Investigating Uncertainty in Ensemble Methods

Gurman Bhullar
Department of Computer Science
University of Toronto
gbhullar@cs.toronto.edu

Rajesh Marudhachalam
Department of Computer Science
University of Toronto
rajesh1804@cs.toronto.edu

Sarah Hindawi
Department of Computer Science
University of Toronto
shindawi@cs.toronto.edu

Sumant Bagri
Department of Computer Science
University of Toronto
sbagri@cs.toronto.edu

Abstract

TBD...

1 Introduction

2 Motivation

3 Related Work

4 Theoretical Analysis

4.1 Quantifying Uncertainty

In the domain of Bayesian inference for probabilistic models, given a dataset $\mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1}^{N}$, we try to model the parameters θ by estimating its posterior distribution using Bayes' rule:

$$\mathbb{P}(\theta|\mathcal{D}) = \frac{\mathbb{P}(\mathcal{D}|\theta)\mathbb{P}(\theta)}{\mathbb{P}(\mathcal{D})} \tag{1}$$

A given ensemble of models $\{\mathbb{P}(y|x;\theta^{(m)})\}_{m=1}^{M}$ is assumed to have been generated from a known (or in most cases a close approximation) of a Bayesian posterior $\mathbb{P}(\theta|\mathcal{D})$. The predictive posterior of the ensemble is then defined as follows:

$$\begin{gathered}\mathbb{P}(y|x,\mathcal{D}) = \mathbb{E}_{\mathbb{P}(\theta|\mathcal{D})}\left[\mathbb{P}(y|x,\theta)\right] \\ \text{where, the sample meanestimator is used as an approximation for the expectation} \\ \mathbb{P}(y|x,\mathcal{D}) \simeq \frac{1}{M}\sum_{m=1}^{M}\mathbb{P}(y|x;\theta^{(m)})\end{gathered} \tag{2}$$

Uncertainty from the predictive posterior of the ensemble can be due to two factors: 1) aleatoric (data) uncertainty or 2) epistemic (knowledge) uncertainty. We can write the total uncertainty as a sum of aleatoric and epistemic uncertainties as shown below,

$$\text{Total Uncertainty = Data Uncertainty + Knowledge Uncertainty} \tag{3}$$

We define the total uncertainty as the entropy in the predictive posterior. This is defined below,

$$\mathcal{H}\left[\mathbb{P}(y|x,\mathcal{D})\right] = \mathbb{E}_{\mathbb{P}(y|x,\mathcal{D})}\left[-\log\mathbb{P}(y|x,\mathcal{D})\right] \tag{4}$$

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Now, the data uncertainty for a generalized predictor can be visualized as the expected prediction uncertainty of each model in the ensemble. Mathematically, this can be written as the expected entropy of each model in the ensemble as below,

$$\text{Expected Data Uncertainty} = \mathbb{E}_{\mathbb{P}(\theta|\mathcal{D})}\left[\mathcal{H}(\mathbb{P}(y|x,\theta))\right] \tag{5}$$

Therefore, we can obtain the knowledge uncertainty by rearranging (3)

$$\begin{aligned}
&\text{Knowledge Uncertainty} = \text{Total Uncertainty} - \text{Data Uncertainty} \\
&\text{Knowledge Uncertainty} \simeq \text{Total Uncertainty} - \text{Expected Data Uncertainty} \\
&\text{Knowledge Uncertainty} \simeq \mathcal{H}\left[\mathbb{P}(y|x,\mathcal{D})\right] - \mathbb{E}_{\mathbb{P}(\theta|\mathcal{D})}\left[\mathcal{H}(\mathbb{P}(y|x,\theta))\right] \\
&\qquad \text{this is precisely the information gain from knowing the data} \\
&\mathcal{I}\left[y,\theta|x,\mathcal{D}\right] = \mathcal{H}\left[\frac{1}{M}\sum_{m=1}^{M}\mathbb{P}(y|x,\theta^{(m)}\right] - \frac{1}{M}\sum_{m=1}^{M}\mathcal{H}\left[\mathbb{P}(y|x,\theta)\right]
\end{aligned} \tag{6}$$

However, we can use the above formulation only for classification tasks where predictions are actually indicative of probabilities. For regression tasks, the predictions are point values and we cannot use them as probabilistic estimates for a model. For regression tasks, uncertainty is estimated using total variation which is defined below,

$$\mathbb{V}_{p(y|x,\mathcal{D})}[y] = \mathbb{V}_{p(y|x,\mathcal{D})}[\mathbb{E}_{p(y|x,\mathcal{D})}[y]] + \mathbb{P}_{p(\theta|\mathcal{D})}[\mathbb{V}_{p(y|x,\theta)}[y]] \tag{7}$$

$$\mathbb{V}_{p(y|x,\mathcal{D})}[y] \simeq \frac{1}{M}\sum_{m=1}^{M}\left[\left(\sum_{m-1}^{M}\frac{\mu_m}{M}\right) - \mu_m\right]^2 + \frac{1}{M}\sum_{m=1}^{M}\sigma_m^2 \quad , \quad \{\mu_m,\sigma_m\} = f(x;\theta^{(m)}) \tag{8}$$

