Bayesian Dark Knowledge

Anoop Korattikara, Vivek Rathod, Kevin Murphy, Max Welling

May 20, 2026

YunSeop Shin, Seoul national university, statistics, IDEA LAB

Notation

- $x \in \mathcal{X}^D$: Model input.
- $y \in \mathcal{Y}$: Model output.
- $\mathcal{D}_N = \{(x_i, y_i)\}_{i=1}^N$: Given Datasets.
- θ : Model parameter.
- $p(y \mid x, \theta)$: Likelihood such that,

$$p(y \mid x, \theta) = \begin{cases} \mathrm{softmax}(f_\theta(x))_y, & \text{for classification,} \\ \mathcal{N}(y; f_\theta(x), \sigma^2), & \text{for regression.} \end{cases} \tag{1}$$

- $p(\theta)$: Given the prior distribution.
- $p(\theta|\mathcal{D}_N) \propto p(\theta) \prod_{i=1}^N p(y_i|x_i, \theta)$: The posterior distribution over the model parameters.
- $S(y|x, w)$: Student model with learnable parameter w.
- $\mathcal{D}' = \{x'_1, \cdots, x'_M\}$: Design points used in training w.

Motivation

- The authors argue that Variational Inference (VI) and Expectation Propagation (EP) often use factorized Gaussian as proxy distributions. This approximation can be restrictive because it may fail to capture posterior correlations, multimodality, or skewness.
- Moreover, these methods can be difficult to derive and implement in practice.
- Online Markov chain Monte Carlo (MCMC) methods, such as Stochastic Gradient Langevin Dynamics (SGLD), are relatively easy to implement.
- However, they require storing many posterior samples and performing multiple forward passes at inference time, which increases both memory usage and computational cost.

Solution

- The authors propose to regard the posterior predictive distribution obtained from MCMC as a teacher distribution.
- Specifically, the teacher distribution is given by

$$p(y \mid x, D_N) \approx \frac{1}{S} \sum_{s=1}^{S} p(y \mid x, \theta^s), \tag{2}$$

  where $\{\theta^s\}_{s=1}^{S}$ are posterior samples generated by MCMC.
- They then distill this teacher distribution into a single student model $S(y \mid x, w)$.
- The student is trained to distill the teacher predictive distribution by minimizing

$$KL\left(p(y \mid x, D_N) \,\|\, S(y \mid x, w)\right). \tag{3}$$

- The authors use SGLD in their implementation.

Overall Method

- Let $\Theta = \{\theta^s\}_{s=1}^S$ be posterior samples generated by SGLD.
- The goal is to train the student model $S(y \mid x, w)$ to distill the posterior predictive distribution.
- For a fixed input x, the objective is

$$\begin{aligned} L(w \mid x) &= KL\left(p(y \mid x, D_N) \,\|\, S(y \mid x, w)\right) \\ &= -\int p(\theta|\mathcal{D}_N)\mathbb{E}_{p(y|x,\theta)} \log S(y \mid x, w) d\theta. \end{aligned} \tag{4}$$

- Since the posterior predictive distribution is intractable, the authors use the Monte Carlo approximation

$$\hat{L}(w \mid x) = -\frac{1}{|\Theta|} \sum_{\theta^s \in \Theta} \mathbb{E}_{p(y|x,\theta^s)} \log S(y \mid x, w). \tag{5}$$

- In practice SGLD updates the teacher parameter θ, and then SGD updates the student parameter w.

