Kernel Implicit Variational Inference

Jiaxin Shi*, Shengyang Sun*, Jun Zhu

Tsinghua University

Table of contents

1. Introduction
2. Background
3. Kernel Implicit Variational Inference
4. Example: Implicit Variational Bayesian Neural Networks
5. Experiments
6. Conclusion

Introduction

Probabilistic Machine Learning

Why modeling **uncertainty** is important?

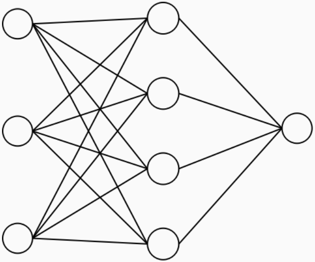
- **Model the data distribution.**
  - Data is uncertain in nature.
- **Calibrate confidence of models.**
  - They should know when they don't know.
- **Smooth predictions to prevent overfitting.**
  - Ground truths are usually smooth.

Bayesian Methods

Bayesian Inference

A mathematically grounded approach to solve for uncertainty.

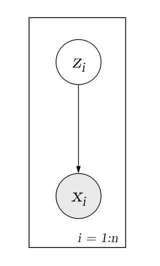
Example: Bayesian Neural Networks



$$\begin{aligned}\mathbf{W} &\sim N(\mathbf{0}, \mathbf{I}),\\ \hat{y} &= f_{\mathrm{NN}}(\mathbf{x}, \mathbf{W}),\\ y &\sim \mathcal{P}(\hat{y}; \theta).\end{aligned}$$

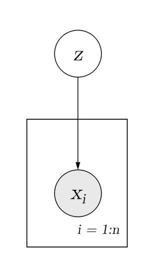
Background

Latent Variable Models (LVM)



(a) Local LVMs

$$p(\mathbf{x}_{1:N}, \mathbf{z}_{1:N}) = \prod_{i=1}^{N} [p(\mathbf{x}_i|\mathbf{z}_i)p(\mathbf{z}_i)]$$



(b) Global LVMs

$$p(\mathbf{x}_{1:N}, \mathbf{z}) = p(\mathbf{z}) \prod_{i=1}^{N} p(\mathbf{x}_i|\mathbf{z})$$

Variational Inference (VI)

Consider a generative model $p(\mathbf{z}, \mathbf{x}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z})$:

- $\mathbf{x}$: observed variables, $\mathbf{z}$: latent variables
- A variational distribution: $q_\phi(\mathbf{z})$ is chosen to approximate $p_\phi(\mathbf{z}|\mathbf{x})$

Objective: **E**vidence **L**ower **BO**und (ELBO)

$$\mathcal{L}(\mathbf{x}; \phi) = \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}(q_\phi(\mathbf{z}) \| p(\mathbf{z})).$$

Traditional Variational Inference

Approximation: Use a factorized variational family

$$q_\phi(\mathbf{z}) = \prod_{k=1}^{d} q_{\phi_k}(\mathbf{z}_k),$$

where $\mathbf{z} \in \mathbb{R}^d$.

Mean Field Variational Inference:

$$\mathcal{L}(\mathbf{x};\phi) = \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}(q_\phi(\mathbf{z}) \| p(\mathbf{z})),$$

$$\nabla_q \mathcal{L} = 0 \quad \Rightarrow \quad q_{\phi_k}(\mathbf{z}_k) \propto e^{\mathbb{E}_{q(\mathbf{z}_{\neg k})}[\log p(\mathbf{x},\mathbf{z})]}.$$

- Analytical, coordinate updates.
- Requires a **closed-form** solution for each update.

Modern Variational Inference

Stochastic [5, 9]: Sample a mini-batch of data $\mathbf{x}_{1:M}$ from the full dataset $\mathbf{x}_{1:N}$.

- Global LVMs:

$$\log p(\mathbf{x}_{1:N}|\mathbf{z}) \simeq \frac{N}{M}\sum_{i=1}^{M}\log p(\mathbf{x}_i|\mathbf{z}).$$

- Local LVMs:

$$\log p(\mathbf{x}_{1:N}) \simeq \frac{N}{M}\sum_{i=1}^{M}\log p(\mathbf{x}_i).$$

Modern Variational Inference

Differentiable [17]:

$$\mathcal{L}(\mathbf{x};\phi) = \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}(q_\phi(\mathbf{z})\|p(\mathbf{z})) \leq \log p_\theta(\mathbf{x})$$

- Update variational parameters ϕ:

$$\phi_{t+1} = \phi_t + \alpha\nabla_\phi\mathcal{L}$$

- Learning model parameters θ:

$$\theta_{t+1} = \theta_t + \alpha\nabla_\theta\mathcal{L}$$

- Many gradient estimators have been developed for low-variance updates of ϕ: **SGVB** (the reparameterization trick) [9], **REINFORCE** [14], **VIMCO** [15], **REBAR** [25], **RELAX** [1], . . .

