Efficient Meta Learning via Minibatch Proximal Update

Pan Zhou

Joint work with Xiao-Tong Yuan, Huan Xu, Shuicheng Yan, Jiashi Feng

National University of Singapore

pzhou@u.nus.edu

Dec 11, 2019
Meta Learning via Minibatch Proximal Update (Meta-MinibatchProx)

Meta-MinibatchProx learns a good prior model initialization $\mathcal{W}$ from observed tasks such that

$\mathcal{W}$ is close to the optimal models of new similar tasks, promoting new task learning.
Meta Learning via Minibatch Proximal Update (Meta-MinibatchProx)

**Meta-MinibatchProx** learns a good *prior model initialization* $\mathbf{w}$ from observed tasks such that $\mathbf{w}$ is close to the optimal models of new similar tasks, promoting new task learning.

- **Training model:** given a task distribution $\mathcal{T}$, we minimize a **bi-level** meta learning model

$$
\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \min_{\mathbf{w}_{T_i}} \mathcal{L}_{D_{T_i}} (\mathbf{w}_{T_i}) + \frac{\lambda}{2} \left\| \mathbf{w} - \mathbf{w}_{T_i} \right\|^2_2,
$$

where each task $T_i \sim \mathcal{T}$ has $K$ training samples $D_{T_i} = \{(x_s, y_s)\}_{s=1}^{K}$

$$
\mathcal{L}_{D_{T_i}} = \frac{1}{K} \sum_{(x, y) \in D_{T_i}} \ell(f(\mathbf{w}, x), y)
$$
is empirical loss with predictor $f$ and loss $\ell$. 
Meta Learning via Minibatch Proximal Update (Meta-MinibatchProx)

**Meta-MinibatchProx** learns a good **prior model initialization** $\mathbf{w}$ from observed tasks such that $\mathbf{w}$ is close to the optimal models of new similar tasks, promoting new task learning.

- **Training model**: given a task distribution $\mathcal{T}$, we minimize a **bi-level** meta learning model

$$
\min_{\mathbf{w}} \quad \frac{1}{n} \sum_{i=1}^{n} \min_{\mathbf{w}_{T_i}} \mathcal{L}_{D_{T_i}}(\mathbf{w}_{T_i}) + \frac{\lambda}{2} \| \mathbf{w} - \mathbf{w}_{T_i} \|_2^2,
$$

where each task $T_i \sim \mathcal{T}$ has $K$ training samples $D_{T_i} = \{(x_s, y_s)\}_{s=1}^K$

$$\mathcal{L}_{D_{T_i}} = \frac{1}{K} \sum_{(x,y) \in D_{T_i}} \ell(f(\mathbf{w}, x), y)$$

is empirical loss with predictor $f$ and loss $\ell$. 
Meta Learning via Minibatch Proximal Update (Meta-MinibatchProx)

Meta-MinibatchProx learns a good prior model initialization $\mathbf{w}$ from observed tasks such that $\mathbf{w}$ is close to the optimal models of new similar tasks, promoting new task learning.

- **Training model:** given a task distribution $\mathcal{T}$, we minimize a bi-level meta learning model

$$
\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \min_{\mathbf{w}_{T_i}} \mathcal{L}_{D_{T_i}}(\mathbf{w}_{T_i}) + \frac{\lambda}{2} \|\mathbf{w} - \mathbf{w}_{T_i}\|_2^2
$$

where each task $T_i \sim \mathcal{T}$ has $K$ training samples $D_{T_i} = \{(x_s, y_s)\}_{s=1}^{K}$

$$
\mathcal{L}_{D_{T_i}} = \frac{1}{K} \sum_{(x, y) \in D_{T_i}} \ell(f(\mathbf{w}, x), y)
$$
is empirical loss with predictor $f$ and loss $\ell$. 
Meta Learning via Minibatch Proximal Update (Meta-MinibatchProx)

Meta-MinibatchProx learns a good prior model initialization $\mathbf{w}$ from observed tasks such that $\mathbf{w}$ is close to the optimal models of new similar tasks, promoting new task learning.

- **Training model:** given a task distribution $\mathcal{T}$, we minimize a bi-level meta learning model

$$
\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \min_{\mathbf{w}_T} \mathcal{L}_{D_{T_i}}(\mathbf{w}_T) + \frac{\lambda}{2} \| \mathbf{w} - \mathbf{w}_{T_i} \|_2^2,
$$

where each task $T_i \sim \mathcal{T}$ has $K$ training samples $D_{T_i} = \{(x_s, y_s)\}_{s=1}^{K}$.

$$
\mathcal{L}_{D_{T_i}} = \frac{1}{K} \sum_{(x,y) \in D_{T_i}} \ell(f(\mathbf{w}, x), y)
$$

is empirical loss with predictor $f$ and loss $\ell$.

