2nd Symposium on Advances in Approximate Bayesian Inference, 2019 1–13

# Meta-Learning for Variational Inference

**Ruqi Zhang** * RZ297@CORNELL.EDU
*Cornell University*

**Yingzhen Li** YINGZHEN.LI@MICROSOFT.COM
*Microsoft Research, Cambridge*

**Christopher De Sa** CDESA@CS.CORNELL.EDU
*Cornell University*

**Sam Devlin** SAM.DEVLIN@MICROSOFT.COM
*Microsoft Research, Cambridge*

**Cheng Zhang** CHENG.ZHANG@MICROSOFT.COM
*Microsoft Research, Cambridge*

## 1. Introduction

Variational inference (VI) is critical for learning probabilistic models (Jordan et al., 1999; Zhang et al., 2018). VI approximates the target distribution by minimizing a divergence objective. Different divergence metrics essentially define different inference algorithms which lead to different properties of the approximation. Therefore the selection of this divergence is one of the crucial factors of making VI successful. The most widely used divergence measure is $\mathrm{KL}(q||p)$ where $p$ is the target distribution and $q$ is the approximated distribution. However, using this KL divergence for VI has been criticized for under-estimating the uncertainty (Bishop, 2006; Blei et al., 2017; Wang et al., 2018), which leads to poor model performance when uncertainty estimation is essential. Many alternative divergence measures have been proposed for VI to alleviate this issue (Minka et al., 2005; Hernández-Lobato et al., 2016; Li and Turner, 2016; Csiszár et al., 2004; Bamler et al., 2017; Wang et al., 2018), which provide better bias and variance trade-offs and lead to better predictive results with more accurate uncertainty estimation.

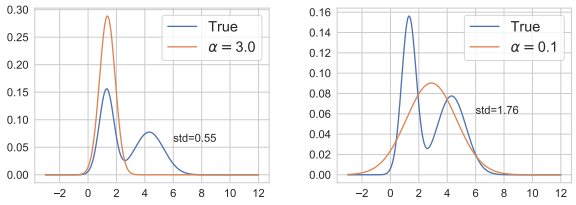



Figure 1: An illustration of approximated distributions to a Gaussian mixture by minimizing $\alpha$-divergence with different $\alpha$.

However, as illustrated by Figure 1, the optimal divergence can vary depending on tasks. Unfortunately, choosing a suitable divergence objective for a specific task is challenging as it requires a thorough understanding of the shape of the target distribution and the desirable properties of the approximated distribution, as well as time-consuming parameter tuning. A crucial question remains to be addressed is: can we automatically choose a suitable divergence which are tailored to specific type of tasks?

* Work done as an intern in Microsoft Research Cambridge.

To answer this question, we propose meta-learning for variational inference (*meta-VI*) which utilizes the advantages of meta-learning to improve approximate Bayesian inference. Meta-learning is to design a learner based on several training tasks that can generalize well to future tasks (Naik and Mammone, 1992; Thrun and Pratt, 2012; Hochreiter et al., 2001). Our meta-VI learns an inference algorithm that is tailored to the problem of interest. Additionally, meta-VI can provide a good initialization of the variational parameters which reduces the training time remarkably.

## 2. Preliminaries

Bayesian inference requires computing the posterior over $\theta$ given the dataset $\mathcal{D}$: $p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}$. The exact posterior $p(\theta|\mathcal{D})$ is generally intractable. Using VI, the approximated posterior $q(\theta)$ is obtained by minimizing a divergence, e.g. $\mathrm{KL}(q(\theta)||p(\theta|\mathcal{D}))$. This turns Bayesian inference into an optimization task (divergence minimization). In practice, VI alternatively maximizes an equivalent objective called the *variational lower bound*:

$$\mathcal{L}_{\mathrm{VI}} = \mathbf{E}_q \left[ \log \frac{p(\theta, \mathcal{D})}{q(\theta)} \right] = \log p(\mathcal{D}) - \mathrm{KL}(q||p) \tag{1}$$

**Renyi's $\alpha$-divergence** $\alpha$-divergence is a rich family that includes many common divergences as special cases (Minka, 2001; Hernández-Lobato et al., 2016; Li and Turner, 2016). Here, we focus on Renyi's definition (Rényi et al., 1961; Li and Turner, 2016):

$$D_\alpha(p||q) = \frac{1}{\alpha - 1} \log \int p(\theta)^\alpha q(\theta)^{1-\alpha} d\theta, \quad \alpha > 0, \alpha \neq 1, \tag{2}$$

where $D_\alpha(p||q) \to \mathrm{KL}(p||q)$ when $\alpha \to 1$. Similar to maximizing Eq.(1), one can maximize the *variational Renyi bound* (VR bound) derived from Renyi's $\alpha$-divergence:

$$L_\alpha(q; \mathcal{D}) = \frac{1}{1 - \alpha} \log \mathbf{E}_{\theta \sim q} \left[ \left( \frac{p(\theta, \mathcal{D})}{q(\theta)} \right)^{1-\alpha} \right] = \log p(\mathcal{D}) - D_\alpha(q||p) \tag{3}$$