Modern Variational Inference

Amortized: For local LVMs, instead of fitting a variational posterior for each local variable $\mathbf{z}_i, i = 1, \ldots, N$, choose a conditional variational family $q_\phi(\mathbf{z}|\mathbf{x})$ to amortize all the local inference problems:

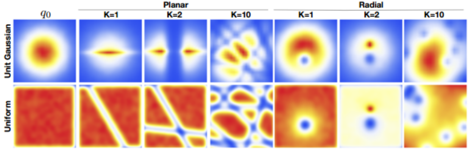
$$\mathcal{L}(\mathbf{x}_i; \phi) = \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathbf{x}_i)}\left[\log p_\theta(\mathbf{x}_i|\mathbf{z}_i)\right] - \mathrm{KL}(q_\phi(\mathbf{z}_i|\mathbf{x}_i) \| p(\mathbf{z}_i)) \leq \log p_\theta(\mathbf{x}_i)$$

Recent Attempts Towards Expressive Posteriors

Matrix Gaussian [12, 23]

$$p(\mathbf{X} \mid \mathbf{M}, \mathbf{U}, \mathbf{V}) = \frac{\exp\left(-\frac{1}{2}\operatorname{tr}\left[\mathbf{V}^{-1}(\mathbf{X}-\mathbf{M})^T\mathbf{U}^{-1}(\mathbf{X}-\mathbf{M})\right]\right)}{(2\pi)^{np/2}|\mathbf{V}|^{n/2}|\mathbf{U}|^{p/2}}$$

Normalizing flow [8, 19]



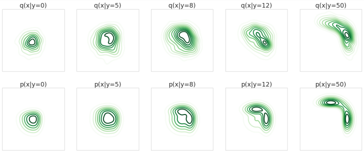
$$\mathbf{z}_t = f(\mathbf{z}_{t-1})$$

$$q(\mathbf{z}_t) = q(\mathbf{z}_{t-1})\left|\det\frac{\partial f(\mathbf{z}_{t-1})}{\partial \mathbf{z}_{t-1}}\right|^{-1}$$

Recent Attempts Towards Expressive Posteriors

Implicit distributions [6, 13]

Variational families that can be constructed by using general deterministic or stochastic transformations, which is not necessarily invertible.



- Known sampling process
- No tractable likelihood

This kind of distribution is called *implicit distributions*.

Related works include prior-contrastive (PC) [6] for global LVMs, and Adversarial Variational Bayes (AVB) [13] for local LVMs.

Implicit Variational Inference

Implicit VI

For variational methods that use an implicit variational posterior (also known as variational programs [18], wild variational approximations [10]), we refer to them as *Implicit Variational Inference* (implicit VI)

Challenge

$$\mathcal{L}(\mathbf{x};\phi) = \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}(q_\phi(\mathbf{z})\|p(\mathbf{z})).$$

Computing $\mathrm{KL}(q_\phi(\mathbf{z})\|p(\mathbf{z}))$ requires to evaluate the density of q_ϕ, which is intractable for an implicit distribution.

Implicit VI: Prior-Contrastive Methods

Recently works inspired by the probabilistic interpretation of GAN [4, 16] has extended the adversarial game approach to variational inference [13, 6, 24].

Key idea

$$\mathrm{KL}(q\|p) = \mathbb{E}_q \log \frac{q_\phi(\mathbf{z})}{p(\mathbf{z})}$$

$\frac{q_\phi(\mathbf{z})}{p(\mathbf{z})}$ can be estimated from samples of the two distributions by using a probabilistic classifier.

$$\max_D \mathbb{E}_{q_\phi(\mathbf{z})} \left[\log\left(D(\mathbf{z})\right)\right] + \mathbb{E}_{p(\mathbf{z})} \left[\log\left(1 - D(\mathbf{z})\right)\right].$$

The optimal solution of problem is $D(\mathbf{z}) = q_\phi(\mathbf{z})/(q_\phi(\mathbf{z}) + p(\mathbf{z}))$. Therefore, the KL term can be approximated as

$$\mathrm{KL}(q_\phi\|p) \approx \mathbb{E}_{q_\phi(\mathbf{z})} \left[\log D(\mathbf{z}) - \log(1 - D(\mathbf{z}))\right].$$

Implicit VI: Prior-Contrastive Methods

This is called **prior-contrastive** (PC) forms of VI in [6]. Its amortized version has been independently developed as **Adversarial Variational Bayes** (AVB) [13].

Problems of discriminator-based approaches

- noisy training due to truncation of inner loop.

Implicit VI: Prior-Contrastive Methods

This is called **prior-contrastive** (PC) forms of VI in [6]. Its amortized version has been independently developed as **Adversarial Variational Bayes** (AVB) [13].

Problems of discriminator-based approaches

- noisy training due to truncation of inner loop.
- Estimation is of high variance due to overfitting of the strong discriminator.

Implicit VI: Prior-Contrastive Methods

This is called **prior-contrastive** (PC) forms of VI in [6]. Its amortized version has been independently developed as **Adversarial Variational Bayes** (AVB) [13].

Problems of discriminator-based approaches

- noisy training due to truncation of inner loop.
- Estimation is of high variance due to overfitting of the strong discriminator.
- Cannot scale towards very high-dimensional latent variables, e.g., weights in a moderate-size neural network.

Kernel Implicit Variational Inference

Kernel Implicit Variational Inference

Kernel Implicit Variational Inference (KIVI)

A new implicit VI method that utilizes kernel regression in the latent space to estimate the gradients of the ELBO with an implicit posterior.

Features

- No noisy gradients: closed-form, globally optimal estimate. No adversarial games.