Small average distance to optimum models of all tasks in expectation.
Meta Learning via Minibatch Proximal Update (Meta-MinibatchProx)

**Meta-MinibatchProx** learns a good **prior model initialization** $\mathbf{w}$ from observed tasks such that $\mathbf{w}$ is close to the optimal models of new similar tasks, promoting new task learning.

- **Test model:** given a randomly sampled task $T \sim \mathcal{T}$ consisting of $K$ samples $D_T = \{(x_s, y_s)\}_{s=1}^{K}$

  $$\min_{\mathbf{w}_T} \mathcal{L}_{D_T}(\mathbf{w}_T) + \frac{\lambda}{2} \| \mathbf{w}^* - \mathbf{w}_T \|_2^2,$$

  where $\mathbf{w}^*$ denotes the learnt prior initialization.
Meta Learning via Minibatch Proximal Update (Meta-MinibatchProx)

Meta-MinibatchProx learns a good prior model initialization $\mathbf{w}$ from observed tasks such that $\mathbf{w}$ is close to the optimal models of new similar tasks, promoting new task learning.

- **Test model:** given a randomly sampled task $T \sim \mathcal{T}$ consisting of $K$ samples $D_T = \{(x_s, y_s)\}_{s=1}^{K}$

  $$\min_{\mathbf{w}_T} \mathcal{L}_{D_T}(\mathbf{w}_T) + \frac{\lambda}{2} \| \mathbf{w}^* - \mathbf{w}_T \|^2_2,$$

  where $\mathbf{w}^*$ denotes the learnt prior initialization.

- **Benefit:** a few data is sufficient for adaptation

  the learnt prior initialization $\mathbf{w}^*$ is close to optimum $\mathbf{w}_T$ when training and test tasks are sampled from the same distribution.
Optimization Algorithm

We use SGD based algorithm to solve bi-level training model:

$$\min_w \left\{ F(w) := \frac{1}{n} \sum_{i=1}^{n} \min_{w_{T_i}} \mathcal{L}_{D_{T_i}}(w_{T_i}) + \frac{\lambda}{2} \| w - w_{T_i} \|_2^2 \right\}$$
We use SGD based algorithm to solve bi-level training model:

\[
\min_w \{ F(w) := \frac{1}{n} \sum_{i=1}^{n} \min_{w_{T_i}} \mathcal{L}_{D_{T_i}}(w_{T_i}) + \frac{\lambda}{2} \| w - w_{T_i} \|_2^2 \}
\]

- Step1. select a mini-batch of task \( \{T_i\} \) of size \( b_s \).
We use SGD based algorithm to solve bi-level training model:

$$\min_w \left\{ F(w) := \frac{1}{n} \sum_{i=1}^{n} \min_{w_{T_i}} \mathcal{L}_{D_{T_i}}(w_{T_i}) + \frac{\lambda}{2} \|w - w_{T_i}\|_2^2 \right\}$$

- Step1. select a mini-batch of task $\{T_i\}$ of size $b_s$.

- Step2. for $T_i$, compute an approximate minimizer:

$$w_{T_i} \approx \arg\min_{w_{T_i}} \{g(w_{T_i}) := \mathcal{L}_{D_{T_i}}(w_{T_i}) + \frac{\lambda}{2} \|w - w_{T_i}\|_2^2\}, \text{ namely } \|\nabla g(w_{T_i})\|_2^2 \leq \epsilon_s$$
We use SGD based algorithm to solve bi-level training model:

$$\min_{w} \left\{ F(w) := \frac{1}{n} \sum_{i=1}^{n} \min_{w_{T_i}} \mathcal{L}_{D_{T_i}}(w_{T_i}) + \frac{\lambda}{2} \|w - w_{T_i}\|_2^2 \right\}$$

- Step 1. select a mini-batch of task \(\{T_i\}\) of size \(b_s\).
- Step 2. for \(T_i\), compute an approximate minimizer:

  \[ w_{T_i} \approx \arg\min_{w_{T_i}} \{ g(w_{T_i}) := \mathcal{L}_{D_{T_i}}(w_{T_i}) + \frac{\lambda}{2} \|w - w_{T_i}\|_2^2 \}, \text{ namely } \|\nabla g(w_{T_i})\|_2^2 \leq \epsilon_s \]

- Step 3. update the prior initialization model:

  \[ w = w - \eta_s \lambda (w - \frac{1}{b_s} \sum_{i=1}^{b_s} w_{T_i}) \]
Optimization Algorithm

We use SGD based algorithm to solve bi-level training model:

\[
\min_w \left\{ F(w) := \frac{1}{n} \sum_{i=1}^{n} \min_{w_{T_i}} \mathcal{L}_{D_{T_i}}(w_{T_i}) + \frac{\lambda}{2} \|w - w_{T_i}\|_2^2 \right\}
\]

- Step1. select a mini-batch of task \( \{T_i\} \) of size \( b_s \).