The parameters, $\{\mu_m, \sigma_m\}$ are estimated by the training algorithm that models a normal distribution over the target y given features x by optimizing the negative log likelihood for $\mathbb{P}(y|x,\theta)$. This is mathematically presented as follows,

$$\begin{aligned}
&F^{(t)}(x) : X \to \mathbb{R}^2 \Rightarrow \left\{\mu_m^{(t)}, \sigma_m^{(t)}\right\} \\
&\qquad \text{such that,} \\
&\mathbb{P}(y|x,\theta^{(t)}) = \mathcal{N}\left(y|\mu_m^{(t)}, \sigma_m^{(t)}\right), \text{ where,} \\
&\hat{\theta}^{(t)} = \arg\min_{\theta}\mathbb{E}_{\mathcal{D}}[-\log p(y|x,\theta)] = \arg\min_{\theta}\left\{-\frac{1}{N}\sum_{i=1}^{N}\log\mathbb{P}(y^{(i)}|x^{(i)},\theta)\right\}
\end{aligned} \tag{9}$$

Using these uncertainty measures, we can formulate a thresholding-based binary classification problem where the predictor/detector $\mathcal{I}_T(x)$ takes in a new input **x** and assigns a label 1 (uncertain prediction) if the uncertainty measure $\mathcal{H}(x)$ is above a threshold T and label 0 (confident prediction) otherwise. This is presented, mathematically, as below,

$$\mathcal{I}_T(x) = \begin{cases} 1 & ,\mathcal{H}(x) > T \\ 0 & ,\mathcal{H}(x) \leq T \end{cases} \tag{10}$$

4.2 Gradient Boosted Decision Trees (GBDT)

Given a dataset $\mathcal{D}$ and a loss function $L : \mathbb{R}^2 \to \mathbb{R}$, the gradient boosting algorithm iteratively constructs a model $F : X \to \mathbb{R}$ to minimize the empirical risk $\mathcal{L}(F|\mathcal{D}) = \mathbb{E}_{\mathcal{D}}[L(F(x),y)]$ **?**. A weak learner $h^{(t)}$ is chosen at each time-step (t) from a family of functions $\mathcal{H}$ and the model is updated as follows,

$$F^{(t)}(x) = F^{(t-1)}(x) + \epsilon \cdot h^{(t)}(x) \tag{11}$$

such that,

$$h^{(t)} = \arg\min_{h\in\mathcal{H}}\mathbb{E}_{\mathcal{D}}\left[\left(-\left.\frac{\partial\mathcal{L}(y,s)}{\partial s}\right|_{s=F^{(t-1)}(x)} - h(x)\right)^2\right] \tag{12}$$

Here, ϵ is the learning rate for the model F. For the GBDT ensemble the weak learners are decision trees defined using a set of parameters $\theta^{(t)}$ such that $h(x, \phi^{(t)}) = \sum_{j=1}^{d} \phi_j^{(t)} \mathbf{1}_{x \in R_j}$. The ensemble is constructed by summing over all the decision trees

4.3 Generating Ensembles

One of the most important requirements for calculating uncertainty is ensemble models is that each model in the ensemble should be sampled from a posterior distribution. $\mathbb{P}(\theta|\mathcal{D})$. Stochastic Gradient Boosting (SGB) and Stochastic Gradient Langevin Boosting (SGLB)

SGB ensembles

SGB ensemble of GBDTs are generated by subsampling the dataset $\mathcal{D}$ at each iteration of training. A randomly subsampled dataset $\mathcal{D}' \subset \mathcal{D}$ is generated either using bootstrap or uniformly randomly. The fraction of the dataset sampled is called the *sample rate*. The sampling is done independently resulting in an ensemble of independent models $\{\theta^{(m)}\}_{m=1}^{M}$.

The model updates are done using (10) and (11).

SGLB ensembles

SGLB combines gradient boosting along with Langevin dynamics **?** to achieve convergence to global optimum even for non-convex loss functions. The algorithm has two main differences compared to SGB **?**

First, Gaussian noise is added to the SE calculation

$$h^{(t)} = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}} \left[\left(\frac{\partial \mathcal{L}(y, s)}{\partial s} \Big|_{s=F^{(t-1)}(x)} - h(x) + v \right)^2 \right], v \sim \mathcal{N}\left(0, \frac{2}{\beta_\epsilon}\mathbf{I}\right) \tag{13}$$

Second,

$$F^{(t)}(x) = (1 - \gamma\epsilon)F^{(t-1)}(x) + \epsilon \cdot h^{(t)}(x, \phi^{(t)}) \tag{14}$$