Kernel Implicit Variational Inference

Kernel Implicit Variational Inference (KIVI)

A new implicit VI method that utilizes kernel regression in the latent space to estimate the gradients of the ELBO with an implicit posterior.

Features

- No noisy gradients: closed-form, globally optimal estimate. No adversarial games.
- Principled control of bias/variance tradeoff.

Kernel Implicit Variational Inference

Kernel Implicit Variational Inference (KIVI)

A new implicit VI method that utilizes kernel regression in the latent space to estimate the gradients of the ELBO with an implicit posterior.

Features

- No noisy gradients: closed-form, globally optimal estimate. No adversarial games.
- Principled control of bias/variance tradeoff.
- Scale to high-dimensional latent-variable models.

Kernel Implicit Variational Inference

Kernel Implicit Variational Inference (KIVI)

A new implicit VI method that utilizes kernel regression in the latent space to estimate the gradients of the ELBO with an implicit posterior.

Features

- No noisy gradients: closed-form, globally optimal estimate. No adversarial games.
- Principled control of bias/variance tradeoff.
- Scale to high-dimensional latent-variable models.
- Applicable to both local and global LVMs.

Kernel Implicit Variational Inference

Estimating the KL-term

$$\mathcal{L}(\mathbf{x};\phi) = \mathbb{E}_{q_\phi(\mathbf{z})}\left[\log p(\mathbf{x}|\mathbf{z})\right] - \mathrm{KL}(q_\phi(\mathbf{z})\|p(\mathbf{z}))$$

Let $\mathbf{z} \in \mathbb{R}^d$ be the latent variable, and the true density ratio is

$$r(\mathbf{z}) = \frac{q(\mathbf{z})}{p(\mathbf{z})}.$$

Consider modeling it with a function $\hat{r} \in \mathcal{H}$, where $\mathcal{H}$ is a *Reproducing Kernel Hilbert Space* (RKHS) induced by a positive definite kernel $k(\mathbf{z},\mathbf{z}') : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$.

Objective composed of **1)** a square loss for regression plus **2)** a penalty for the complexity of the function (measured by the RKHS norm $\|\hat{r}\|^2_{\mathcal{H}}$):

$$\min_{\hat{r}\in\mathcal{H}} \; \mathcal{L}(\hat{r}) + \frac{\lambda}{2}\|\hat{r}\|^2_{\mathcal{H}}.$$

Here λ is the regularization coefficient.

KIVI: Estimating the KL-term

Objective $\min_{\hat{r}\in\mathcal{H}} \mathcal{L}(\hat{r}) + \frac{\lambda}{2}\|\hat{r}\|^2_{\mathcal{H}}.$

Squared Loss

- For the squared loss we choose the form used by the unconstrained Least Square Importance Fitting (uLSIF) [7]:

$$\mathcal{J}(\hat{r}) = \frac{1}{2}\int(\hat{r}(\mathbf{z}) - r(\mathbf{z}))^2 p(\mathbf{z})\, d\mathbf{z} = \frac{1}{2}\,\mathbb{E}_p\hat{r}(\mathbf{z})^2 - \mathbb{E}_q\hat{r}(\mathbf{z}) + C,$$

where C is a constant.

KIVI: Estimating the KL-term

Objective $\min_{\hat{r}\in\mathcal{H}} \mathcal{L}(\hat{r}) + \frac{\lambda}{2}\|\hat{r}\|_{\mathcal{H}}^2.$

Squared Loss

- For the squared loss we choose the form used by the unconstrained Least Square Importance Fitting (uLSIF) [7]:

$$\mathcal{J}(\hat{r}) = \frac{1}{2}\int(\hat{r}(\mathbf{z}) - r(\mathbf{z}))^2 p(\mathbf{z})\, d\mathbf{z} = \frac{1}{2}\,\mathbb{E}_p\hat{r}(\mathbf{z})^2 - \mathbb{E}_q\hat{r}(\mathbf{z}) + C,$$

  where C is a constant.

- Then $\mathcal{L}(\hat{r})$ is defined by the Monte Carlo estimate of $\mathcal{J}(\hat{r})$, using samples from p and q:

$$\mathcal{L}(\hat{r}) = \hat{\mathcal{J}}(\hat{r}) = \frac{1}{2n_p}\sum_{i=1}^{n_p}\hat{r}(\mathbf{z}_i^p)^2 - \frac{1}{n_q}\sum_{j=1}^{n_q}\hat{r}(\mathbf{z}_j^q) + C,$$

$$\mathbf{z}_i^p \sim p(\mathbf{z}),\ \mathbf{z}_j^q \sim q(\mathbf{z}).$$

KIVI: Estimating the KL-term

Objective $\min_{\hat{r}\in\mathcal{H}} \mathcal{L}(\hat{r}) + \frac{\lambda}{2}\|\hat{r}\|_{\mathcal{H}}^2.$

Proposition

The optimal solution of the above equation lies in the linear subspace spanned by the kernel functions with the samples $(\mathbf{z}_{1:n_p}^p, \mathbf{z}_{1:n_q}^q)$ as bases, i.e., $\hat{r}$ has the form:

$$\hat{r} = \sum_{i=1}^{n_p} \alpha_i k(\mathbf{z}_i^p, \cdot) + \sum_{j=1}^{n_q} \beta_j k(\mathbf{z}_j^q, \cdot).$$

Proof.