- Step2. for \( T_i \), compute an approximate minimizer:

\[
w_{T_i} \approx \arg\min_{w_{T_i}} \{ g(w_{T_i}) := \mathcal{L}_{D_{T_i}}(w_{T_i}) + \frac{\lambda}{2} \|w - w_{T_i}\|_2^2 \}, \text{ namely } \|\nabla g(w_{T_i})\|_2^2 \leq \epsilon_s
\]

- Step3. update the prior initialization model:

\[
w = w - \eta_s \lambda \left( w - \frac{1}{b_s} \sum_{i=1}^{b_s} w_{T_i} \right)
\]

**Theorem 1 (convergence guarantees, informal).**

1. Convex setting, i.e. convex \( \phi_{D_{T_i}}(w) \). We prove \( \mathbb{E}[\|w^S - w^*\|_2^2] \leq O\left(\frac{1}{S}\right) \).

2. Nonconvex setting, i.e. smooth \( \phi_{D_{T_i}}(w) \). We prove \( \mathbb{E}_s[\|\nabla F(w^s)\|_2^2] \leq O\left(\frac{1}{\sqrt{S}}\right) \).
Generalization Performance Guarantee

• Ideally, for a given task $T \sim \mathcal{T}$, one should train the model on the population risk
  \[
  \text{Population solution: } \mathbf{w}^*_T, P = \arg\min_{\mathbf{w}_T} \left\{ \mathcal{L}(\mathbf{w}_T) := \mathbb{E}_{(x,y) \sim T} \ell(f(\mathbf{w}_T, x), y) \right\}.
  \]

• In practice, we have only $K$ samples and adapt the learnt prior model $\mathbf{w}^*$ to the new task:
  \[
  \text{Empirical solution: } \mathbf{w}^*_T = \arg\min_{\mathbf{w}_T} \mathcal{L}_{D_T}(\mathbf{w}_T) + \frac{\lambda}{2} \|\mathbf{w}^* - \mathbf{w}_T\|_2^2.
  \]

• Since $\mathbf{w}^*_T, P \neq \mathbf{w}^*_T$, why $\mathbf{w}^*_T$ is good for generalization in few-shot learning problem?
Generalization Performance Guarantee

• Ideally, for a given task $T \sim \mathcal{T}$, one should train the model on the population risk
  Population solution: $w^*_{T,P} = \arg\min_{w_T} \{ \mathcal{L}(w_T) := \mathbb{E}_{(x,y) \sim T} \ell(f(w_T, x), y) \}$.

• In practice, we have only $K$ samples and adapt the learnt prior model $w^*$ to the new task:
  Empirical solution: $w^*_T = \arg\min_{w_T} \mathcal{L}_{D_T}(w_T) + \frac{\lambda}{2} \|w^* - w_T\|_2^2$.

• Since $w^*_{T,P} \neq w^*_T$, why $w^*_T$ is good for generalization in few-shot learning problem?

Theorem 2 (generalization performance guarantee, informal).
Suppose each loss $\phi_{D_{T,i}}(w)$ is convex and is smooth. Let $D_T = \{(x_i, y_i)\}_{i=1}^K \sim T$. Then we have
\[ \mathbb{E}_{T \sim \mathcal{T}} \mathbb{E}_{D_T \sim T} (\mathcal{L}(w^*_T) - \mathcal{L}(w^*_{T,P})) \leq \frac{c}{\sqrt{K}} \mathbb{E}[\|w^* - w^*_{T,P}\|_2^2] \] with a constant $c$. (1)

Remark: strong generalization performance, as our training model guarantees
the learnt prior $w^*$ is close to the optimum model $w^*_{T,P}$. 
Experimental results

**Few-shot regression**: smaller mean square error (MSE) between prediction and ground truth

(a) Visual illustration

(b) MSE

**Few-shot classification**: higher classification accuracy

- **miniImageNet**
  - 1-shot 5-way: 0.8%
  - 5-shot 5-way: 1.44%
  - 1-shot 5-way: 3.31%
  - 5-shot 5-way: 1.15%

- **tieredImageNet**
  - 1-shot 5-way: 2.41%
  - 5-shot 5-way: 5.15%
  - 1-shot 10-way: 1.12%
  - 5-shot 10-way: 1.18%
POSTER # 26

05:00 -- 07:00 PM @ East Exhibition Hall B + C

Thanks!