The reparameterization trick (Salimans et al., 2013; Kingma and Welling, 2013) is commonly used in practice for gradient ascent based optimization of the VR bound Eq.(3), where sampling $\theta \sim q_\phi(\theta)$ is conducted by first sampling $\epsilon \sim p(\epsilon)$ from a simple distribution independent with the variational parameter $\phi$ (e.g. Gaussian) then parameterizing $\theta = h_\phi(\epsilon)$. Using the reparameterization trick (Kingma and Welling, 2013) and Monte Carlo (MC) approximation, the gradient of VR bound w.r.t. $\phi$ with $K$ particles approximation is

$$\nabla_\phi L_\alpha(q_\phi; x) = \sum_{k=1}^{K} \left[ w_{\alpha,k} \nabla_\phi \log \frac{p(h_\phi(\epsilon_k), x)}{q(h_\phi(\epsilon_k))} \right] \quad \text{where } w_{\alpha,k} = \frac{\left( \frac{p(h_\phi(\epsilon_k), x)}{q(h_\phi(\epsilon_k))} \right)^{1-\alpha}}{\sum_{k=1}^{K} \left[ \left( \frac{p(h_\phi(\epsilon_k), x)}{q(h_\phi(\epsilon_k))} \right)^{1-\alpha} \right]} \tag{4}$$

**$f$-divergence** $f$-divergence defines a more general family of divergences (Csiszár et al., 2004; Minka et al., 2005). It can be defined using a twice differentiable convex function $f : \mathbb{R}_+ \to \mathbb{R}$ (Csiszár et al., 2004):

$$D_f(p||q) = \mathbf{E}_{\theta\sim q}\left[f\left(\frac{p(\theta)}{q(\theta)}\right) - f(1)\right]. \tag{5}$$

This family includes KL-divergence in both directions which can be seen by taking $f(t) = -\log t$ for $\mathrm{KL}(q||p)$ and $f(t) = t\log t$ for $\mathrm{KL}(p||q)$. It also contains $\alpha$-divergence which takes $f(t) = \frac{t^\alpha}{\alpha(\alpha-1)}$ for $\alpha \in \mathbb{R}\backslash\{0,1\}$. Although $f$-divergence family is very rich due to the usage of arbitrary twice differentiable convex function, it requires significant expertise to design a suitable $f$ function for a specific task.

## 3. Meta-VI

The goal of meta-learning a variational inference algorithm is to learn a divergence objective based on training tasks, so that the resulting VI algorithm produces an approximated distribution with desired properties on a certain type of tasks. To achieve this, we first construct a learnable divergence family, then design a meta-loss function that gives guidance for updating the divergence.

Assume we have $M$ training tasks $T_1, ..., T_M$ sampled from an underlying task distribution $p(\mathcal{T})$. Each task has its own probabilistic model $p_{T_i}(\theta_i, \mathcal{D}_{T_i})$. Let $D_\eta(\cdot||\cdot)$ be the learnable divergence parameterized by $\eta$, then for each task the approximated posterior $q_{\phi_i}(\theta_i)$ is computed by minimizing $D_\eta(p_{T_i}(\theta_i|\mathcal{D}_{T_i})||q_{\phi_i}(\theta_i))$. In the rest of the paper we write $D_\eta(q_{\phi_i}, T_i) = D_\eta(p_{T_i}(\theta_i|\mathcal{D}_{T_i})||q_{\phi_i}(\theta_i))$ for brevity. During meta-training, we define a meta-loss function $\mathcal{J}(q_{\phi_i}, T_i)$ which is optimized w.r.t. the divergence parameter $\eta$. This meta-loss function is designed to evaluate the desired properties of the approximated distribution, e.g. log-likelihood. During meta-testing, a new task is sampled from $p(\mathcal{T})$, and the learned divergence $D_\eta$ is used to optimize the variational distribution. The above meta-learning settings are practical as demonstrated in many previous work (Finn et al., 2017, 2018; Kim et al., 2018), including meta-learning for Bayesian inference (Gong et al., 2018). Attaining common knowledge based on the previous tasks has been proved to be useful for the future tasks.

We first present our method assuming the parameteric form of $D_\eta$ is given. Then we will provide the details of parameterization of two divergence families: $\alpha$-divergence and $f$-divergence and how they fit in this framework.

The idea of meta-learning divergences is that we first optimize the approximated posterior by minimizing the current divergence, then update the divergence using the feedback from the meta-loss. Formally speaking, for each task $T_i$ we perform $B$ gradient descent steps on the variational parameters $\phi_i$ using the current divergence $D_\eta$ as in the typical VI optimization:

$$\phi_i \leftarrow \phi_i - \beta\nabla_{\phi_i} D_\eta(q_{\phi_i}, T_i) \tag{6}$$

where $\beta$ is the learning rate. By doing so the updated variational parameters are a function of the divergence parameter $\eta$. Then we update the divergence parameter $\eta$ by one-step gradient descent using the meta-loss $\mathcal{J}$:

$$\eta \leftarrow \eta - \gamma\nabla_\eta \frac{1}{M}\sum_i \mathcal{J}(q_{\phi_i}, T_i) \tag{7}$$

where $\gamma$ is the learning rate. We call meta learning divergence objective *meta-D* and outline the algorithm in Algorithm 1 in the appendix.

**Meta-learning within $\alpha$-divergence family** The parameterization of Renyi's $\alpha$ divergence (2) is straightforward: $\eta = \alpha$. As the VR bound (3) is an equivalent optimization objective to Renyi's $\alpha$-divegrence, it means $\nabla_{\phi_i} D_\eta = -\nabla_{\phi_i} \mathcal{L}_\alpha$.

