \frac{\exp\left(-d\left(f_\theta(\tilde{\mathrm{x}}), P_c^{(t-1)}\right)\right)}{\sum_{c'=1}^{C}\exp\left(-d\left(f_\theta(\tilde{\mathrm{x}}), P_{c'}^{(t-1)}\right)\right)} \tag{1}$$

where $d(\cdot, \cdot)$ is Euclidean distance and $P^{(t-1)}$ denotes $t-1$ steps updated prototype. We then update the prototypes of class c based on the confidence scores (or soft labels) $q_c^{(t-1)}(\tilde{\mathrm{x}})$ for all $\tilde{\mathrm{x}} \in \mathcal{Q}_x$:

$$P_c^{(t)} = \frac{\sum_{\mathbf{x}\in\mathcal{S}_c} 1 \cdot f_\theta(\mathbf{x}) + \sum_{\tilde{\mathbf{x}}\in\mathcal{Q}_x} q_c^{(t-1)}(\tilde{\mathbf{x}}) \cdot f_\theta(\tilde{\mathbf{x}})}{\sum_{\mathbf{x}\in\mathcal{S}_c} 1 + \sum_{\tilde{\mathbf{x}}\in\mathcal{Q}_x} q_c^{(t-1)}(\tilde{\mathbf{x}})} \tag{2}$$

which is the weighted average that we previously mentioned. Note that the confidence of the support examples is always 1, since their class labels are observed. We repeat the process until $t = 1, \ldots, T$.

However, for Eq. (2), we argue that it is not reasonable, because the scale between the original prototype in the calculation of the updated prototype is limited by the number of shots in each class. Here, we modified the equation as:

$$P_c^{(t)} = \frac{\sum_{\mathbf{x}\in\mathcal{S}_c} \alpha \cdot f_\theta(\mathbf{x}) + \sum_{\tilde{\mathbf{x}}\in\mathcal{Q}_x} q_c^{(t-1)}(\tilde{\mathbf{x}}) \cdot f_\theta(\tilde{\mathbf{x}})}{\sum_{\mathbf{x}\in\mathcal{S}_c} \alpha + \sum_{\tilde{\mathbf{x}}\in\mathcal{Q}_x} q_c^{(t-1)}(\tilde{\mathbf{x}})} \tag{3}$$

We will further do some experiments to test the sensitivity of the model with regard to α.

3.3 Meta-Confidence Transduction

Meta-learning confidence with input-adaptive distance metric Meta-learning confidence with input-adaptive distance metric We first propose to meta-learn the input-adaptive metric by performing transductive inference during training with query instances, to obtain a metric that yield performance improvements when performing transductive inference using it. Specifically, we meta-learn the distance metric d_ϕ in Eq. (3), which we define as Euclidean distance with normalization and instance-wise metric scaling g_ϕ^I, or pair-wise metric scaling g_ϕ^P :

$$d_\phi^I(\mathbf{a}_1, \mathbf{a}_2) = \left\| \frac{\mathbf{a}_1 / \|\mathbf{a}_1\|_2}{g_\phi^I(\mathbf{a}_1)} - \frac{\mathbf{a}_2 / \|\mathbf{a}_2\|_2}{g_\phi^I(\mathbf{a}_2)} \right\|^2, \quad d_\phi^P(\mathbf{a}_1, \mathbf{a}_2) = \left\| \frac{\mathbf{a}_1 / \|\mathbf{a}_1\|_2}{g_\phi^P(\mathbf{a}_1, \mathbf{a}_2)} - \frac{\mathbf{a}_2 / \|\mathbf{a}_2\|_2}{g_\phi^P(\mathbf{a}_1, \mathbf{a}_2)} \right\|_2^2 \tag{4}$$

for all $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{R}^l$. Note that the normalization allows the confidence to be mainly determined by metric scaling. In order to obtain the optimal scaling function $g_\phi \in \left\{g_\phi^I, g_\phi^P\right\}$ for transduction, we first compute the query likelihoods after T transduction steps, and then optimize ϕ, the parameter of the scaling function g_ϕ by minimizing the following instance-wise loss for $d_\phi \in \left\{d_\phi^I, d_\phi^P\right\}$:

$$
\begin{aligned}
L_I^\tau(\theta,\phi) &= \frac{1}{|\mathcal{Q}|}\sum_{(\tilde{\mathbf{x}},\tilde{y})\in\mathcal{Q}} -\log p(\tilde{y}\mid\tilde{\mathbf{x}},\mathcal{S};\theta,\phi)\\
&= \frac{1}{|\mathcal{Q}|}\sum_{(\tilde{\mathbf{x}},\tilde{y})\in\mathcal{Q}} \left\{ d_\phi\left(f_\theta(\tilde{\mathbf{x}}),P_c^{(T)}\right) + \sum_{c'=1}^{C}\exp\left(-d_\phi\left(f_\theta(\tilde{\mathbf{x}}),P_{c'}^{(T)}\right)\right)\right\}
\end{aligned}
$$

As for g_ϕ, we simply use a CNN with fully-connected layers which takes either the feature map of an instance or the concatenated feature map of a pair of instances as an input. We set the number of transduction steps to $T=1$ for training to minimize the computational cost, but use $T=10$ for test.

4 Proposed Method

4.1 Transductive Inference

MCT proposed the transductive inference with soft k-means that use the confidence scores of query examples to update the initial prototype $P_c^{(0)}$ for each class iteratively. Inspired by [1], instead of taking each support examples in the support set $\mathcal{S}_c$ with uniform weight, we consider the different modals of the prototype for each class with the variable queries.

First, for each query example $\tilde{\mathbf{x}}\in Q_x$, we compute the similarity score with all the support instances $\mathbf{x}\in\mathcal{S}_c$:

$$
att(\tilde{\mathbf{x}},\mathbf{x}) = \frac{\exp(f_{att}(Wf_\theta(\mathbf{x}),\tilde{W}f_\theta(\tilde{\mathbf{x}})))}{\sum_{\mathbf{x}'\in\mathcal{S}_c}\exp(f_{att}(Wf_\theta(\mathbf{x}'),\tilde{W}f_\theta(\tilde{\mathbf{x}})))}
$$

where $f_\theta(x)\in\mathbb{R}^l$ is the learnt embedding function that maps an input $\mathbf{x}$ to a latent embedding z in an l-dimensional metric space, which is usually a convolutional network. $W,\tilde{W}\in\mathbb{R}^{d\times l}$ are two weight matrix to transform the learned feature embedding into support and query space respectively, and $f_{att}(\cdot,\cdot)$ is trained to learn the similarity score of two embeddings.

Then the initial prototype $P_{c,\tilde{\mathbf{x}}}^{(0)}$ for each class $c=1,\dots,C$ and query $\tilde{\mathbf{x}}\in Q_x$ can be computed like prototypical networks [13] but with the attention weights:

$$
P_{c,\tilde{\mathbf{x}}}^{(0)} = \sum_{\mathbf{x}\in\mathcal{S}_c} att(\tilde{\mathbf{x}},\mathbf{x})\cdot f_\theta(\mathbf{x}).
$$

For each step $t=1,\dots,T$, we compute the confidence score for each query example $\tilde{\mathbf{x}}\in Q_x$, which measures the the possibility that it is belong to class c as follows:

$$
q_{c,\tilde{\mathbf{x}}}^{(t-1)}(\tilde{\mathbf{x}}') = \frac{\exp\left(-f_d\left(f_\theta(\tilde{\mathbf{x}}'),P_{c,\tilde{\mathbf{x}}}^{(t-1)}\right)\right)}{\sum_{c'=1}^{C}\exp\left(-f_d\left(f_\theta(\tilde{\mathbf{x}}'),P_{c',\tilde{\mathbf{x}}}^{(t-1)}\right)\right)},
$$

where $f_d(\cdot,\cdot)$ is the distance function and $P_{c,\tilde{\mathbf{x}}}^{(t-1)}$ is the $t-1$ steps updated prototype. At the end of each iteration, we update the prototype based on the calculated confidence score:

$$P_{c,\tilde{\mathbf{x}}}^{(t)} = \frac{\alpha \sum_{\mathbf{x}\in\mathcal{S}_c} att(\tilde{\mathbf{x}},\mathbf{x}) \cdot f_\theta(\mathbf{x}) + \sum_{\tilde{\mathbf{x}}'\in\mathcal{Q}_x} q_{c,\tilde{\mathbf{x}}}^{(t-1)}(\tilde{\mathbf{x}}') \cdot f_\theta(\tilde{\mathbf{x}}')}{\alpha + \sum_{\tilde{\mathbf{x}}'\in\mathcal{Q}_x} q_{c,\tilde{\mathbf{x}}}^{(t-1)}(\tilde{\mathbf{x}}')},$$

where α is a hyper-parameter that balance the weight between original prototype and confidence score. We repeat the process until $t = 1,\ldots,T$.

4.2 Distance Metric Learning

To measure the transductive inference during training with query instances, we have to obtain a distance metric $f_d(\cdot,\cdot)$ with meta-learning. Refer to relation network [15], we learn the distance score with the embedding module:

$$f_d(\mathbf{x}_i,\mathbf{x}_j) = g_\phi\left(\mathcal{C}\left(f_\varphi\left(x_i\right), f_\varphi\left(x_j\right)\right)\right),$$

where $\mathcal{C}$ is the concatenate operation and g_ϕ, f_φ are two DNN modules to learn the distance and embedding-transfer function respectively.

We optimize all the above parameters $w = \{\phi,\varphi,\theta,\ldots\}$ by minimizing the following instance-wise loss with regularization:

$$L_I^\tau = \frac{1}{|\mathcal{Q}|} \sum_{(\tilde{\mathbf{x}},\tilde{y})\in\mathcal{Q}} -\log p(\tilde{y} \mid \tilde{\mathbf{x}},\mathcal{S}) + \lambda\|w\|_2^2$$

4.3 Model's drawbacks

Since our proposed model has a large number of parameters, the optimization of the model is extremely hard. We encountered a lot of obstacles when we were training the model.

First, the complexity of attention module is to high for the feature maps of the images. Take ResNet-12 as backbone example, the size of output feature for an image in the miniImageNet dataset is $(512, 6, 6)$. The corresponding attention weight matrix $W \in \mathbb{R}^2$ will be too large for storage and calculation. Therefore, we employ a flatten and linear operation for the output features for all the images. However, unlike CNN layer, linear transform is not suitable for image features to some extent.

Second, since we have generated a corresponding prototype for each query, it also introduces abundant parameters that are hard to optimize. Through the experiment we have observed that the value of tensor which stores the hundred number of prototype images will become uniform as the number of training epochs increases.

Compared to the meta-learnt distance function, above drawbacks of our attention model to evaluate the similarity between queries and prototypes and develop unique prototypes for each query image will be hard to optimize and lead to model collapse. We are still looking forward to developing a method to learn how to measure the relationship between a group of images with decreased parameters.

5 Experiments

5.1 Experimental settings

Datasets Here we intend to use two datasets: Omniglot, miniImageNet and tieredImageNet. The Omniglot dataset is designed for developing more human-like learning algorithms. It contains 1623 different handwritten characters from 50 different alphabets. Each of the 1623 characters was drawn online via Amazon's Mechanical Turk by 20 different people [7]. The miniImageNet dataset contains 100 classes randomly chosen from ILSVRC-2012 [12] and 600 images of size 84×84 pixels per class. It is split into 64 base classes, 16 validation classes and 20 novel classes. Note that we will choose those two datasets at least. If time is sufficient, we will also do experiments on other datasets to evaluate our model.

Baselines We will use various state-of-the-art baselines for few-shot classification, including MAML [3], Matching Networks [17],Learning to Compare [15].

