TAEML: Task-Adaptive Ensemble of Meta-Learners

Minseop Park[*] AITRICS
Saehoon Kim AITRICS
Jungtaek Kim POSTECH
Yanbin Liu UTS
Seungjin Choi POSTECH

Abstract

Most of meta-learning methods assume that a set of tasks in the meta-training phase is sampled from a single dataset. Thus, when a new task is drawn from another dataset, the performance of meta-learning methods is degraded. To alleviate this effect, we introduce a task-adaptive ensemble network that aggregates meta-learners by putting more weights on the learners that are expected to perform well to the given task. Experiments demonstrate that our task-adaptive ensemble significantly outperforms previous meta-learners and their uniform averaging.

1 Introduction

The primary interest of this paper is *few-shot classification*: the objective is to learn a function that classifies each instance in a query set Q into N classes in a support set S, where each class has K trainable examples. The episodic training strategy [14, 12] generalizes to a novel task by learning a set of tasks $\mathcal{E} = \{E_i\}_{i=1}^{T}$, where E_i is the ith episode (a tuple of Q and S) and T is the number of training elements. A common practice to train/validate a meta-learner has a major limitation, where tasks in the phase of meta-training/-test are sampled from the same dataset. Similar to supervised learning in which training and test distributions are typically matched, meta-learning implicitly assumes that tasks from meta-training/-test share the similar high-level concepts. Then, the learner performs poorly to a novel task that does not share common attributes of the tasks in meta-training.

Rather than learning with a single dataset, we expect that meta-learners trained from multiple datasets perform robustly to a novel task, because it is possible that the novel task and some of the tasks in meta-training could share attributes. Yet, a naïve training from diverse sets does not perform well because the model considers many irrelevant tasks. To alleviate the effects, an obvious approach is to retrieve similar ones of a novel task and to train a meta-learner from them. This may perform well enough to the target task, but we observe some limitations: (i) learning an appropriate metric between datasets is quite challenging and (ii) a meta-learner is always trained from scratch whenever a new task is given. Instead of selecting similar datasets, it is better to keep multiple meta-learners trained from datasets and determine how to aggregate them effectively. This encourages us to build an ensemble of meta-learners where weights are adaptively determined when a novel task is given.

The major concern of building such model is that the model has to determine the weights while it glimpses the target task, contrast to the regular supervised learning which has abundant validating examples to evaluate the base learners. To solve this, we train the model to learn how to ensemble given an episode. More specifically, our ensemble meta-learner is established by putting more weight on the base-learner that is expected to perform well to the tasks in the meta-test phase. To evaluate the model given a small number of instances, we employ the embedding structure of an episode which can be an indicator of model performance. Then, a task-adaptive ensemble network is introduced such that more attention is given to meta-learners based on their embeddings. Since our model determines how to ensemble by observing the behavior of task-given meta-learners, it can be viewed as a similar idea of meta-recognition system [11] that analyzes and predicts the recognition system based on their outputs. Hence, learning to evaluate meta-learners can be considered as two layers of meta-learning.

[*] mike_seop@aitrics.com

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

Our contribution is two-fold: (i) we point out a major limitation of the conventional approaches and propose TAEML as a solution, such that the ensemble weights on base meta-learners are task-adaptively determined (ii) we observe that our ensemble network achieves the best performance among the state-of-the-art algorithms, including a uniform averaging of multiple meta-learners.

2 Proposed Method

2.1 Task-Adaptive Ensemble of Meta-Learners (TAEML)

We introduce the network that takes embedding of an episode produced by a base meta-learner to determine the ensemble weight, denoted as $f_e : \{\boldsymbol{\zeta}_m\}_{m=1}^{M} \to [0, \lambda]^M$, where M is the number of learners, λ is a scale parameter, and $\boldsymbol{\zeta}_m$ means representation of (Q, S) with the mth base-learner. We choose the prototypical network [12] as a base meta-learner, because a task is well-represented by residuals between a set of prototypes and queries:

$$[\boldsymbol{\zeta}_m]_{i,j,:} = (\boldsymbol{\eta}_{q_i} - \boldsymbol{\eta}_{p_j}) \odot (\boldsymbol{\eta}_{q_i} - \boldsymbol{\eta}_{p_j}), \tag{1}$$

where $\odot$ denotes an element-wise product, $\boldsymbol{\zeta}_m \in \mathbb{R}^{N_Q \times N \times d_1}$ means the representation of an episode from the mth base meta-learner with N_Q queries. $\boldsymbol{\eta}_{q_i}$ and $\boldsymbol{\eta}_{p_j}$ denote d_1-dimensional vectors of the ith query and jth prototype, respectively. We remark that our representation does not depend on K and is easily obtained from the last layer of the mth prototypical network. Note that our model is invariant to the type of base meta-learner and can be expanded to gradient-based meta-learning models [3, 9] or more well-performing meta-learners [13].