**Meta-learning within $f$-divergence family** We wish to parameterize the $f$-divergence (5) by parameterizing the convex function $f$ using a neural network. However, it is less straight-forward to specify the convexity constraint for neural networks . Fortunately, the $f$-divergence and its gradient can be specified through its second order derivative $f''$ without the original $f$ (Wang et al., 2018). Let $\theta = h_\phi(\epsilon)$ using reparameterization trick (Salimans et al., 2013; Kingma and Welling, 2013). Assume $\nabla_\theta \log\left(\frac{p(\theta)}{q_\phi(\theta)}\right)$ exists, then

$$\nabla_\phi D_f(p||q_\phi) = -\mathbf{E}_{\epsilon,\theta=h_\phi(\epsilon)}\left[g_f\left(\frac{p(h_\phi(\epsilon))}{q_\phi(h_\phi(\epsilon))}\right)\nabla_\phi h_\phi(\epsilon)\nabla_\theta \log\left(\frac{p(\theta)}{q_\phi(\theta)}\right)\right] \tag{8}$$

where $g_f(t) = f''(t)t^2$. This implies that we can define the gradient of $f$-divergence through $f''$. In addition, as shown in Wang et al. (2018), for any non-negative function $g$ on $\mathbb{R}_+$, there exists a function $f$ such that $g(t) = f''(t)t^2$. If $g_f(t)$ is strictly positive, i.e. $g_f(1) > 0$, then $D_f(p||q_\phi) = 0$ implies $p = q_\phi$. Given these guarantees, we propose to parameterize $f$ implicitly by parameterizing $g_f$ which can be any non-negative function. We turn the problem into using a neural network to express a non-negative function which is strictly positive at $t = 1$. We further restrict the form of the function to be $g_f(t) = \exp(r_\eta(t))$ where $r_\eta(t)$ is a neural network with parameter $\eta$. This definition of $g_f$ is strictly positive for all $t$. Then using Eq. (8), we compute the gradient $\nabla_{\phi_i} D_\eta = \nabla_{\phi_i} D_{f_\eta}$.

Besides the above setting, we also consider a few-shot learning set-up which learns a good initialization of variational parameters, similar to the model-agnostic meta-learning (MAML) framework (Finn et al., 2017, 2018; Kim et al., 2018). We present this setting in appendix A.

## 4. Experiments

We verify the proposed meta-VI approach (Algorithm 1) can learn a good divergence by considering a 1-d distribution approximation problem. More experimental results can be found in the appendix. Here, each task includes approximating a mixture of two Gaussians (see the appendix) by a Gaussian distribution which is attained by $\min_\phi D_\eta(p||q_\phi)$.

We test the meta-VI approach with two types of meta-loss: $D_{0.5}(q||p)$ ($\alpha$-divergence with $\alpha = 0.5$) and total variation (TV). If $D_{0.5}$ is the metric in use, then a good divergence will be $D_{0.5}$ itself. The goal of testing with meta-loss $D_{0.5}$ is to verify that our method is able to learn the preferred divergence given a rich enough family of candidate divergences $\{D_\eta\}$. As in this case the preferred divergence is known, we can directly evaluate the learned divergence by comparing it with the known preferred divergence. We use TV to evaluate the performance of our method when meta-loss is beyond the divergence family. BO (Snoek et al., 2012) is used to optimize $\alpha$ as a baseline. We learn the divergence on $M = 10$ tasks and set $B = 1$.

In Table 1, we report the learned value of $\alpha$ in Eq.(3) from meta-$\alpha$ and BO. When the meta-loss is $D_{0.5}$, the learned $\alpha$ from meta-$\alpha$ is very close to 0.5 which demonstrates that our method can essentially learn a good $\alpha$. BO is less computationally efficient, as it needs

Table 1: Learned value of $\alpha$ from meta-$\alpha$ and BO. BO with 8 iterations has similar running time as meta-$\alpha$.

| **Methods** | $\alpha = 0.5$ | TV |
|---|---|---|
| meta-$\alpha$ | 0.52±0.01 | 0.31±0.01 |
| BO (8 iters) | 0.81±0.03 | 0.69±0.08 |
| BO (16 iters) | 0.54±0.07 | 0.32±0.03 |

Table 2: Value of meta-loss over 10 test tasks.

| **Methods** | $\alpha = 0.5$ | TV |
|---|---|---|
| ground truth | 0.0811±0.0277 | - |
| meta-$\alpha$ | 0.0811±0.0277 | 0.0855±0.0149 |
| meta-$f$ | **0.0795**±0.0301 | **0.0806**±0.0163 |
| BO (8 iters) | 0.0833±0.0289 | 0.0879±0.0143 |
| BO (16 iters) | 0.0811±0.0277 | 0.0855±0.0149 |

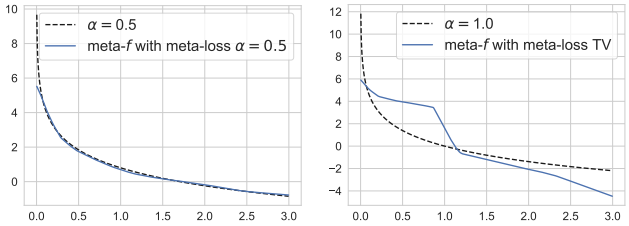



Figure 2: Visualizing the learned $\log f''$.