Evaluation Metrics The major evaluation metric is accuracy. Here, we will use different experiments for evaluating our model:

- Implement our proposed model and compare it with other baselines on accuracy.
- In the support set, choose different N (number of categories) and K (number of labeled samples for each category) in the N way K shot few-shot classification problem to see the model performance.
- The MCT described in [6] leverages a confidence score in the query set which is similar to our ideas. However, they only use a mean pooling operation in the support set. We reckon that attention model used in the support set is more important, because different samples contribute their properties to their categories in various degree. Therefore, taking the proportion of different samples in a category may better improve the performance and generalization of the model. We should compare our model with MCT to validate our assumption.

5.2 Main results

We evaluate the performance of MCT model and test the different scalar weight α of initial prototype and previous one during the iteration. Due to the huge amount of parameters, the training of proposed attention model for distance learning is failed and we will not show the performance. All the codes are attached and we are still working to optimize the attention model.

The results show that the value of α is not so significant to the final performance of the MCT model, since the iterate round is very small in the training stage.

Model	miniImageNet	
	1-shot	5-shot
MAML [3]	48.70%	63.11%
Matching [17]	43.56%	55.31%
LC [15]	57.02%	71.07%
MTL [14]	61.20%	75.53%
MCT-pair	**64.49**%	81.63%
$\alpha = 0$	63.47%	81.54%
$\alpha = 0.2$	63.07%	**81.64**%
$\alpha = 0.5$	61.95%	80.74%
$\alpha = 5$	63.36%	80.94%
Self-adaptive α	63.15%	80.86%
No scalar	62.69%	72.04%

Table 1. Average classification performance over 1000 randomly generated episodes. We consider 5-way classification on all the datasets. Some results are reported from [6].

6 Conclusion

In this report, we discuss some confidence measurements between images. The results of meta-learnt confidence transduction highly improve the performance of prototype method for few shot classification with a meta-learnt distance metric. Although we also discuss a possible way that using the attention weight between query image and images in the support set and learning the prototypes for each query, the huge amount of parameters lead to model collapse and train failure. We are looking forward for a more reasonable solution to the problems.

References

1. Allen, K.R., Shelhamer, E., Shin, H., Tenenbaum, J.B.: Infinite mixture prototypes for few-shot learning. arXiv preprint arXiv:1902.04552 (2019)
2. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. PAMI **28**(4), 594–611 (2006)
3. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML (2017)
4. Hou, R., Chang, H., Ma, B., Shan, S., Chen, X.: Cross attention network for few-shot classification. Advances in Neural Information Processing Systems **32**, 4003–4014 (2019)
5. Kim, J., Kim, T., Kim, S., Yoo, C.D.: Edge-labeling graph neural network for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11–20 (2019)
6. Kye, S.M., Lee, H.B., Kim, H., Hwang, S.J.: Transductive few-shot learning with meta-learned confidence. arXiv preprint arXiv:2002.12017 (2020)
7. Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B.: Human-level concept learning through probabilistic program induction. Science **350**(6266), 1332–1338 (2015)

8. Li, X., Sun, Q., Liu, Y., Zhou, Q., Zheng, S., Chua, T.S., Schiele, B.: Learning to self-train for semi-supervised few-shot classification. In: Advances in Neural Information Processing Systems. pp. 10276–10286 (2019)
9. Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S.J., Yang, Y.: Learning to propagate labels: Transductive propagation network for few-shot learning. arXiv preprint arXiv:1805.10002 (2018)
10. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning (2016)
11. Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S.: Meta-learning for semi-supervised few-shot classification. arXiv preprint arXiv:1803.00676 (2018)
12. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
13. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: NIPS. pp. 4077–4087 (2017)
14. Sun, Q., Liu, Y., Chua, T.S., Schiele, B.: Meta-transfer learning for few-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 403–412 (2019)
15. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: CVPR. pp. 1199–1208 (2018)
16. Tseng, H.Y., Lee, H.Y., Huang, J.B., Yang, M.H.: Cross-domain few-shot classification via learned feature-wise transformation. In: ICLR (2020)
17. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: NIPS. pp. 3630–3638 (2016)
18. Vladimir, N.V., Vapnik, V.: Statistical learning theory. Xu JH and Zhang XG. translation. Beijing: Publishing House of Electronics Industry, 2O04 (1998)