View Author Feedback

Paper ID
6885

Paper Title
Learning to Learn Kernels with Variational Random Features

AUTHOR FEEDBACK QUESTIONS

1. Author Response to Reviewers Please use this space to respond to any questions raised by reviewers, or to clarify any misconceptions. Please do not include any links to external material, nor include "late-breaking" results that are not responsive to reviewer concerns. We request that you understand that this year is especially difficult for many people, and to be considerate in your response.
We thank all reviewers for their constructive and considerate feedback in judging our work on learning adaptive kernels via variational Random Fourier Features (RFFs) for meta-learning “an interesting take on this problem” (R3), “novel and well-motivated” (R4), “original and efficient” (R5) and “could open new avenues for meta-learning research that draw from more theoretical work” (R6). We further clarify presentation, motivation and experiments next.

* Presentation (R3, R4, R6)
We agree with R6 narrowing down the scope of Related Works to focus on similarity metrics in meta-learning would strengthen the paper. Together with the references suggested by R3 and R4 this will better highlight our key contribution. Thank you. Betinetto et al. are already cited. They learn a closed-form solver for few-shot learning, while we learn adaptive kernels with variational RFFs. Wenliang et al. train kernels by meta-learning for density estimation, not for few-shot learning. Tossou et al. learn and adapt deep embeddings for few-shot regression; we learn adaptive kernels with variational RFFs for both few-shot regression and classification. Qiao et al. learn an adaptive distance metric by an embedding function with similar/dissimilar pair-constraint and regularization. We learn adaptive kernels by inferring task-specific RFFs.

* Random Fourier Features Motivation and Clarification (R3)
We use RFFs that learn task-specific kernels to improve each few-shot task while maintaining efficiency. While the number of shots n is small per task, we usually need to train on ~1.2M tasks. Modeling bandwidth is also an option, at the expense of manipulating exponential functions, which we avoid by using a dot product to compute kernels. We agree the external feature extractor is important, we show there is always room to improve them further by our task-adaptation. In the sine experiments, similarly to classification, MetaVRF produces kernels for KRR to fit sine functions, not to pick up the structure of the data. We will clarify this in the paper.

* Variational Inference Formula Clarification (R3, R4, R5, R6)
Our formula is able to learn other kernels by specifying their parameters as latent variables. The shared knowledge is the Fourier basis. We tried other attention mechanisms, like DotProd, but Laplace is best. Variable b in Eq.7 used to generate the RFFs, does not affect ELBO. To R3, not Eq. 8, but its ELBO in Eq.12 is optimised, which explicitly relates to the kernel closed form solution via Eq. 4 and 5. As R4 and R5 suggest, the parameter \lambda can indeed be adapted. Thank you. In response to R6, the conditional prior p(w|x,S) takes a diagonal Gaussian form (see Bochner’s theorem). It is parameterised by a prior net (Table C.6), taking as input the vector representation obtained by cross attention of x and S (Eq.7 in supplemental), outputting the Gaussian mean and variance. The log-likelihood is implemented as a cross-entropy loss between prediction and ground truth, it is not directly taking the logits. We will update the paper.

* LSTM Motivation and Clarification (R3, R4, R5)
The motivation for LSTM to model task dependency, over alternatives like graph attention, is that its sequential nature enables handling a huge number of related tasks during training.
In our few-shot learning, all ~1.2M related train tasks (8 tasks/batch*150K batches) are sampled from train data. It

gradually accumulates prior knowledge in the cell state, which is updated by each task and passed across batches. The LSTM memory carries the accumulated knowledge and its final state forms an inductive bias for new test tasks. For the Bi-LSTM the inputs are the current support set, the state and output from both directions (Eq.14). The Bi-LSTM reduces the impact of task order by receiving information from both directions. Because we randomly sample a bunch of episodes in each iteration, order matters even less. The batch size does not impact results much, as long as it is not too small.

* Experiments (R6)
Following standard few-shot settings, support set S is sampled randomly from the meta-test set. This affects results, but to minimize bias we sample 2000 tasks for test (where 600 is typical in previous works). Shots in the 100-way setting are aligned with training settings, i.e., the same number (1 or 5) of shots. The performance is not surprising, as our method essentially learns similarity, which is more generalisable to a larger number of classes. We acknowledge MetaVRF needs more compute than ProtoNet, due to the kernel learning step. The benefit is higher accuracy, e.g., 54.2% vs 47.4% on mini-ImageNet. MetaVRF performs well with low sampling rates because it offers an adaptive similarity metric suited to each task. We will also add the suggested fixed RBF kernel baseline: 46.7% and 60.3% on mini-ImageNet, much lower than our 54.2% and 67.8%. Thank you.

We will fix the remaining typos, make table fonts consistent, and thank all reviewers.

---

3. I certify that this author response conforms to the ICML Code of Conduct (https://www.icml.cc/public/CodeOfConduct)
Agreement accepted