Table 3: Rank of meta-loss over 10 test tasks.

| **Methods** | $\alpha = 0.5$ | TV |
|---|---|---|
| meta-$\alpha$ | **2.10**±0.70 | 2.10±0.30 |
| meta-$f$ | **2.10**±1.37 | **1.00**±0.00 |
| BO (8 iters) | 3.50±0.67 | 4.00±0.00 |
| BO (16 iters) | 2.30±0.90 | 2.90±0.30 |

to train a model from scratch every single time when evaluating a new value of $\alpha$, while our method can update $\alpha$ based on the current model. We also consider learning $f$-divergence and visualize in Figure 2 the learned $\log f''$. When $D_{0.5}$ is in use as the meta-loss, the corresponding $\log f^{*''}$ for $D_{0.5}$ is analytical (see the appendix), and we see from Figure 2 (a) that the learned $\log f''$ and $\log f^{*''} + 0.8$ are almost identical. This means meta-VI has learned the optimal divergence $D_{0.5}$ ($f(t)$ and $e^{0.8} \times f(t)$ define the same divergence).

In the case of using TV as the meta-loss, the optimal divergence is not analytic. Therefore, we instead report in Table 2 the meta-losses on 10 test tasks, which are obtained by executing the learned divergence minimization algorithm for 2000 iterations. The error bar is large due to the large variance among different tasks, so we also report the ranking in Table 3. It clearly shows that meta-$\alpha$ and meta-$f$ are superior over BO. Moreover, meta-$f$ outperforms meta-$\alpha$ when the meta-loss is TV. From Figure 2 (b), we can see that the learned $f$-divergence is not inside $\alpha$-divergence, showing the benefit of using a larger divergence family.

## References

Robert Bamler, Cheng Zhang, Manfred Opper, and Stephan Mandt. Perturbative black box variational inference. In *Advances in Neural Information Processing Systems*, pages 5079–5088, 2017.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer Science+ Business Media, 2006.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Imre Csiszár, Paul C Shields, et al. Information theory and statistics: A tutorial. *Foundations and Trends® in Communications and Information Theory*, 1(4):417–528, 2004.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, pages 9516–9527, 2018.

Gustavo L Gilardoni. On pinsker's and vajda's type inequalities for csiszár's $f$-divergences. *IEEE Transactions on Information Theory*, 56(11):5377–5386, 2010.

Wenbo Gong, Yingzhen Li, and José Miguel Hernández-Lobato. Meta-learning for stochastic gradient mcmc. *arXiv preprint arXiv:1806.04522*, 2018.

F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):19, 2016.

José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Daniel Hernández-Lobato, Thang Bui, and Richard Turner. Black-box $\alpha$-divergence minimization. 2016.

Sepp Hochreiter, A Steven Younger, and Peter R Conwell. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks*, pages 87–94. Springer, 2001.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Taesup Kim, Jaesik Yoon, Ousmane Dia, Sungwoong Kim, Yoshua Bengio, and Sungjin Ahn. Bayesian model-agnostic meta-learning. *arXiv preprint arXiv:1806.03836*, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Yingzhen Li and Richard E Turner. Rényi divergence variational inference. In *Advances in Neural Information Processing Systems*, pages 1073–1081, 2016.

Chao Ma, Wenbo Gong, José Miguel Hernández-Lobato, Noam Koenigstein, Sebastian Nowozin, and Cheng Zhang. Partial vae for hybrid recommender system. In *NIPS Workshop on Bayesian Deep Learning*, 2018a.

Chao Ma, Sebastian Tschiatschek, Konstantina Palla, Jose Miguel Hernandez Lobato, Sebastian Nowozin, and Cheng Zhang. Eddi: Efficient dynamic discovery of high-value information with partial vae. *arXiv preprint arXiv:1809.11142*, 2018b.

Thomas P Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.

Tom Minka et al. Divergence measures and message passing. Technical report, Technical report, Microsoft Research, 2005.

Devang K Naik and RJ Mammone. Meta-neural networks that learn by learning. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 1, pages 437–442. IEEE, 1992.

Alfréd Rényi et al. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California, 1961.

Tim Salimans, David A Knowles, et al. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8(4):837–882, 2013.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

Dilin Wang, Hao Liu, and Qiang Liu. Variational inference with tail-adaptive f-divergence. In *Advances in Neural Information Processing Systems*, pages 5737–5747, 2018.

Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

## Appendix A. Meta-Learning Divergence Objective and Variational Parameters

**Algorithm 1** Meta-$D$

**Input:** $M$: number of training tasks. $\beta$, $\gamma$: learning rate hyperparameters. Initialize $\eta$, $\phi_i$, $i = 1, \dots, M$ ($\phi_i$ can have different structures).
**loop**
  **for** $T_i$, $i = 1, \dots, M$ **do**
    **for** $B$ times **do**
      Update the variational parameters with the current divergence: $\phi_i \leftarrow \phi_i - \beta \nabla_{\phi_i} D_\eta(q_{\phi_i}, T_i)$.
    **end for**
  **end for**
  Update $\eta \leftarrow \eta - \gamma \nabla_\eta \frac{1}{M} \sum_i \mathcal{J}(q_{\phi_i}, T_i)$
**end loop**
**Output:** $\eta$