Algorithm

Algorithm 1: Distilled SGLD

Input: $\mathcal{D}_N = \{(x_i, y_i)\}_{i=1}^N$, minibatch size M, number of iterations T, teacher learning schedule η_t, student learning schedule ρ_t, teacher prior λ, student prior γ

for $t = 1 : T$ **do**

// Train teacher (SGLD step)

Sample minibatch indices $S \subset [1, N]$ of size M

Sample $z_t \sim \mathcal{N}(0, \eta_t I)$

Update $\theta_{t+1} := \theta_t + \frac{\eta_t}{2}\left(\nabla_\theta \log p(\theta|\lambda) + \frac{N}{M}\sum_{i\in S} \nabla_\theta \log p(y_i|x_i, \theta)\right) + z_t$

// Train student (SGD step)

Sample $\mathcal{D}'$ of size M from student data generator

$w_{t+1} := w_t - \rho_t \left(\frac{1}{M}\sum_{x'\in\mathcal{D}'} \nabla_w \hat{L}(w, \theta_{t+1}|x') + \gamma w_t\right)$

- The algorithm alternates between an SGLD update for the teacher and an SGD update for the student.
- Importantly, each student update uses only the current posterior sample θ_{t+1}.
- Next, we define $\hat{L}(w \mid \theta, x)$ separately for classification and regression.

Classification

- For classification, each teacher network uses a softmax likelihood:

$$p(y = k \mid x, \theta^s) = \mathrm{softmax}(f(x \mid \theta^s))_k. \tag{6}$$

- The student network also outputs a softmax distribution:

$$S(y = k \mid x, w) = \mathrm{softmax}(g(x \mid w))_k. \tag{7}$$

- The distillation loss becomes the cross entropy between the teacher predictive distribution and the student predictive distribution:

$$\hat{L}(w \mid \theta^s, x) = -\sum_{k=1}^{K} p(y = k \mid x, \theta^s) \log S(y = k \mid x, w). \tag{8}$$

Regression

- For regression, the teacher likelihood is assumed to be Gaussian:

$$p(y \mid x, \theta^s) = \mathcal{N}\left(y; f(x \mid \theta^s), \lambda_n^{-1}\right), \tag{9}$$

  where λ_n is the noise precision.

- The student approximates the posterior predictive distribution by another Gaussian:

$$S(y \mid x, w) = \mathcal{N}\left(y; \mu(x, w), \exp(\alpha(x, w))\right). \tag{10}$$

- Thus, the student network outputs both the predictive mean $\mu(x, w)$ and the log variance $\alpha(x, w)$.

- The resulting loss is

$$\hat{L}(w \mid \theta^s, x) = \frac{1}{2}\left[\alpha(x, w) + e^{-\alpha(x,w)}\left\{\left(f(x \mid \theta^s) - \mu(x, w)\right)^2 + \frac{1}{\lambda_n}\right\}\right]. \tag{11}$$

Comparison with Our Method

- Both methods use SGLD samples, but use them for different purposes.
- Bayesian Dark Knowledge directly distills the student predictive distribution $S(y \mid x, w)$.
- That is, its final target is the posterior predictive distribution itself.
- In contrast, our method approximates the posterior distribution over the prediction function f.
- The posterior predictive distribution is then obtained from the learned posterior distribution of the prediction function.
- Therefore, Bayesian Dark Knowledge is a predictive distribution distillation method, whereas our method approximates the posterior distribution of the prediction function.

Comparison with Our Method

Let x_1 and x_2 be two given inputs, and let Y_1 and Y_2 be their corresponding labels. Given a prediction function f, the two labels are conditionally independent:

$$Y_1 \perp Y_2 \mid f, x_1, x_2, D. \tag{12}$$

However, after marginalizing out f, they are generally dependent:

$$Y_1 \not\perp Y_2 \mid x_1, x_2, D. \tag{13}$$

Therefore, the Bayesian probability that the two labels are identical is

$$\begin{aligned} p(Y_1 = Y_2 \mid x_1, x_2, D) &= \sum_{c=1}^{C} p(Y_1 = c, Y_2 = c \mid x_1, x_2, D) \\ &= \sum_{c=1}^{C} \int p(Y_1 = c \mid x_1, f) p(Y_2 = c \mid x_2, f) \pi(f \mid D)\, df. \end{aligned}$$

Comparison with Our Method

In contrast, a method that only distills the posterior predictive distribution can only form

$$\sum_{c=1}^{C} p(Y_1 = c \mid x_1, D) p(Y_2 = c \mid x_2, D), \tag{14}$$

which ignores posterior dependence across inputs.