This can be seen as the generalization of the representer theorem [20] to the density ratio problem. So the proof follows the same procedure. See Appendix. □

KIVI: Estimating the KL-term

Objective $\min_{\hat{r}\in\mathcal{H}} \mathcal{L}(\hat{r}) + \frac{\lambda}{2}\|\hat{r}\|^2_{\mathcal{H}}.$

Plug in the optimal form

$$\hat{r} = \sum_{i=1}^{n_p} \alpha_i k(\mathbf{z}_i^p, \cdot) + \sum_{j=1}^{n_q} \beta_j k(\mathbf{z}_j^q, \cdot)$$

and let derivatives to be zeros, we get the optimal solution:

$$\boldsymbol{\beta} = \frac{1}{\lambda n_q}\mathbf{1}, \quad \boldsymbol{\alpha} = -\frac{1}{\lambda n_p n_q}\left(\frac{1}{n_p}\mathbf{K}_p + \lambda\mathbf{I}\right)^{-1}\mathbf{K}_{pq}\mathbf{1},$$

where $(\mathbf{K}_p)_{i,j} = k(\mathbf{z}_i^p, \mathbf{z}_j^p)$, $(\mathbf{K}_{pq})_{i,j} = k(\mathbf{z}_i^p, \mathbf{z}_j^q)$, and $(\mathbf{K}_q)_{i,j} = k(\mathbf{z}_i^q, \mathbf{z}_j^q)$.

Note

$\mathbf{K}_p, \mathbf{K}_{pq}, \mathbf{K}_q$ are submatrices of the Gram matrix formed by $\mathbf{z}^p_{1:n_p}, \mathbf{z}^q_{1:n_q}$:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_p & \mathbf{K}_{pq} \\ \mathbf{K}_{pq}^T & \mathbf{K}_q \end{bmatrix}.$$

KIVI: Estimating the KL-term

The reverse ratio trick

$$\mathcal{J}(\hat{r}) = \frac{1}{2}\int (\hat{r}(\mathbf{z}) - r(\mathbf{z}))^2 p(\mathbf{z})\, d\mathbf{z} = \frac{1}{2}\,\mathbb{E}_p \hat{r}(\mathbf{z})^2 - \mathbb{E}_q \hat{r}(\mathbf{z}) + C.$$

- **Key observation**: The squared loss $\hat{\mathcal{J}}(\hat{r})$ we use puts more weights into regions where the probability mass of p is high, while $\mathrm{KL}(q\|p)$ chooses q as base measure.

KIVI: Estimating the KL-term

The reverse ratio trick

$$\mathcal{J}(\hat{r}) = \frac{1}{2}\int (\hat{r}(\mathbf{z}) - r(\mathbf{z}))^2 p(\mathbf{z})\, d\mathbf{z} = \frac{1}{2}\,\mathbb{E}_p \hat{r}(\mathbf{z})^2 - \mathbb{E}_q \hat{r}(\mathbf{z}) + C.$$

- **Key observation**: The squared loss $\hat{\mathcal{J}}(\hat{r})$ we use puts more weights into regions where the probability mass of p is high, while $\mathrm{KL}(q\|p)$ chooses q as base measure.
- **Solution**: Instead of estimating $\frac{q}{p}$, we choose to estimate $\frac{p}{q}$ and compute the KL term as

  $$\mathrm{KL}(q\|p) = -\mathbb{E}_q \log \frac{p}{q}.$$

  We denote the estimated reverse density ratio as $\hat{r}_{pq}$, then the corresponding KL estimate is $-\mathbb{E}_q \log \hat{r}_{pq}$.

KIVI: Gradient Estimation of the KL-term

To estimate the gradient of the KL term w.r.t. variational parameters ϕ. First it's easy to prove as in [6] that

$$\nabla_\phi \mathrm{KL}(q_\phi \| p) = -\nabla_\phi \mathbb{E}_{q_\phi} \log \frac{p}{q_\phi} = -\nabla_\phi \mathbb{E}_{q_\phi} \log \frac{p}{q}.$$

Note

The above equation indicates that we can use any approximation of the density ratio, and the gradients w.r.t. ϕ won't change as long as the approximation is accurate.

Now replace p/q on the right side with $\hat{r}_{pq}$:

$$\nabla_\phi \mathrm{KL}(q_\phi \| p) \approx -\nabla_\phi \mathbb{E}_{q_\phi} \log \hat{r}_{pq}.$$

Then, the reparameterization trick [9] can be used:

$$-\nabla_\phi \mathbb{E}_{q_\phi} \log \hat{r}_{pq} = -\mathbb{E}_{\epsilon \sim N(0,I)} \nabla \log \hat{r}_{pq}(\mathbf{z}^q(\epsilon; \phi)).$$