**Algorithm 2** Meta-$D\&\phi$

**Input:** $p(\mathcal{T})$: distribution over tasks. $\beta$, $\gamma$, $\tau$: learning rate hyperparameters. Initialize $\phi$, $\eta$
**loop**
  Sample $M$ tasks $T_i \sim p(\mathcal{T})$.
  **for all** $T_i$ **do**
    Update the variational parameters with the current divergence: $\phi_i \leftarrow \phi - \beta \nabla_\phi D_\eta(q_\phi, T_i)$.
  **end for**
  Update $\phi \leftarrow \phi - \tau \nabla_\phi \frac{1}{M} \sum_i \mathcal{J}(q_{\phi_i}, T_i)$; $\eta \leftarrow \eta - \gamma \nabla_\eta \frac{1}{M} \sum_i \mathcal{J}(q_{\phi_i}, T_i)$
**end loop**
**Output:** $\eta$, $\phi$

In addition to learning the divergence objective, we also consider the setting where fast adaptation of the variational parameters to new tasks is desirable. Similar to MAML, the probabilistic models $\{p_{T_i}(\theta_i, \mathcal{D}_{T_i})\}$ share the same architecture, and the goal is to learn an initialization of variational parameters $\phi_i \leftarrow \phi$. On a specific task, $\phi$ is adapted to be $\phi_i$ according to the learnable divergence $D_\eta$:

$$\phi_i \leftarrow \phi - \beta \nabla_\phi D_\eta(q_\phi, T_i). \tag{9}$$

Again the updated $\phi_i$ is a function of both $\eta$ and $\phi$. Here we simply assume the number of gradient steps to be $B = 1$, and it is straightforward to extend the method to $B > 1$. For meta-update, besides updating divergence parameters $\eta$ with Eq.(7), we also use the same meta-loss to update $\phi$:

$$\phi \leftarrow \phi - \tau \nabla_\phi \frac{1}{M} \sum_i \mathcal{J}(q_{\phi_i}, T_i). \tag{10}$$

We call meta-VI with learning both the divergence objective and variational parameters' initialization *meta-D&$\phi$* and summarize the algorithm in Algorithm 2. Similar to the previous section, the divergence families in consideration are $\alpha$-divergence and $f$-divergence.

## Appendix B. Computing Equation (8) in Practice

With dataset $\mathcal{D}$, the density ratio in f-divergence becomes $\frac{p(\theta|D)}{q_\phi(\theta)} = \frac{p(D|\theta)p(\theta)}{q_\phi(\theta)p(D)}$. We estimate $p(D)$ through importance sampling and MC approximation: $p(D) = E_{\theta \sim p(\theta)}[p(D|\theta)] = E_{\theta \sim q_\phi(\theta)}[\frac{p(D|\theta)p(\theta)}{q_\phi(\theta)}] \approx \frac{1}{K} \sum_k^K \frac{p(D|\theta_k)p(\theta_k)}{q_\phi(\theta_k)}$ where $\theta_k \sim q_\phi(\theta)$. After doing this, the density

ratio becomes $\frac{p(\theta_k|D)}{q_\phi(\theta_k)} = \frac{p(D|\theta_k)p(\theta_k)}{q_\phi(\theta_k)} \Big/ \frac{1}{K}\sum_k^K \frac{p(D|\theta_k)p(\theta_k)}{q_\phi(\theta_k)}$ which can be regarded as a self-normalized estimator, similar to the normalization importance weight in Li and Turner (2016). A self-normalized estimator generally helps stabilize the training especially at the beginning. We use this estimator for regression tasks and recommender system.

## Appendix C. Additional Experimental Results and Setting Details

### C.1. Model Architecture for $f$-divergence

On all experiments, we parameterize $g(t)$ in $f$-divergence by a neural network with 2 hidden layers with 100 hidden units and RELU nonlinearilities.

### C.2. Approximate Mixture of Gaussians

The mixture of Gaussian distribution $p(\theta) = 0.5\mathcal{N}(\theta; \mu_1, \sigma_1^2) + 0.5\mathcal{N}(\theta; \mu_2, \sigma_2^2)$ is generated by

$$\mu_1 \sim \text{Unif}[0, 3] \quad \sigma_1 \sim \text{Unif}[0.5, 1.0]; \qquad \mu_2 = \mu_1 + 3 \quad \sigma_2 = \sigma_1 * 2.$$

Therefore each task has a different target distribution but with similar properties (i.e. the distance between two modes is the same and the standard deviation of the second mode is 2 times larger than that of the first mode). The choice of the divergence affects the properties of the approximated Gaussian distribution as shown in Figure 1.

Here we test meta-learning both the divergence objective and the variational parameters (Algorithm 2). We use Algorithm 2 without updating divergence as a baseline, denoting by VB&$\phi$. During training, we sample 10 tasks each time and perform $B = 20$ inner loop gradient updates. The learned $\alpha$ is different from Table 1 (see Table 4). We conjecture that this is related to the learned $\phi$ and the horizon length. During meta-testing, we use the learned $\phi$ for variational parameter initialization, and train the variational parameters with the learned divergence for 20 and 100 iterations respectively to evaluate the effect of the learned divergence in short and long horizon. We summarize the meta-loss in Table 5 and the ranking in Table 6. Our methods are not only better than VB&$\phi$ after 20 updates but also better after 100 updates. This demonstrates the benefit of learning a divergence for the tasks instead of the conventional VB. To further explore the reason of getting lower meta-loss of meta-$D$&$\phi$, we visualize the approximated distribution of all methods after 20 steps in Figure 3. The approximated distributions obtained by meta-$D$&$\phi$ tend to fit the mixture of Gaussians more globally (mass-covering) than VB&$\phi$. This mass-covering behaviour results in better meta-loss. Compared to learning divergence only, learning variational parameter initialization helps shorten the training time on new tasks (100 iterations v.s. 2000 iterations). Notably, meta-VI is able to provide this initialization along with divergence learning without extra cost.

