Meta-Learning Probabilistic Inference for Prediction

J. Gordon, J. Bronskill, M. Bauer, S. Nowozin, R. E. Turner

ICLR 2019

presented by: Patrik Barkman
Few-shot learning

Source

Target
Meta-Learning
(Learning to learn)
Meta-Learning Probabilistic Inference for Prediction

- General probabilistic framework for few-shot learning
- Neural network based implementation of the framework
- New state-of-the art in few-shot learning benchmarks
Outline

▶ Background

▶ Probabilistic framework

▶ Implementation

▶ Experiments

▶ Summary
Background

- Siamese networks (2015)
- Matching networks (2016)
- Prototypical networks (2017)
- Model-agnostic meta-learning (2017)
- Meta-Learner LSTM (2017)
Background

- Siamese networks (2015)
- Matching networks (2016)
- Prototypical networks (2017)
- Model-agnostic meta-learning (2017)
- Meta-Learner LSTM (2017)

Background

- Siamese networks (2015)
- Matching networks (2016)
- Prototypical networks (2017)
- Model-agnostic meta-learning (2017)
- Meta-Learner LSTM (2017)

\[ P(\hat{y}|\hat{x}, S) = \sum_{i=1}^{k} a(\hat{x}, x_i, S)y_i \]
Background

- Siamese networks (2015)
- Matching networks (2016)
- Prototypical networks (2017)
- Model-agnostic meta-learning (2017)
- Meta-Learner LSTM (2017)

Background

- Siamese networks (2015)
- Matching networks (2016)
- Prototypical networks (2017)
- Model-agnostic meta-learning (2017)
- Meta-Learner LSTM (2017)

Background

- Siamese networks (2015)
- Matching networks (2016)
- Prototypical networks (2017)
- Model-agnostic meta-learning (2017)
- Meta-Learner LSTM (2017)

Ravi, S., & Larochelle, H. Optimization as a Model for Few-Shot Learning. ICLR (2017)
Probabilistic framework

Meta-Learning Probabilistic Inference for Prediction (ML-PIP)
Probabilistic multi-task learning

\[
p(\tilde{y}^{(t)}|\tilde{x}^{(t)}, \theta) = \int p(\tilde{y}^{(t)}|\tilde{x}^{(t)}, \psi^{(t)}, \theta) p(\psi^{(t)}|\tilde{x}, D^{(t)}, \theta) \, d\psi^{(t)}
\]
Approximating the predictive distribution

\[ p(\tilde{y}^{(t)}|\tilde{x}^{(t)}, D^{(t)}, \theta) = \int p(\tilde{y}^{(t)}|\tilde{x}^{(t)}, \psi^{(t)}, \theta)p(\psi^{(t)}|\tilde{x}, D^{(t)}, \theta) \, d\psi^{(t)} \]

1. Approximate posterior distribution

\[ p(\psi^{(t)}|\tilde{x}^{(t)}, D^{(t)}, \theta) \approx q_\phi(\psi^{(t)}|D^{(t)}, \theta) \]

e.g. \( \psi^{(t)} \sim \mathcal{N}(\mu, \sigma), \{\mu, \sigma\} = f(D^{(t)}; \phi) \)

2. Compute approximate predictive distribution

\[ q_\phi(\tilde{y}^{(t)}|\tilde{x}^{(t)}, D^{(t)}, \theta) = \int p(\tilde{y}^{(t)}|\tilde{x}^{(t)}, \psi^{(t)}, \theta)q_\phi(\psi^{(t)}|D^{(t)}, \theta) \, d\psi^{(t)} \]

e.g. using Monte Carlo sampling
Meta-learning the predictive distribution

\[ q_\phi(\tilde{y}(t)|\tilde{x}(t), D(t), \theta) = \int p(\tilde{y}(t)|\tilde{x}(t), \psi(t), \theta)q_\phi(\psi(t)|D(t), \theta) \, d\psi(t) \]

Consider tasks as samples from some distribution

\[ D, \tilde{x}, \tilde{y} \sim p(D, \tilde{x}, \tilde{y}) \]

