Bayesian ensembling - *Bensemble*

Sobolevsky Fedor, Nabiev Muhammadsharif, Vasilenko Dmitriy, Kasiuk Vadim

Moscow Institute of Physics and Technology

2025

Library Interface

- **Unified API**: Single entry point for all ensembling methods.
- **Model Agnostic**: Works with any differentiable model (Neural Networks, etc.).
- **Posterior Sampling**: Each algorithm provides a different strategy for sampling models from the posterior.
- **Ensemble Generation**: Easy generation of model ensembles for uncertainty quantification and improved performance.
- **Hyperparameter Tuning**: Built-in methods for optimizing algorithm-specific parameters.

Example Usage:

Algorithm 1: Practical Variational Inference (Graves)

Problem:

- Standard neural networks use **point estimates** of weights $\rightarrow$ prone to **overfitting**.
- True **Bayesian posterior** $p(w \mid D)$ is **intractable** for large networks.

Goal:

- Approximate posterior with a **variational distribution** $q_\theta(w) = \mathcal{N}(\mu, \sigma^2)$.
- Optimize **ELBO** to make $q_\theta(w)$ close to $p(w \mid D)$.
- Make it **practical** and compatible with gradient-based training.

Benefit:

- Captures **uncertainty** in predictions.
- Improves **generalization**.
- Enables **weight pruning** and compact models.
- Simple to integrate into **existing networks**.

Variational Inference & ELBO

- Introduce **variational posterior** $q_\theta(w)$.
- Define **ELBO (Evidence Lower Bound)**:

$$\log p(D) \geq \mathbb{E}_{q_\theta(w)}[\log p(D \mid w)] - \mathrm{KL}(q_\theta(w) \| p(w))$$

- ELBO balances:
  - **Accuracy:** $\mathbb{E}_{q_\theta}[\log p(D|w)]$
  - **Regularization:** $\mathrm{KL}(q_\theta \| p)$
- Maximizing ELBO $\rightarrow q_\theta(w)$ approximates true posterior.

Reparameterization Trick

- To backprop through stochastic weights:

$$w = \mu + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

- $\odot$ — element-wise multiplication
- Allows **gradient-based optimization** of μ and σ
- Enables **stochastic gradient descent** on ELBO

Algorithm 2: Scalable Laplace Approximation

- **Problem**: Exact Bayesian inference in neural networks is intractable due to large parameter spaces.
- **Goal**: A scalable, post hoc method to approximate the posterior without retraining the model.
- **Benefit**: Fast uncertainty estimates for pre-trained models used in production.

Algorithm 2: Scalable Laplace Approximation

Laplace Approximation

$$p(\theta|\mathcal{D}) \approx \mathcal{N}(\theta; \theta^*, \bar{H}^{-1})$$

where θ^* is the MAP estimate and $\bar{H}$ is the average Hessian of the negative log-posterior.

Kronecker-Factored Hessian

For layer λ, the Hessian block is approximated as:

$$H_\lambda \approx \mathbb{E}[\mathcal{Q}_\lambda] \otimes \mathbb{E}[\mathcal{H}_\lambda]$$

where $\mathcal{Q}_\lambda$ is the covariance of inputs and $\mathcal{H}_\lambda$ is the pre-activation Hessian.

Algorithm 2: Scalable Laplace Approximation

Posterior Sampling

Sample weights for layer λ from:

$$W_\lambda \sim \mathcal{MN}(W_\lambda^*, \bar{\mathcal{Q}}_\lambda^{-1}, \bar{\mathcal{H}}_\lambda^{-1})$$

Applied after training, requiring no changes to the original training procedure.

Algorithm 3: Variational Renyi Bound (VR)

- **Problem**: Traditional VI (e.g., VAE) uses KL divergence, which can lead to under-estimated uncertainty (mode-seeking).
- **Goal**: Generalize VI to the rich family of Renyi divergences, enabling interpolation between mode-seeking ($\alpha \to \infty$) and mass-covering ($\alpha \to -\infty$) behavior.
- **Benefit**: Better uncertainty estimates and tighter bounds on the marginal likelihood.