Table 4: Meta-$D$&$\phi$ on MoG: learned value of $\alpha$.

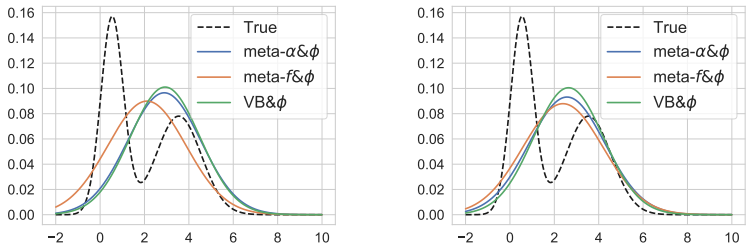
| **Methods** | $\alpha = 0.5$ | TV |
|---|---|---|
| meta-$\alpha$&$\phi$ | 0.88 | 0.77 |

Table 5: Meta-$D\&\phi$ on MoG: value of meta-loss over 10 test tasks.

| **Methods\Meta-loss** | $\alpha = 0.5$ (20 iters) | TV (20 iters) | $\alpha = 0.5$ (100 iters) | TV (100 iters) |
|---|---|---|---|---|
| meta-$\alpha\&\phi$ | 0.1207±0.0500 | 0.0982±0.0166 | 0.0879±0.0305 | **0.0903**±0.0149 |
| meta-$f\&\phi$ | **0.0793**±0.0237 | **0.0935**±0.0152 | **0.0784**±0.0332 | 0.0918±0.0151 |
| VB$\&\phi$ | 0.1237±0.0539 | 0.1026±0.0181 | 0.0905±0.0332 | 0.0926±0.0153 |

Table 6: Meta-$D\&\phi$ on MoG: rank of meta-loss over 10 test tasks.

| **Methods\Meta-loss** | $\alpha = 0.5$ (20 iters) | TV (20 iters) | $\alpha = 0.5$ (100 iters) | TV (100 iters) |
|---|---|---|---|---|
| meta-$\alpha\&\phi$ | 2.10±0.54 | 1.80±0.60 | 2.20±0.75 | **1.40**±0.66 |
| meta-$f\&\phi$ | **1.20**±0.60 | **1.50**±0.81 | **1.40**±0.80 | 2.10±0.83 |
| VB-MAML | 2.70±0.46 | 2.70±0.46 | 2.40±0.49 | 2.50±0.50 |




(a) Meta-loss: $\alpha = 0.5$ (b) Meta-loss: TV

Figure 3: Meta-$D\&\phi$ on MoG: visualization of approximated distribution after 20 updates.

### C.2.1. SETTING DETAILS

TV is defined as

$$TV(p, q) = \sup_x |p(x) - q(x)| = \frac{1}{2} \int |p(x) - q(x)| dx.$$

For $\alpha \in (0, 1]$, TV is related to $\alpha$-divergence by $\frac{\alpha}{2} TV^2 \leq D_\alpha(p||q)$ (Gilardoni, 2010).

We set the search region for BO to be $\alpha \in [0, 3]$ which includes many common divergence such as KL, Helinger distance ($\alpha = 0.5$) and $\chi^2$-divergence ($\alpha = 2$). We note that BO is not applicable when the divergence set is $f$-divergence which is parameterized by a neural network.

The expectation in Eq.(3) and (8) is computed by MC approximation with 1000 particles. Note that $p(\theta)$ is computable, since we know the parameters of $p$.

Bayesian optimization is implemented through a public package.[1] The acquisition function is the upper confidence bound with kappa 0.1. We used the same data of the training tasks for BO. Specifically, the objective function that BO wants to minimize is the meta-loss ($D_{0.5}$ or TV). Every time BO selects an $\alpha$, we train 10 models with that $\alpha$-divergence on the support sets of 10 training tasks respectively and get the mean of log-likelihood on the query sets of the 10 training tasks. Each time the model is trained for 2000 iterations.

When $D_{0.5}$ is in use as the meta-loss, ideally the learned f-divergence should be close to $D_{0.5}$. When the f-divergence is $D_{0.5}$, the $f$ function is $f(t) = \frac{t^{0.5}}{-0.5^2}$, and the analytical

1. https://github.com/fmfn/BayesianOptimization

Table 7: Meta-$D$ on regression: results are over 10 test tasks (1000 epochs).

| | Test LL | RMSE |
|---|---|---|
| VB | -0.6377±0.0433 | 0.4522±0.0196 |
| meta-$\alpha$ | -0.4596±0.0857 | 0.4500±0.0236 |
| meta-$f$ | **-0.4390**±0.1084 | 0.4599±0.0200 |

Table 8: Meta-$D\&\phi$ on regression: results are over 10 test tasks (500 epochs).

| | Test LL | RMSE |
|---|---|---|
| VB$\&\phi$ | -0.6354±0.0599 | 0.4556±0.0247 |
| meta-$\alpha\&\phi$ | -0.4967±0.0647 | 0.4562±0.0207 |
| meta-$f\&\phi$ | **-0.4852**±0.0853 | 0.4552±0.0217 |

