Unsupervised Meta-Learning for Few-Shot Image Classification

Siavash Khodadadeh
Dept. of Computer Science
siavash.khodadadeh@knights.ucf.edu

Ladislau Bölöni
Dept. of Computer Science
lboloni@cs.ucf.edu

Mubarak Shah
Center for Research in Computer Vision
shah@crcv.ucf.edu

Motivation

Meta learning (learning to learn) algorithms facilitate few-shot learning

Model-agnostic meta learning (MAML) [1]

- Meta-learning phase: learn from tasks $T_1 \ldots T_n$
- Target learning phase: few shot learning on T_{n+1}

The good:

- Model-agnostic: no requirement for the network structure
- Very little labeled data needed at the target phase

The bad:

- A lot of labeled data needed at the meta-learning phase
- Tasks $T_1 \ldots T_n$ need to be drawn from the same distribution as the target - we need good knowledge of the target domain

Keep the general flow of MAML but use unlabeled data for

Algorithm 1: Unsupervised Meta-learning with Tasks constructed by Random sampling and Augmentation (UMTRA)

require : N: class-count, N_{MB}: meta-batch size, N_U: no. of updates
require : $\mathcal{U} = \{\ldots x_i \ldots\}$ unlabeled dataset
require : α, β: step size hyperparameters
require : $\mathcal{A}$: augmentation function
randomly initialize θ;
while *not done* **do**
  for *i in* $1 \ldots N_{MB}$ **do**
    Sample N data points $x_1 \ldots x_N$ from $\mathcal{U}$;
    $\mathcal{T}_i \leftarrow \{x_1, \ldots x_N\}$;
  end
  foreach $\mathcal{T}_i$ **do**
    Generate training set $\mathcal{D}_i = \{(x_1, 1), \ldots (x_N, N)\}$;
    $\theta'_i = \theta$;
    for *j in* $1 \ldots N_U$ **do**
      Evaluate $\nabla_{\theta'_i} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$;
      Compute adapted parameters with gradient descent: $\theta'_i = \theta'_i - \alpha \nabla_{\theta'_i} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$;
    end
    Generate validation set for the meta-update $\mathcal{D}'_i = \{(\mathcal{A}(x_1), 1), \ldots, (\mathcal{A}(x_N), N)\}$
  end
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$ using each $\mathcal{D}'_i$;
end

References:

[1] C. Finn, P. Abbeel, and S. Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." In Proc. of the 34th Int'l Conf. on Machine Learning (ICML-2017), pp. 1126-1135,, 2017.

Unsupervised Meta-learning with Tasks constructed by Random sampling and Augmentation (UMTRA)

Start with an unlabeled data set D of samples

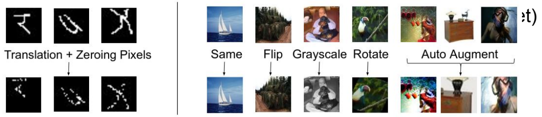
- We assume a large number of classes C present, but we don't know the class of each sample

Generate synthetic one shot training tasks $T_1 \ldots T_n$

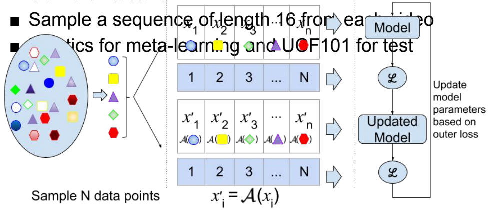
- Random selection of samples with artificial labels
- The large number of classes ensures high likelihood that classes are different

Create validation data by augmentation $x_v = A(x)$

- Augmentation type varies by domain, the objective is to maintain class membership of the augmented data



- On video: use frame selection to create augmented video
  - C3D architecture
  - Sample a sequence of length 16 from each video
  - Kinetics for meta-learning and UCF101 for test



Experiments

Few shot learning benchmarks: outperforms other unsupervised algorithms and alternates for best with CACTUs-MAML [2].

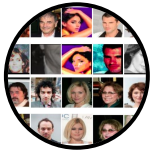
Massive reduction of required labeled data: 25000 to 25 for Omiglot (5,5) for 95.43% vs 98.83 vs MAML.

		Omniglot				Mini-Imagenet			
Algorithm (N, K)	**Clustering**	**(5,1)**	**(5,5)**	**(20,1)**	**(20,5)**	**(5,1)**	**(5,5)**	**(5,20)**	**(5,50)**
Training from scratch	N/A	52.50	74.78	24.91	47.62	27.59	38.48	51.53	59.63
k_{nn}-nearest neighbors	BiGAN	49.55	68.06	27.37	46.70	25.56	31.10	37.31	43.60
linear classifier	BiGAN	48.28	68.72	27.80	45.82	27.08	33.91	44.00	50.41
MLP with dropout	BiGAN	40.54	62.56	19.92	40.71	22.91	29.06	40.06	48.36
cluster matching	BiGAN	43.96	58.62	21.54	31.06	24.63	29.49	33.89	36.13
CACTUs-MAML	BiGAN	58.18	78.66	35.56	58.62	36.24	51.28	61.33	66.91
CACTUs-ProtoNets	BiGAN	54.74	71.69	33.40	50.62	36.62	50.16	59.56	63.27
k_{nn}-nearest neighbors	ACAI / DC	57.46	81.16	39.73	66.38	28.90	42.25	56.44	63.90
linear classifier	ACAI / DC	61.08	81.82	43.20	66.33	29.44	39.79	56.19	65.28
MLP with dropout	ACAI / DC	51.95	77.20	30.65	58.62	29.03	39.67	52.71	60.95
cluster matching	ACAI / DC	54.94	71.09	32.19	45.93	22.20	23.50	24.97	26.87
CACTUs-MAML	ACAI / DC	68.84	87.78	48.09	73.36	39.90	**53.97**	**63.84**	**69.64**
CACTUs-ProtoNets	ACAI / DC	68.12	83.58	47.75	66.27	39.18	53.36	61.54	63.55
UMTRA (ours)	N/A	**83.80**	**95.43**	**74.25**	**92.12**	**39.93**	50.73	61.11	67.15
MAML (Supervised)	N/A	94.46	98.83	84.60	96.29	46.81	62.13	71.03	75.54
ProtoNets (Supervised)	N/A	98.35	99.58	95.31	98.81	46.56	62.29	70.05	72.04

Works on video classification as well:
Test on **all samples** of each class
by looking at just **one example** for train

Algorithm	**Test Accuracy / F1-Score**
Training from scratch	29.30 / 20.48
Pre-trained on Kinetics	45.51 / 42.49
UMTRA on unlabeled Kinetics (ours)	**60.33 / 58.47**
Supervised MAML on Kinetics	71.08 / 69.44

...d for face recognition



Algorithm (N, K)	**(5, 1)**	**(5, 5)**	**(5, 10)**
Training from scratch	26.86	39.65	50.61
UMTRA	**33.43**	**50.19**	**58.84**

This research is based upon work supported in parts by the National Science Foundation under Grant numbers IIS-1409823 and IIS-1741431 and Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA R&D Contract No. D17PC00345.