Continuous Meta-Learning Without Tasks

Chen Zhao

https://arxiv.org/pdf/1912.08866.pdf

1 Introduction

Meta-learning is a promising strategy for learning to efficiently learn within newtasks, using data gathered from a distribution of tasks. However, the meta-learningliterature thus far has focused on thetask segmentedsetting, where at train-time,offline data is assumed to be split according to the underlying task, and at test-time,the algorithms are optimized to learn in a single task. In this work, we enable theapplication of generic meta-learning algorithms to settings where this task seg-mentation is unavailable, such as continual online learning with a time-varyingtask. We present meta-learning via online changepoint analysis (MOCA), an ap-proach which augments a meta-learning algorithm with a differentiable Bayesianchangepoint detection scheme. The framework allows both training and testingdirectly on time series data without segmenting it into discrete tasks. We demon-strate the utility of this approach on a nonlinear meta-regression benchmark aswell as two meta-image-classification benchmarks

2 Motivations

- There are many applications where task segmentation is unavailable, which have thus far been under-addressed in the meta-learning literature. For example, consider a robot which must learn to adapt to a changing environment. The robot may switch from one environment to another during the course of deployment, and these task switches may not be directly observed.
- Using an existing time series from interaction to craft a meta-dataset may require a difficult or expensive process of detecting switches in task.
- In this work, we aim to enable meta-learning in task-unsegmented settings, operating directly on time series in which the latent task undergoes discrete, unobserved switches, rather than requiring a pre-segmented meta-dataset.

3 Contributions

The primary contribution of this work is an algorithmic framework for task unseg-mented meta-learning which we refer to as meta-learning via online change-point analysis (MOCA).MOCA wraps arbitrary meta-learning algorithms in a differentiable changepoint estimation algorithm, enabling application of meta-learning algorithms directly to problems in the continuous learning setting. By backpropagating through the changepoint estimation framework, MOCA learns botha rapidly adaptive underlying predictive model (in the form of the meta-learning model), as well as an effective changepoint detection algorithm. MOCA is a generic framework which can be paired with many existing meta-learning algorithms. We demonstrate the performance of MOCA on both regression and classification settings with unobserved task switches. Both our problem setting and an illustration of the MOCA algorithm are presented.

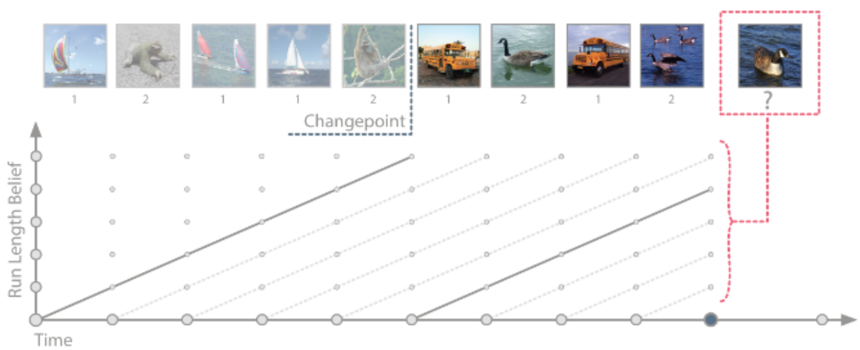

Figure 1: An illustration of a simplified version of our problem setting and of the MOCA algorithm. We observe a time series of data, in which the data $\boldsymbol{x}$ is presented (in this case, an image), based on which a probabilistic prediction is made (which we denote $\hat{\boldsymbol{y}}$), after which the true label $\boldsymbol{y}$ is received (in this case, class labels taking value 1 or 2). Unobserved change in the underlying task (here labeled "changepoint" result in changes to the generative model of $\boldsymbol{x}$ and/or $\boldsymbol{y}$. In the above image, the images corresponding to label 1 switch from sailboats to school buses, while the images corresponding to label 2 switch from sloths to geese[1]. MOCA recursively estimates the time since the last changepoint, and conditions an underlying meta-learning model only on data that is relevant to the current task.

4 Problem Formulation

The goal is to optimize a learning agent to perform well in this setting. Let $p_\theta(\hat{y}_t|x_{1:t}, y_{1:t1})$ by the agent's prediction for y_t given input x_t and the past labeled examples. We will evaluate the learner's performance through a negative log likelihood loss, and our objective is as follows: We assume that we have access to a representative time series generated in the same manner from the same distribution of tasks, and use this time series to optimize θ in an offline, meta-training phase. Critically, however, in stark contrast to standard meta-learning approaches, we do not assume access to task segmentation, i.e. that

$$
\begin{aligned}
\min_{\boldsymbol{\theta}} \quad & \mathbb{E}\left[\sum_{t=1}^{\infty} -\log p_{\boldsymbol{\theta}}(\boldsymbol{y}_t \mid \boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1})\right] \\
\text{subject to} \quad & \boldsymbol{x}_t, \boldsymbol{y}_t \sim \mathcal{T}_t, \quad \mathcal{T}_t = \begin{cases} \mathcal{T}_{t-1} & \text{w.p. } 1-\lambda \\ \mathcal{T}_{t,\text{new}} & \text{w.p. } \lambda \end{cases} \\
& \mathcal{T}_1 \sim p(\mathcal{T}), \quad \mathcal{T}_{t,\text{new}} \sim p(\mathcal{T})
\end{aligned}
$$

this offline data is pre-grouped by latent parameter $\mathcal{T}$. Moreover, we highlight that we consider the case of individual data points provided sequentially, in contrast to the common "k-shot,n-way" problem setting prevalent in few-shot learning (especially classification). Our setting may easily be extended to the setting in which multiple data points are observed simultaneously.

Algorithm 1 Meta-Learning via Online Changepoint Analysis: Training

Require: Training data $\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}$, number of training iterations N, initial model parameters $\boldsymbol{\theta}$

```
for i = 1 to N do
    Sample training batch x_{1:T}, y_{1:T} from the full timeseries.
    Initialize belief over run length b_1(r_1 = 0) = 1
    Initialize posterior statistics η_0[r = 0] according to θ
    for t = 1 to T do
        Observe x_t
        Compute b_t(r_t | x_t) according to (3)
        Predict p_θ(ŷ_t | x_{1:t}, y_{1:t-1}) according to (7)
        Observe y_t
        Incur NLL loss ℓ_t = − log p_θ(y_t | x_{1:t}, y_{1:t-1})
        Compute updated posteriors η_t[r_t] for all r_t according to (8)
        Compute b_t(r_t | x_t, y_t) according to (4)
        Compute updated belief over run length b_{t+1} according to (6) and (5)
    end for
    Compute ∇_θ Σ_{t=k}^{k+T} ℓ_t and perform gradient descent update to θ
end for
```