Algorithm 3: Variational Renyi Bound (VR)

- **Core Idea**: Minimize the Renyi α-divergence $D_\alpha[q||p]$ between approximate posterior q and true posterior p.
- **VR Bound**: Derive a new variational bound $\mathcal{L}_\alpha$ that generalizes the ELBO. For $\alpha \to 1$, recover standard VI (KL divergence).
- **Optimization**: Use reparameterization trick and Monte Carlo sampling to optimize $\mathcal{L}_\alpha$ stochastically.
- **Special Case**: $\alpha \to -\infty$ (VR-max) focuses on the sample with the highest importance weight, leading to a fast, high-quality approximation.

Algorithm 3: Variational Renyi Bound (VR)

VR Bound

$$\mathcal{L}_\alpha(q; \mathcal{D}) = \frac{1}{1-\alpha} \log \mathbb{E}_{q(\boldsymbol{\theta})} \left[\left(\frac{p(\boldsymbol{\theta}, \mathcal{D})}{q(\boldsymbol{\theta})} \right)^{1-\alpha} \right]$$

Gradient (Reparameterized)

$$\nabla_\phi \mathcal{L}_\alpha = \mathbb{E}_\epsilon \left[w_\alpha(\epsilon; \phi, \mathcal{D}) \nabla_\phi \log \frac{p(g_\phi(\epsilon), \mathcal{D})}{q(g_\phi(\epsilon))} \right]$$

where w_α is the normalized importance weight.

Key Properties

- Continuous and non-increasing in α.
- For $\alpha < 0$, $\mathcal{L}_\alpha$ is an upper bound on $\log p(\mathcal{D})$; for $\alpha > 0$, a lower bound.
- Enables smooth interpolation between VI ($\alpha = 1$), IWAE ($\alpha = 0$), and VR-max ($\alpha \to -\infty$).

Algorithm 4: Probabilistic Backpropagation (PBP)

- **Problem**: Backpropagation provides point estimates; hyperparameter tuning is costly; predictive uncertainty is ignored.
- **Goal**: A scalable, Bayesian alternative to backprop that provides uncertainty.
- **Benefit**: Combines the efficiency of backprop with the advantages of Bayesian inference.

Algorithm 4: Probabilistic Backpropagation (PBP)

- **Core Idea**: Maintain a Gaussian posterior over each weight. Use moment propagation to compute means and variances of network outputs, and then update the posteriors using gradients of the marginal likelihood.
- **Assumed Density Filtering (ADF)**: Sequentially incorporate data points, approximating the true posterior with a factorized Gaussian.
- **Moment Matching**: Update the Gaussian parameters to match the moments of the posterior after incorporating each data point.
- **Efficiency**: Similar computational cost to backpropagation, but with built-in uncertainty estimation.

Algorithm 4: Probabilistic Backpropagation (PBP)

Factorized Gaussian Posterior

$$q(\mathcal{W}) = \prod_{l,i,j} \mathcal{N}(w_{ij,l}; m_{ij,l}, v_{ij,l})$$

Forward Propagation of Moments

For each layer, compute mean and variance of pre-activations $\mathbf{a}_l$ and activations $\mathbf{z}_l$:

$$\mathbf{m}^{\mathbf{a}_l} = \mathbf{M}_l \mathbf{m}^{\mathbf{z}_{l-1}} / \sqrt{V_{l-1} + 1}, \quad \mathbf{v}^{\mathbf{a}_l} = \cdots$$

$$m_i^{b_l} = \Phi(\alpha_i) v_i', \quad v_i^{b_l} = \cdots$$

Posterior Update via Moment Matching

$$m^{\text{new}} = m + v \frac{\partial \log Z}{\partial m}, \quad v^{\text{new}} = v - v^2 \left[\left(\frac{\partial \log Z}{\partial m} \right)^2 - 2 \frac{\partial \log Z}{\partial v} \right]$$