Minimize expected divergence

\[ \min_{\phi, \theta} \mathbb{E}_{p(D, \tilde{x})} \left[ \text{KL} \left[ p(\tilde{y}|\tilde{x}, D, \theta) \parallel q_\phi(\tilde{y}|\tilde{x}, D, \theta) \right] \right] \]
Meta-learning the predictive distribution

\[
\mathcal{L}(\theta, \phi) = \mathbb{E}_{p(D, \tilde{x}, \tilde{y})} \left[ \log \int p(\tilde{y} | \tilde{x}, \psi, \theta) q_\phi(\psi \mid D, \theta) \, d\psi \right]
\]

\[
\hat{\mathcal{L}}(\theta, \phi) = \frac{1}{MT} \sum_{m,t} \log \frac{1}{L} \sum_{l} p(\tilde{y}_m^{(t)} \mid \tilde{x}_m^{(t)}, \psi_l^{(t)}, \theta) \quad \psi_l^{(t)} \sim q_\phi(\psi \mid D^{(t)}, \theta) \\
D^{(t)}, \tilde{x}_m^{(t)}, \tilde{y}_m^{(t)} \sim p(D^{(t)}, \tilde{x}_m^{(t)}, \tilde{y}_m^{(t)})
\]
Inference

Given a new dataset $D$ and test input $x$

1. Sample $L$ task-specific parameters

   $$\psi_l \sim q_\phi(\psi_l|D, \theta)$$

2. Estimate predictive distribution

   $$\hat{q}_\phi(y|x, D, \theta) = \frac{1}{L} \sum_{l=1}^{L} p(y|x, \psi_l, \theta)$$
Unification

- Gradient-based Meta-Learning (MAML, Meta-Learner LSTM)
- Metric-based few-shot learning (Prototypical networks, Matching networks)
- Amortized MAP inference (hyperfine networks)
- Conditional models trained via maximum likelihood (neural processes)
Implementation

Versatile Amortized Inference (VERSA)
A versatile system

Inference system that is rapid and flexible

amortization network $\rightarrow$ rapid flexibility?
Flexibility challenges

- Datasets as input (i.e. unordered sets as input)
- Different types of tasks (e.g. number of classes)
- High dimensional output space (i.e. many parameters)
Sets as inputs

permutation-invariant instance-pooling

\[ f(\{x_1, \ldots, x_n\}) \approx g(h(x_1), \ldots, h(x_n)) \]

Few-shot Classification

$N$-way, $k$-shot learning

= discriminate between $N$ classes given $k$ examples of each class.
Few-shot Classification

\( N \)-way, \( k \)-shot learning

= discriminate between \( N \) classes given \( k \) examples of each class.

What if \( N \) and \( k \) varies between tasks?
Few-shot Classification

Let $\psi \in \mathbb{R}^{d \times C}$ be the parameters of a linear classifier.

Assume context independency

$$q_{\phi}(\psi|D, \theta) \approx \prod_{c=1}^{C} q_{\phi}(\psi_c|\{h_{\theta}(x_n^c)\}_{n=1}^{k_c}, \theta)$$

Theoretical support from density estimation and empirically justified for $h_{\theta}$ with sufficient capacity
Experiments

Toy-data
Image classification
Image reconstruction

https://github.com/Gordonjo/versa
Toy-data

Ground truth model

$$p(\theta) = \delta(\theta), \quad p(\psi^{(t)}|\theta) = \mathcal{N}(\psi^{(t)}; \theta, \sigma^2_{\psi})$$

$$(y_{n}^{(t)}|\psi^{(t)}) = \mathcal{N}(y_{n}^{(t)}; \psi^{(t)}, \sigma^2_{y})$$
Toy-data

Ground truth model

\[ p(\theta) = \delta(\theta), \quad p(\psi^{(t)}|\theta) = \mathcal{N}(\psi^{(t)}; \theta, \sigma^2_{\psi}) \]

\[ (y^{(t)}_n|\psi^{(t)}) = \mathcal{N}(y^{(t)}_n; \psi^{(t)}, \sigma^2_y) \]

\[ \implies p(\psi^{(t)}|D^{(t)}, \sigma^2_y) = \mathcal{N}(\psi^{(t)}; \hat{\mu}, \hat{\sigma}^2) \]

\[ \hat{\mu} = \hat{\sigma}^2 \left( \frac{1}{\sigma^2_y} \sum_{n=1}^{N} y^{(t)}_n + \frac{\theta}{\sigma^2_{\psi}} \right) \]

\[ \frac{1}{\hat{\sigma}^2} = \frac{1}{\sigma^2_{\psi}} + \frac{N}{\sigma^2_y} \]

Amortization model

\[ q_\phi(\psi|D^{(t)}) = \mathcal{N}(\psi; \mu^{(t)}_q, \sigma^{(t)}_q) \]

\[ \mu^{(t)}_q = w_{\mu} \sum_{n=1}^{N} y^{(t)}_n + b_{\mu}, \quad \sigma^{(t)}_q = \exp \left( w_{\sigma} \sum_{n=1}^{N} y^{(t)}_n + b_{\sigma} \right) \]
Toy-data

\[ T = 250 \text{ tasks, } k \in \{5, 10\} \text{ shots, } M = 15 \text{ test observations.} \]
Image Classification