KIVI: The Algorithm

Algorithm 1 Kernel Implicit Variational Inference (KIVI)

```
Require: Observed data $\mathbf{x}$, model $p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$.
Require: Implicit variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$, $n_p$, $n_q$, $M$.
repeat
    Sample from prior: $\mathbf{z}_i^p \sim p(\mathbf{z}),\ i = 1, \ldots, n_p$.
    Sample from variational: $\mathbf{z}_j^q \sim q(\mathbf{z}|\mathbf{x}),\ j = 1, \ldots, n_q$.
    Compute the density ratio $\hat{r}_{pq}$ and clip $\hat{r}_{pq}$ to be positive at $\mathbf{z}^q$s.
    Compute $\hat{\mathcal{L}} = \frac{1}{M}\sum_{m=1}^{M} \log p(\mathbf{x}|\mathbf{z}_m^q) + \frac{1}{n_q}\sum_{j=1}^{n_q} \log \hat{r}_{pq}(\mathbf{z}_j^q)$.
    Estimate $\nabla_\phi \mathcal{L}$ with the reparameterization trick.
    Do gradient descent with $\nabla_\phi \mathcal{L}$.
    (Optional) For parameter learning, do gradient descent with $\nabla_\theta \mathcal{L}$.
until Convergence
```

KIVI: Summary

KIVI addresses existing challenges of implicit VI methods.

- The ratio estimates are given in **closed-forms**, thus not having the problem of not catching up.
- The **bias/variance trade-off** of the estimation can be controlled by the regularization coefficient λ.
- KIVI is directly applicable to both global and local LVMs, which is an advantage over nonparametric VI methods (e.g., PMD [3] and SGVD [11]).

Note: Effects of λ

- When λ is set smaller, the estimation is more aggressive to match the samples.
- When λ is set larger, the estimated ratio function is smoother.

Choosing the appropriate λ, the variance of estimation can be controlled while maintaining a reasonably good fit.

Example: Implicit Variational Bayesian Neural Networks

Example: Implicit Variational Bayesian Neural Networks

In BNNs, a prior is specified over the neural network parameters $\mathbf{W} = \{\mathbf{W}_l\}_{l=1}^{L}$, where $\mathbf{W}_l$ indicates weights in the l-th layer. Given input $\mathbf{x}$, the output y is modeled with

$$\mathbf{W} \sim N(\mathbf{0}, \mathbf{I}), \quad \hat{y} = f_{\mathrm{NN}}(\mathbf{x}, \mathbf{W}), \quad y \sim \mathcal{P}(\hat{y}; \theta), \tag{1}$$

Dataset: $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}, \mathbf{Y} = \{y_i\}_{i=1}^{N}$. We have the ELBO:

$$\mathcal{L}(\mathbf{Y}, \mathbf{X}; \phi) = \mathbb{E}_{q_\phi(\mathbf{W})} \log p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) - \mathrm{KL}(q_\phi(\mathbf{W}) \| p(\mathbf{W})).$$

The variational posterior is usually set to be factorized by layer: $q_\phi(\mathbf{W}) = \prod_{l=1}^{L} q_{\phi_l}(\mathbf{W}_l)$. Enabled to learn implicit variational posterior, we propose to adopt a general distribution without an explicit density function, which has a form of

$$\mathbf{W}_l^0 \sim N(\mathbf{0}, \mathbf{I}), \quad \mathbf{W}_l^q = g_{\phi_l}(\mathbf{W}_l^0). \tag{2}$$

Example: Implicit Variational Bayesian Neural Networks

$$\mathbf{W}_l^0 \sim N(\mathbf{0}, \mathbf{I}), \quad \mathbf{W}_l^q = g_{\phi_l}(\mathbf{W}_l^0). \tag{3}$$

How to design a flexible and efficient g?

We present *Matrix Multiplication Neural Network* (MMNN), an efficient framework for sampling large matrices. Deploying MMNN, KIVI can easily scale up to large BNNs.

Example: Implicit Variational Bayesian Neural Networks

Algorithm 2 Matrix Multiplication Neural Network (MMNN)

```
Require: Input matrix $\mathbf{X}_0$
Require: network parameters $\{\mathbf{W}_i^l, \mathbf{B}_i^l, \mathbf{W}_i^r, \mathbf{B}_i^r\}_{i=1}^L$
for $i = 1, \dots, L$ do
    left multiplication: $\mathbf{X}_i = \mathbf{W}_i^l \mathbf{X}_{i-1} + \mathbf{B}_i^l$
    right multiplication: $\mathbf{X}_i = \mathbf{X}_i \mathbf{W}_i^r + \mathbf{B}_i^r$
    if $i \leq L - 1$ then
        $\mathbf{X}_i = \mathrm{Relu}\left(\mathbf{X}_i\right)$
    end if
end for
Output $\mathbf{X}_L$
```

Example: Implicit Variational Bayesian Neural Networks

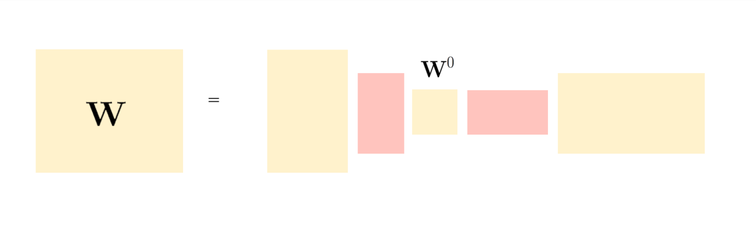


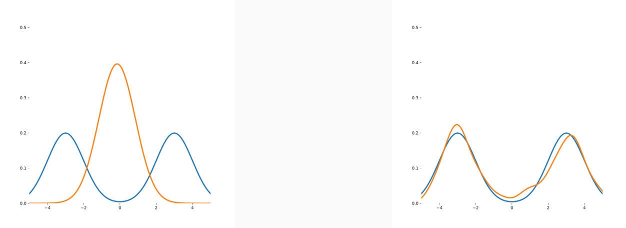
Figure 3: A 2-layer implicit posterior (bias ignored)