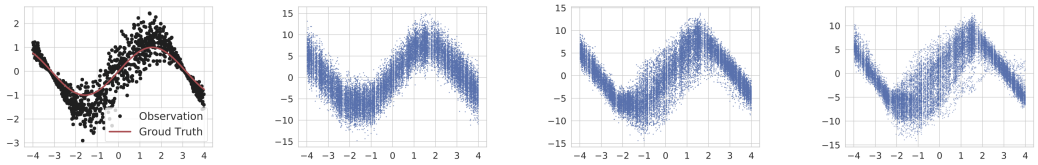
expression of $\log f''(t)$ is $-1.5\log t + C$ with $C$ reflecting the scaling constant in $f$. In Figure 2, we compare the learned $\log f''(t)$ and the ground truth $-1.5\log t + C$. We found that the learned $\log f''(t)$ is very close to $-1.5\log t + 0.8$. This means that our method has learned the optimal divergence $D_{0.5}$ (because the definition of $f$-divergence is invariant to constant scaling of the function $f$, i.e. $f$ and $e^{0.8} \times f$ define the same divergence).

### C.3. Regression Tasks with Bayesian Neural Networks

The second test considers Bayesian neural network regression. The distribution of ground truth regression function is defined by a (which is a function of $x$, see Figure 4 (a)): $y = A\sin(x+b) + A/2|\cos((x+b)/2)|\epsilon$, where the amplitude $A \in [5, 10]$, the phase $b \in [0, 1]$ and $\epsilon \sim \mathcal{N}(0, 1)$. The heteroskedastic noise makes the uncertainty estimate crucial when compared with the sinusoid function fitting task in Finn et al. (2017); Kim et al. (2018). The model is a two-layer neural network with hidden layer size 20 and RELU nonlinearities. We use marginal log-likelihood as the meta-loss.

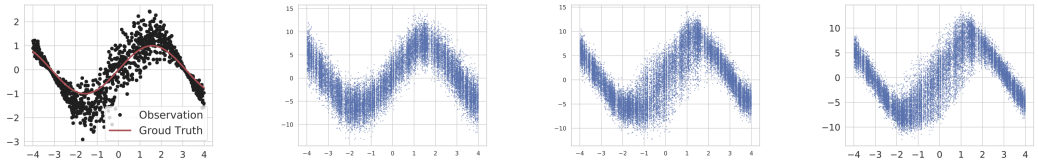
For meta-learning divergence only, the training set size is 1000 and is obtained by sampling $x \in [-4, 4]$ uniformly. We use $M = 20$, $B = 1$, $K = 50$ and batch size 40 of which 20 data points (the support set) are for updating $\phi_i$ and 20 points (the query set) are for updating $\eta$. We train meta-$D$ for 1500 epochs. To evaluate the performance, we train the model with the learned divergence and VB respectively on new tasks for 1000 epochs. The quantitative results are summarized in Table 7. We can see that the test log-likelihood of both meta-$\alpha$ and meta-$f$ are significantly better than VB and the root mean square error (RMSE) are similar for all methods. We visualize the predictive distribution on an example sinusoid function in Figure 4. All methods fit the mean well which is consistent with the RMSE results. However, VB fails to capture the heteroskedastic uncertainty and instead uses homoskedastic noise to fit the data. On the other hand, meta-$\alpha$ and meta-$f$ can reason about the heteroskedastic noise. This explains the results of better test log-likelihood.

For learning both divergence and variational parameters initialization, we sample 20 tasks where each task has 40 data points. We use 20 points for $\phi_i$ and the other 20 points for updating divergence $\eta$ and the shared initialization $\phi$. We set $B = 1$. To evaluate, we start with the learned initialization and train the variational parameters with the learned divergence for 500 epochs. Similar to the results of learning only the divergence objective, meta-$\alpha\&\phi$ and meta-$f\&\phi$ are able to model heteroskedastic predictive distribution while VB$\&\phi$ cannot. The quantitative evaluation are given in Table 8 and an example of predictive distribution is given in Figure 5. Meta-$D\&\phi$ converges faster than meta-$D$, indicating that learning model initialization can shorten the training time on new tasks. We report the learned value of $\alpha$ in Eq.(3) from meta-$\alpha$ and meta-$\alpha\&\phi$ in Table 9.




(a) Ground Truth (b) VB (c) meta-$\alpha$ (d) meta-$f$

Figure 4: Meta-$D$ on regression: the predictive distribution on a sinusoid wave.




(a) Ground Truth (b) VB&$\phi$ (c) meta-$\alpha$&$\phi$ (d) meta-$f$&$\phi$

Figure 5: Meta-$D$&$\phi$ on regression: Predictive distribution on a sinusoid wave.