**Omniglot**
- 1623 characters
- 50 languages
- 20 instances for each character

**miniImageNet**
- 60,000 images
- 100 classes
- 600 instances for each class
Image classification

Feature extraction
\[ \tilde{x} \rightarrow h_{\theta}(\tilde{x}) \]

Amortization Network
\[ h_{\theta}(x_1^{(1)}) \ldots h_{\theta}(x_{k_1}^{(1)}) \]

Linear Classifier
\[ \begin{pmatrix} w_1^{(1)} & \cdots & w_l^{(C)} \end{pmatrix} \]

\[ \phi \]

Softmax output
\[ p(\hat{y}|\tilde{x}, \theta, \psi_t) \]

\[ h_{\theta}(x_1^{(C)}) \ldots h_{\theta}(x_{k_C}^{(C)}) \]

\[ k_1 \text{ train examples from class 1} \]

\[ k_C \text{ train examples from class } C \]
Image classification

New state-of-the-art

- 20-way, 1-shot Omniglot (97.66%, ▲ 0.02%)
- 5-way 5-shot miniImageNet (67.37%, ▲ 1.38%)

On par with state-of-the-art

- 5-way, 1-shot Omniglot (99.70%)
- 5-way, 5-shot Omniglot (99.75%)
- 5-way 1-shot miniImageNet (53.40%)

Worse than state-of-the-art

- 20-way 5-shot Omniglot (98.77%, ▼ 0.59%)
Image classification

Performance is robust to variations in “way” and “shots”
Image classification

ML-PIP

\[ \mathcal{L}_{\text{ML-PIP}} = \frac{1}{T} \sum_{t=1}^{T} \frac{1}{M_t} \sum_{m=1}^{M_t} \log \frac{1}{L} \sum_{l=1}^{L} p(\tilde{y}_m^{(t)} | \tilde{x}_m^{(t)}, \psi^{(t)}_l, \theta) \]

Variational inference

\[ \mathcal{L}_{\text{VI}} = \frac{1}{T} \sum_{t=1}^{T} \left( \sum_{(x,y) \in D^{(t)}} \left( \frac{1}{L} \sum_{l=1}^{L} \log p(y|x, \psi^{(l)}, \theta) \right) - KL \left[ q_{\phi}(\psi|D^{(t)}, \theta) \parallel p(\psi|\theta) \right] \right) \]

<table>
<thead>
<tr>
<th>Method</th>
<th>Omniglot</th>
<th>5-way NLL</th>
<th>20-way NLL</th>
<th>miniImageNet</th>
<th>5-way NLL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-shot</td>
<td>5-shot</td>
<td>1-shot</td>
<td>5-shot</td>
<td>1-shot</td>
</tr>
<tr>
<td>Amortized VI</td>
<td>0.179 ± 0.009</td>
<td>0.137 ± 0.004</td>
<td>0.456 ± 0.010</td>
<td>0.253 ± 0.004</td>
<td>1.328 ± 0.024</td>
</tr>
<tr>
<td>Non-Amortized VI</td>
<td>0.144 ± 0.005</td>
<td>0.025 ± 0.001</td>
<td>0.393 ± 0.005</td>
<td>0.078 ± 0.002</td>
<td>1.183 ± 0.023</td>
</tr>
<tr>
<td><strong>VERSA</strong></td>
<td>0.010 ± 0.005</td>
<td>0.007 ± 0.003</td>
<td>0.079 ± 0.009</td>
<td>0.031 ± 0.004</td>
<td>0.859 ± 0.015</td>
</tr>
</tbody>
</table>
Image reconstruction

Given an image of an object, produce an image of the object in any rotation
Image reconstruction

ShapeNetCore v2

- 12 object categories
- 37,108 objects
- 36 views for each object

https://www.shapenet.org/
Image reconstruction
Image reconstruction

<table>
<thead>
<tr>
<th>Model</th>
<th>MSE</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-VAE 1-shot</td>
<td>0.0269</td>
<td>0.5705</td>
</tr>
<tr>
<td>VERSA 1-shot</td>
<td>0.0108</td>
<td>0.7893</td>
</tr>
<tr>
<td>VERSA 5-shot</td>
<td>0.0069</td>
<td>0.8483</td>
</tr>
</tbody>
</table>

MSE = mean square error  
SSIM = structural similarity index
Image reconstruction
Summary

- Unifying probabilistic framework
- Flexible and rapid implementation
- Tested on
  - Image classification
  - Image reconstruction
- New state-of-the-art