To model the implicit posterior of $\mathbf{W}_l$, we only need to randomly sample a matrix $\mathbf{W}_l^0$ of smaller size $M_0 \times N_0$, and feed it forward through the MMNN to get the output variational samples $(\mathbf{W}_l^q)$:

$$\mathbf{W}_l^0 \sim N(\mathbf{0}, \mathbf{I}), \quad \mathbf{W}_l^q = \mathrm{MMNN}_{\phi_l}(\mathbf{W}_l^0). \tag{4}$$

Experiments

Experiments: Toy 1-D Gaussian Mixture



(a) VI (normal posterior)

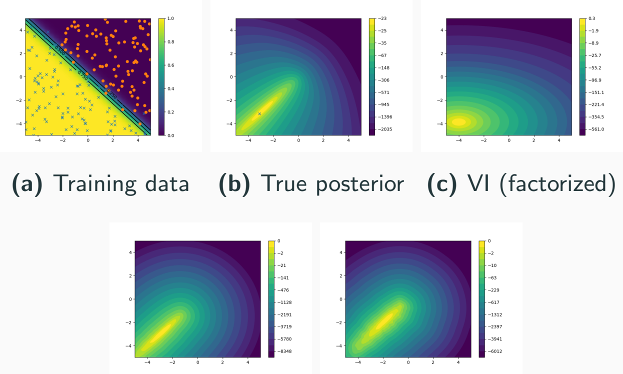
(b) KIVI

Figure 4: Fitting Gaussian Mixture distribution

Experiments: 2-D Bayesian Logistic Regression

$$\mathbf{w} \sim N(\mathbf{0}, \mathbf{I}), \quad y_i \sim \mathrm{Bernoulli}(\sigma(\mathbf{w}^T \mathbf{x}_i)), \quad i = 1, \ldots, N$$

where $\mathbf{w}, \mathbf{x}_i \in \mathbb{R}^2$; σ is the sigmoid function. $N = 200$ data points ($\{(x_i, y_i)\}_{i=1}^{200}$) are generated from the true model as the training data.



(a) Training data **(b)** True posterior **(c)** VI (factorized)

(d) HMC **(e)** KIVI

Experiments: Bayesian Neural Networks

Regression

Table 1: Average test set RMSE, predictive log-likelihood for the regression datasets.

Dataset	Avg. Test RMSE			Avg. Test LL		
	SVGD	Dropout	KIVI	SVGD	Dropout	KIVI
Boston	2.957±0.099	2.97±0.19	**2.798±0.173**	-2.504±0.029	**-2.46±0.06**	-2.527±0.102
Concrete	5.324±0.104	5.23±0.12	**4.702±0.116**	-3.082±0.018	**-3.04±0.02**	-3.054±0.043
Energy	1.374±0.045	1.66±0.04	**0.467±0.015**	-1.767±0.024	-1.99±0.02	**-1.298±0.005**
Kin8nm	0.090±0.001	0.10±0.00	**0.075±0.001**	0.984±0.008	0.95±0.01	**1.162±0.008**
Naval	0.004±0.000	0.01±0.00	**0.001±0.000**	4.089±0.012	3.80±0.01	**5.501±0.121**
Combined	4.033±0.033	4.02±0.04	**3.976±0.037**	-2.815±0.008	-2.80±0.01	**-2.794±0.009**
Protein	4.606±0.013	4.36±0.01	**4.255±0.019**	-2.947±0.003	-2.89±0.00	**-2.868±0.005**
Wine	**0.609±0.010**	0.62±0.01	0.629±0.008	**-0.925±0.014**	-0.93±0.01	-0.958±0.015
Yacht	0.864±0.052	1.11±0.09	**0.737±0.068**	**-1.225±0.042**	-1.55±0.03	-2.123±0.010
Year	**8.684±NA**	8.849±NA	8.950±NA	**-3.580±NA**	-3.588±NA	-3.615±NA

Experiments: Bayesian Neural Networks

Regression

Table 2: Test RMSE, log-likelihood for the regression datasets. Factorized and NF represent VI with factorized normal posteriors and normalizing flow, respectively.

RMSE	**Factorized**	**NF**	**KIVI**
boston	3.42±0.19	3.43±0.19	**2.80**±**0.17**
concrete	6.00±0.10	6.04±0.10	**4.70**±**0.12**
energy	2.42±0.06	2.48±0.09	**0.47**±**0.02**
kin8nm	0.09±0.00	0.09±0.00	**0.08**±**0.00**
naval	0.01±0.00	0.01±0.00	**0.00**±**0.00**
LL	**Factorized**	**NF**	**KIVI**
boston	-2.66±0.04	-2.66±0.04	**-2.53**±**0.10**
concrete	-3.22±0.06	-3.24±0.06	**-3.05**±**0.04**
energy	-2.34±0.02	-2.36±0.03	**-1.30**±**0.01**
kin8nm	0.96±0.01	1.01±0.01	**1.16**±**0.01**
naval	4.00±0.11	4.04±0.12	**5.50**±**0.12**