We now introduce our ensemble prediction denoted as $p_e(E) \in [0, 1]^{N_Q \times N}$, which is a stack of predicted class probabilities of each query. It is formally defined as below:

$$[p_e(E)]_q = \text{softmax}\left(\sum_{m=1}^{M} f_e(\boldsymbol{\zeta}_m; \boldsymbol{\theta}) p_m(\boldsymbol{x}_q | S)\right), \tag{2}$$

where $[p_e(E)]_q$ refers an ensemble prediction of the qth query that is a weighted combination of $p_m(q, S)$, which is the predicted class probability of mth base learner. The weight on the mth base learner is determined by $f_e(\boldsymbol{\zeta}_m; \boldsymbol{\theta}) \in [0, \lambda]$ that denotes our ensemble network parameterized by θ.

The objective function for training TAEML is then introduced as follows:

$$\arg\max_{\theta} \mathbb{E}_{R \sim D} \mathbb{E}_{E \sim R} \left[\frac{1}{N_Q} \sum_{q,n} y_{q,n} \log p_e(E)_{q,n}\right], \tag{3}$$

where D refers the distribution of datasets, R means a single dataset, and E represents an episode. And, $y_{q,n}$ denotes a one-hot vector of the nth dimension of qth query's label. Similarly, $p_e(E)_{q,n}$ denotes the nth component of qth query's ensemble prediction. We employ the cross entropy loss over an episode between the true label and ensemble prediction.

We remark that the ensemble network in TAEML is trained in a transductive setting. In the perspective of regularization, it is better to find the ensemble weights that generally work well in the similar tasks. TAEML is then designed to take a set of queries as an input, instead of a single query. In other words, this generates episodic-wise coefficients for the base learners. While testing, the model makes a prediction either in transductive or non-transductive way by limiting the number of queries. A transductive meta-learner [10] is recently proposed by employing a neighborhood structure of queries. Yet, this does not take into account how to establish a task-adaptive ensemble network in a transductive setting.

2.2 Network Structure

TAEML is a simple multilayer perceptron that maps $\boldsymbol{\zeta}_m$ into a single scalar. For ease of implementation, our network is designed as a convolutional neural network. Figure 1 graphically shows how to transform the residuals $\{\boldsymbol{\zeta}_m\}_{m=1}^{M}$ into $[0, \lambda]^M$. Specifically, TAEML is composed of two convolutional layers, a global average pooling layer, and a dense layer. First, a residual $\boldsymbol{\zeta}_m$ is processed by $(1, N)$-size convolution filters along queries, with d_2 output channels. Then, 1-by-1 convolution with depth d_3 is employed to reduce the number of channels, which is followed by a pooling layer that averages the vectors of all queries. The final dense layer produces an ensemble weight for each base meta-learner.