Table 9: Learned value of $\alpha$ of meta-$\alpha$ and meta-$\alpha$&$\phi$ on regression.

| | meta-$\alpha$ | meta-$\alpha$&$\phi$ |
|---|---|---|
| $\alpha$ | 0.1666 | 0.1020 |

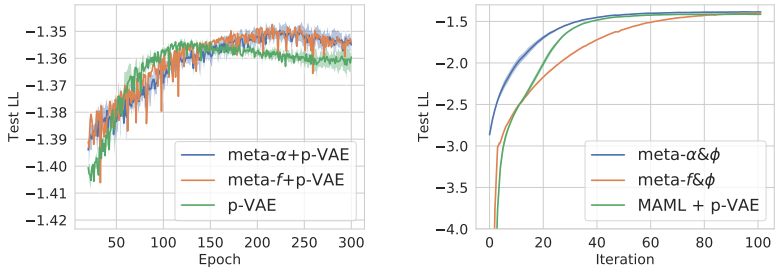
Table 10: Learned value of $\alpha$ of meta-$\alpha$ and meta-$\alpha$&$\phi$ on MovieLens.

| | meta-$\alpha$ | meta-$\alpha$&$\phi$ |
|---|---|---|
| $\alpha$ | 0.9029 | 1.0602 |

### C.4. Recommender System with Partial Variational Auto-encoders

We test our method on recommender systems with Partial Variational Auto-encoders (p-VAEs). P-VAE is a recently proposed model to deal with partially observed data and has been used to do user rating prediction in recommender system (Ma et al., 2018b,a). Similar to vanilla VAE (Kingma and Welling, 2013), p-VAE uses the KL-divergence as the variational objective. We apply our proposed method to the divergence objective in p-VAE.

We consider MovieLens 1M dataset (Harper and Konstan, 2016) which contains 1,000,206 ratings of 3,952 movies from 6,040 users. We split the users into seven age groups: under




(a) Meta-$D$ (b) Meta-$D$&$\phi$

Figure 6: Test log-likelihood of meta-VI on MovieLens. (b) The final results of meta-$\alpha$&$\phi$, meta-$f$&$\phi$ and MAML+p-VAE are -1.3855, -1.3985 and -1.4140 respectively.

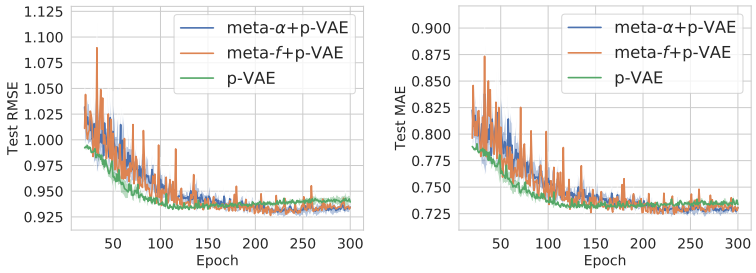



Figure 7: Meta-$D$ on MovieLens: Comparison of meta-$D$ and p-VAE in terms of test RMSE and test MAE.

18, 18-24, 25-34, 35-44, 45-49, 50-55 and above 56, and regard predicting the ratings of users within the same age group as a task since the users with similar age may have similar preferences. We select four as training tasks (under 18, 25-34, 45-49, above 56) and use the remaining as test tasks. We use marginal log-likelihood as the meta-loss.

For the setting of learning divergence only, during meta-training, we sample 100 users per task (400 users in total) and use half of the observed ratings to compute Eq.(6) and the other half for computing the meta-loss. The number of training epoch is 400. During meta-testing, we use 90%/10% training-test split for the three test tasks and train p-VAE with the learned divergence. The baseline p-VAE is directly trained on test tasks with KL-divergence. From Figure 6 (a), we can see that the combination of meta-$D$ and p-VAE outperforms vanilla p-VAE in terms of test log-likelihood, showing that meta-$D$ has learn a suitable divergence that leads to better test performance.

For learning both divergence and variational parameters, the setup of training is the same as learning divergence only except that now we also perform updates in Eq.(10). We compare our method with getting a p-VAE model initialization only (obtained by Algorithm 2 without updating $\eta$). This can be regarded as a combination of MAML and p-VAE. During evaluation, we apply 60%/40% training-test split for the test tasks and train the learned p-VAE model with the learned divergence. Figure 6 (b) implies that all methods can converge quickly on the new task with only 100 iterations. Both meta-$\alpha\&\phi$ and meta-$f\&\phi$ are better than p-VAE at the beginning, indicating that the learned divergence can help fast adaptation. Besides, meta-$\alpha\&\phi$ and meta-$f\&\phi$ also converge better than p-VAE in the end. This shows the learned divergence helps in both short and long horizon.

Again we provide the value of learned $\alpha$ in Eq.(3) from meta-$\alpha$ and meta-$\alpha\&\phi$ in Table 10. Besides the test log-likelihood, there are other popular evaluation metrics being used in recommender system and sometimes they are not consistent with each other. Therefore, we also evaluate the performance of our method in terms of other common metrics: test root mean square error (RMSE) and test mean absolute error (MAE). For both metrics, our methods converge better than the baseline in the setting of learning inference algorithm and the setting of learning inference algorithm and model parameters (see Figures 7 and 8).

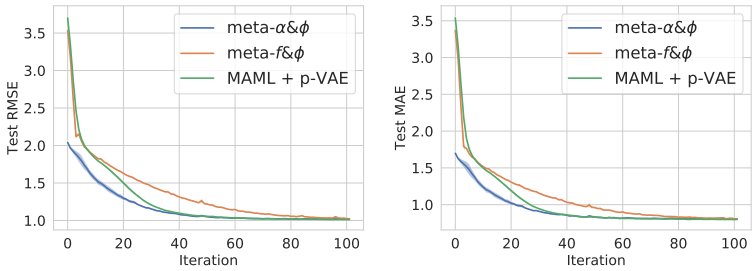



Figure 8: Meta-$D$&$\phi$ on MovieLens: Comparison of meta-$D$&$\phi$ and MAML+p-VAE in terms of test RMSE and test MAE.