Experiments: Bayesian Neural Networks

Classification

Method	# Hidden	# Weights	Test err.
SGD [21]	800	1.3m	1.6%
Dropout [22]			$\approx$ 1.3%
Dropconnect [26]	800	1.3m	**1.2%***
Bayes B. [2],	400	500k	1.82%
with Gaussian posterior	800	1.3m	1.99%
	1200	2.4m	2.04%
Bayes B. [2],	400	500k	1.36%*
with scale mixture prior	800	1.3m	1.34%*
	1200	2.4m	1.32%*
KIVI	400	500k	**1.29%**
	800	1.3m	**1.22%**
	1200	2.4m	**1.27%**

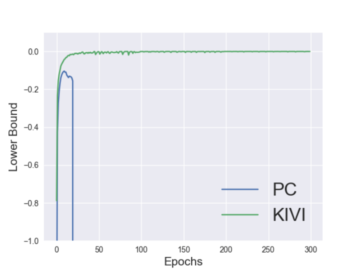
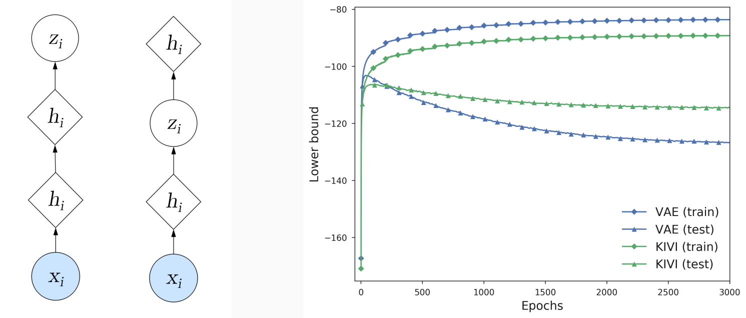


Figure 6: Results for MNIST classification. The left table shows the test error rates. $\star$ indicates results that are not directly comparable to ours: [26] used an ensemble of 5 networks, and the second part of [2] changed the prior to a scale mixture. The plot on the right shows training lower bound in MNIST classification with prior-contrastive (PC) and KIVI.

Experiments: Variational Autoencoders

MNIST: Overfitting



(a) **(b)**

Figure 7: Variational Autoencoders: (a) Gaussian posterior vs. implicit posterior; (b) Training and evaluation curves of the lower bounds on statically binarized MNIST.

Experiments: Variational Autoencoders

CelebA: Interpolation

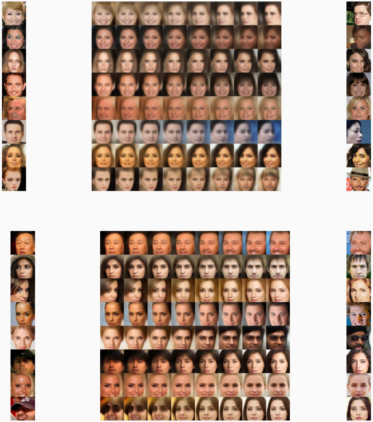
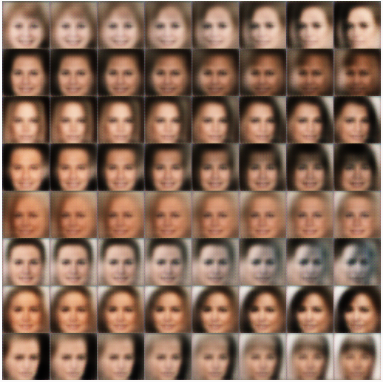


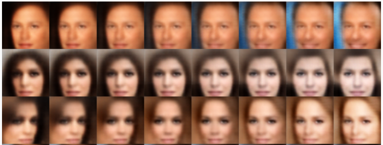
Figure 8: Interpolation experiments for CelebA: AVB (top); KIVI (bottom).

Experiments: Variational Autoencoders

CelebA: A walk through the training process

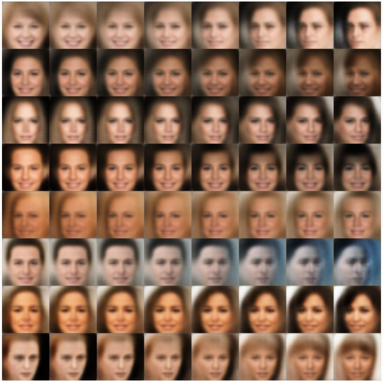


(a) AVB



Experiments: Variational Autoencoders

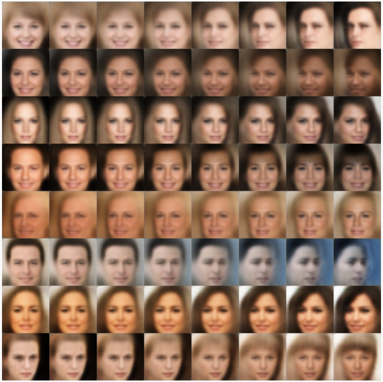
CelebA: A walk through the training process



(a) AVB

Experiments: Variational Autoencoders

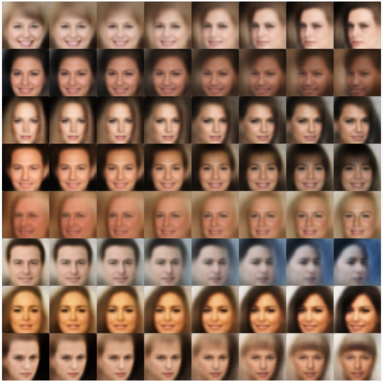
CelebA: A walk through the training process