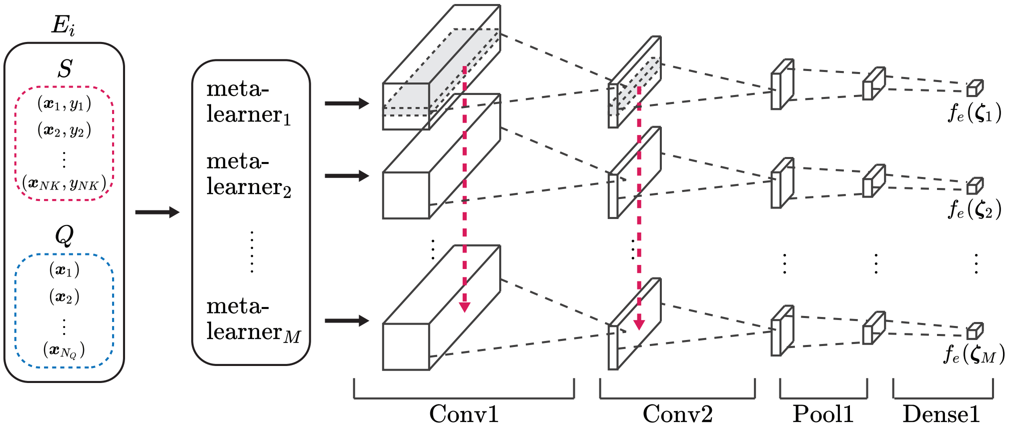
E_i
S
$(\boldsymbol{x}_1, y_1)$
$(\boldsymbol{x}_2, y_2)$
$(\boldsymbol{x}_{NK}, y_{NK})$
Q
$(\boldsymbol{x}_1)$
$(\boldsymbol{x}_2)$
$(\boldsymbol{x}_{N_Q})$
meta-learner$_1$
meta-learner$_2$
meta-learner$_M$
$f_e(\boldsymbol{\zeta}_1)$
$f_e(\boldsymbol{\zeta}_2)$
$f_e(\boldsymbol{\zeta}_M)$
Conv1
Conv2
Pool1
Dense1

Figure 1: A network structure of TAEML that produces the ensemble weights for the base meta-learners. The inputs of network are a query set Q and support set S. Each meta-learner generates the representation of an episode $\boldsymbol{\zeta}_m$, which are fed into a series of convolutional, pooling, and dense layers. The output of TAEML is a set of ensemble coefficients on each base meta-learner.

2.3 Training Strategy

We now describe our training procedure for TAEML, which is trained in an episodic way. We are given M training datasets, where classes are split into two subsets for training the base meta-learners and TAEML. Before training the TAEML, each base meta-learner is trained from a single dataset, and their parameters are fixed throughout the training procedure. After that, a single dataset (but exclusively split) is randomly drawn from M datasets, and an episode is sampled from it to optimize Eq. 3. We then obtain the input of TAEML, $\{\boldsymbol{\zeta}_i\}_{i=1}^{M}$, from M pretrained meta-learners, and update the parameters by minimizing the cross entropy between the ensemble prediction (Eq. 2) and the labels of the sampled queries in the episode, with learning rate β.

3 Experiment

3.1 Datasets and Model Configurations

We used five benchmark datasets (CIFAR100 [6], VOC2012 [1], AwA2 [16], Caltech256 [4], Omniglot [7]) for meta-training/validation, and used other datasets (CIFAR10 [6], miniImageNet [2], CUB200 [15], Caltech101 [4], MNIST [8]) for meta-test. In former datasets, 80% of classes in each dataset were randomly chosen to train base meta-learners, and rest 20% of classes to train TAEML. All of the images were resized to 84-by-84 and converted to three-channel images. We employed dataset-specific prototypical networks for base meta-learners of TAEML.

We compared TAEML into dataset-specific prototypical networks and meta-learners trained from multiple datasets. In addition, we compared a uniform averaging of dataset-specific prototypical networks to validate that our task-adaptive ensemble does not rely on a trivial averaging. Hyperparameters used in prototypical network were identical to the original implementation except that the output of the last layer were l_2 normalized to improve the performance. We set hyperparameters of TAEML as follows: $d_2 = 512$, $d_3 = 128$, $\lambda = 10^2$, fixed learning rate $\beta = 10^{-4}$, using Adam optimizer [5] with 15 queries for each class.

3.2 Results

Table 1 shows 10-way 5-shot classification accuracy of TAEML and several baselines. We validated two types of TAEML ($B_Q = 1$ and $B_Q = 30$): each of them sequentially evaluates B_Q queries from N_Q queries in the meta-test phase, where N_Q is equal to 150. We measured the averaged classification accuracy of 600 episodes randomly sampled from each test dataset.

Table 1: 10-way 5-shot classification results of TAEML and baselines. TAEML outperforms all baselines in case of CUB200, CIFAR10, Caltech101, and miniImagenet (abbreviated as mImgnet).

Model	Meta-			Meta-test		
	training	MNIST	CUB200	CIFAR10	Caltech101	mImgnet
Dataset-specific Prototypical Net	AwA2	0.686	0.374	0.261	0.545	0.330
	CIFAR100	0.781	0.280	0.461	0.597	0.337
	Omniglot	0.778	0.173	0.173	0.320	0.220
	VOC2012	0.688	0.263	0.246	0.491	0.302
	Caltech256	0.806	0.394	0.322	0.776	0.432
Prototypical Net	Multiple datasets	0.780	0.417	0.362	0.751	0.430
MAML		**0.821**	0.361	0.415	0.706	0.409
Uniform Averaging		0.797	0.402	0.381	0.730	0.438
TAEML ($B_Q = 1$)		0.813	0.415	0.454	0.770	0.447
TAEML ($B_Q = 30$)		0.812	**0.429**	**0.466**	**0.788**	**0.461**

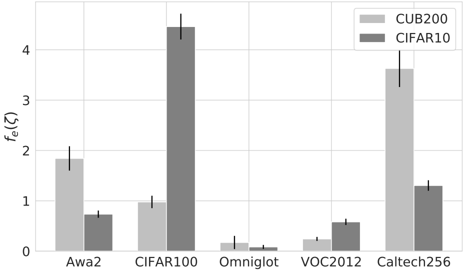

Figure 2: Ensemble coefficients (vertical axis) on base meta-learners trained from the associated datasets (horizontal axis) for 10-way 5-shot TAEML ($B_Q = 30$).

Table 2: Accuracy of TAEML ($B_Q = 15$) in 5-way classification task sampled from miniImagenet of which non-overlapping classes are included in the meta-training.

Model	1-shot	5-shot
ProtoNet-single	0.507	0.679
ProtoNet-multiple	0.478	0.663
Uniform Averaging	0.449	0.655
TAEML	**0.512**	**0.698**

Figure 2 shows the average ensemble weights on each base meta-learners of TAEML ($B_Q = 30$), where each meta-learner corresponds to the dataset used in meta-training. When there exists a relevant dataset to the target task, TAEML tended to assign relatively high attention to the dataset. We observed that the weight on CIFAR100 is much higher than the ones on other datasets, given the target task from CIFAR10. On the contrary, TAEML assigned weights more evenly for the tasks sampled from CUB200, compared to the ones from CIFAR10. The average weights are well aligned to their substantive performance on the target dataset, which cannot be inferred in advance since the validating examples are given as a single episode.

Table 2 shows 5-way few-shot miniImagenet classification to validate that our model improves the performance in the conventional setting by adding supplementary datasets. In this experiment, miniImagenet dataset is into meta-training, meta-validation, and meta-test as in [12], each of the split is used to train base-learners, train TAEML, and test respectively. And, the datasets used in 10-way 5-shot classification task are added in each split. We randomly shuffled validation/test set 5 times and reported average accuracy in each 600 episodes. While naïvely adding datasets degenerates the model performance (ProtoNet-multiple and Uniform Averaging), our model performs robustly by adjusting the coefficient for the base learners even if added datasets are not relevant to the target one.

4 Conclusion

In this paper, we proposed a task-adaptive ensemble of meta-learners, referred to as TAEML for few-shot classification. We observed that a common practice for meta-learning has a major limitation; tasks used in training and testing are sampled from the same dataset. To resolve this critical issue, we tackled a challenging problem in which a test task is sampled from a novel dataset. We then proposed an ensemble network that learns how to adaptively aggregate base meta-learners for the given task. Extensive experiments on diverse datasets confirm that TAEML outperforms other baselines.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-01779, A machine learning and statistical inference framework for explainable artificial intelligence).

References

[1] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 2010.

[2] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 2004.

[3] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

[4] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

[5] D. K. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[6] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[7] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 2015.

[8] Y. LeCun and C. Cortes. The MNIST database of handwritten digits, 1998. `http://yann.lecun.com/exdb/mnist/`.

[9] Y. Lee and S. Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[10] Y. Liu, J. Lee, M. Park, S. Kim, and Y. Yang. Transductive propagation network for few-shot learning. *arXiv preprint arXiv:1805.10002*, 2018.

[11] W. J. Scheirer, A. Rocha, R. J. Micheals, and T. E. Boult. Meta-recognition: The theory and practice of recognition score analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[12] J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[13] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[14] O. Vinyals, C. Blundell, T. P. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[15] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical report, California Institute of Technology, 2010.

[16] Y. Xian, C. Lampert, B. Schiele, and Z. Akata. Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.