(a) AVB

Experiments: Variational Autoencoders

CelebA: A walk through the training process



(a) AVB

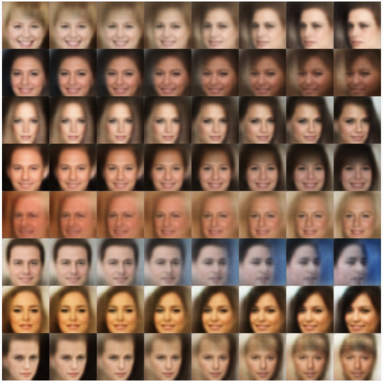
Experiments: Variational Autoencoders

CelebA: A walk through the training process

(a) AVB

Experiments: Variational Autoencoders

CelebA: A walk through the training process



(a) AVB

Conclusion

Conclusion

We present an implicit variational inference method named **Kernel Implicit Variational Inference** (KIVI), which addresses the existing challenges of implicit VI, including noisy estimation and scalability with high-dimensional latent variable models.

We successfully apply this approach to Bayesian neural networks and achieve superior performance on both regression and classification tasks. We also demonstrate that KIVI can be applied to learn local latent variable models like VAEs with implicit posteriors successfully.

Questions?

References i

- Anonymous.
  Backpropagation through the void: Optimizing control variates for black-box gradient estimation.
  International Conference on Learning Representations, 2018.

- C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra.
  Weight uncertainty in neural networks.
  arXiv preprint arXiv:1505.05424, 2015.

- B. Dai, N. He, H. Dai, and L. Song.
  Provable bayesian inference via particle mirror descent.
  arXiv preprint arXiv:1506.03101, 2015.

References ii

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio.
Generative adversarial nets.
In *Advances in neural information processing systems*, pages 2672–2680, 2014.

M. D. Hoffman, D. M. Blei, C. Wang, and J. W. Paisley.
Stochastic variational inference.
Journal of Machine Learning Research, 14(1):1303–1347, 2013.

F. Huszár.
Variational inference using implicit distributions.
arXiv preprint arXiv:1702.08235, 2017.

T. Kanamori, S. Hido, and M. Sugiyama.
A least-squares approach to direct importance estimation.
Journal of Machine Learning Research, 10(Jul):1391–1445, 2009.

References iii

D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling.
Improved variational inference with inverse autoregressive flow.

In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

D. P. Kingma and M. Welling.
Auto-encoding variational bayes.
arXiv preprint arXiv:1312.6114, 2013.

Q. Liu and Y. Feng.
Two methods for wild variational inference.
arXiv preprint arXiv:1612.00081, 2016.

References iv

- Q. Liu and D. Wang.
  Stein variational gradient descent: A general purpose bayesian inference algorithm.
  In *Advances In Neural Information Processing Systems*, pages 2370–2378, 2016.

- C. Louizos and M. Welling.
  Structured and efficient variational deep learning with matrix gaussian posteriors.
  arXiv preprint arXiv:1603.04733, 2016.

- L. Mescheder, S. Nowozin, and A. Geiger.
  Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks.
  arXiv preprint arXiv:1701.04722, 2017.

References v

A. Mnih and K. Gregor.
Neural variational inference and learning in belief networks.
In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

A. Mnih and D. J. Rezende.
Variational inference for monte carlo objectives.
arXiv preprint arXiv:1602.06725, 2016.

S. Mohamed and B. Lakshminarayanan.
Learning in implicit generative models.
arXiv preprint arXiv:1610.03483, 2016.

J. Paisley, D. Blei, and M. Jordan.
Variational bayesian inference with stochastic search.
arXiv preprint arXiv:1206.6430, 2012.

References vi

R. Ranganath, D. Tran, J. Altosaar, and D. Blei.
Operator variational inference.
In *Advances in Neural Information Processing Systems*, pages 496–504, 2016.

D. J. Rezende and S. Mohamed.
Variational inference with normalizing flows.
arXiv preprint arXiv:1505.05770, 2015.

B. Schölkopf, R. Herbrich, and A. Smola.
A generalized representer theorem.
In *Computational learning theory*, pages 416–426. Springer, 2001.

References vii

P. Y. Simard, D. Steinkraus, and J. C. Platt.
Best practices for convolutional neural networks applied to visual document analysis.
In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 958–963, Aug 2003.

N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov.
Dropout: a simple way to prevent neural networks from overfitting.
Journal of machine learning research, 15(1):1929–1958, 2014.

S. Sun, C. Chen, and L. Carin.
Learning structured weight uncertainty in bayesian neural networks.
In *Artificial Intelligence and Statistics*, pages 1283–1292, 2017.

References viii

- D. Tran, R. Ranganath, and D. M. Blei.
  Deep and hierarchical implicit models.
  arXiv preprint arXiv:1702.08896, 2017.

- G. Tucker, A. Mnih, C. J. Maddison, and J. Sohl-Dickstein.
  Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models.
  arXiv preprint arXiv:1703.07370, 2017.

- L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus.
  Regularization of neural networks using dropconnect.
  In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pages 1058–1066, 2013.