Here, β is called the diffusion temperature and helps in controlling the random exploration induced by v. The random exploration is what helps SGLB achieve global optimum

Optimizers and Training Metrics

Classification

The LogLoss optimizer is used for training the classification models using SGB and SGLB algorithms. This is defined as follows,

$$\mathcal{L}\left(y, \{\mathbb{P}(y|x;\theta^{(m)})\}_{m=1}^{M}\right)_{LogLoss} = -\frac{\sum_{m=1}^{M} \left\{ w_m \left(t \log\left[\mathbb{P}(y|x;\theta^{(m)})\right] + (1-t)\log\left[1 - \mathbb{P}(y|x;\theta^{(m)})\right]\right)\right\}}{\sum_{i=m}^{M} w_m} \tag{15}$$

The $0-1$ Loss is used as the evaluation metric during training and testing

Regression

For regression, the RMSEWithUncertainty optimizer is used. This is described below,

$$\mathcal{L}\left(y, \{\mathbb{P}(y|x;\theta^{(m)})\}_{m=1}^{M}\right)_{\text{RMSEWithUncertainty}} = -\frac{\sum_{i=1}^{N} \left\{ w_m \log\left[\mathcal{N}(y|a_{m,0}, e^{2a_{m,1}})\right]\right\}}{\sum_{m=1}^{M} w_m}$$

$$\mathcal{L}_{\text{RMSEWithUncertainty}} = \frac{1}{2}\log(2\pi) + \frac{\sum_{m=1}^{M}\left\{w_m\left(a_{m,0} + \frac{1}{2}e^{-2a_{m,1}}(t-a_{m,0})^2\right)\right\}}{\sum_{m=1}^{M} w_m} \tag{16}$$

Here $a_{m,0}$ is the mean prediction (μ_m) and $a_{m,1}$ is the $\log(\sigma_m)$ prediction such that the model m is sampled from $\mathcal{N}(y|\mu_m, \sigma_m^2) = \frac{1}{\sqrt{2\pi\sigma_m^2}}\exp\left(-\frac{(y-\mu_m)^2}{2\sigma^2}\right)$

The RMSE score is used as the evaluation metric during training and testing

4.4 Metrics for evaluating uncertainty

It is important to know whether the prediction uncertainty from an ensemble (as computed using methods stated above) is able to provide any meaningful insights on predictions: Whether the model will make an error in prediction or whether it will classify it correctly.

AUC-ROC (for out-of-domain datasets)

OOD detection is done using a dataset consisting of both in-domain ($\mathcal{I} \in \mathcal{D}_{in}$) and out-of-domain ($\mathcal{O} \in \mathcal{D}_{out}$) datapoints. Points belonging to $\mathcal{I}$ are labeled as 1 while points in $\mathcal{O}$ are labeled as 0. Therefore the model for the ensemble becomes $F : X \to \{0, 1\}$. The model is trained on the training dataset and its predictions $\mathcal{P}$ are collected from the test dataset ($\mathcal{D}_{test}$) with true labels $\mathcal{T}$. The strategy is to then evaluate the *true positive rate (tpr)* and the *false positive rates (fpr)* as follows

$$\begin{aligned} tpr &= \frac{\sum_{x\in\mathcal{D}_{test}} \mathcal{I}_T(x|\mathcal{T}(x)=0)}{\sum_{x\in\mathcal{D}_{test}} \mathbb{1}\{\mathcal{T}(x)=0\}} \\ fpr &= \frac{\sum_{x\in\mathcal{D}_{test}} \mathcal{I}_T(x|\mathcal{T}(x)=1)}{\sum_{x\in\mathcal{D}_{test}} \mathbb{1}\{\mathcal{T}(x)=1\}} \end{aligned} \tag{17}$$

where $\mathcal{I}_T$ is the detector/predictor as defined in (4.1)

Subsequently, we plot the ROC curvce as: tpr vs fpr, and compute the area under the curve (AUC). The higher the value of AUC-ROC the better is the out-of-domain detection.

Prediction-Rejection Ratio (PRR)

For a classification problem, the measure of uncertainty provides a means to label model predictions as certain or uncertain. Ideally, the goal is to be able to detect all the inputs which the model has misclassified with a label of "uncertain prediction". If the detector $\mathcal{I}_T$ (for a given uncertaintly threshold T), is able to achieve this then the model can choose not to provide any prediction for these inputs, or pass them over to an "oracle" (human) as "rejected-for-prediction". This can be visualized as a rejection curve $\mathcal{R}$ on the percentage of data points in $f * N(\mathcal{D}_{test}) : f \in [0, 1]$

The base error (e_{base}) in classification is defined as the error after the model(F) (and the detector $\mathcal{I}_T$) have gone through the whole test dataset $\mathcal{D}_{test}$. We consider two boundary cases for the detector $\mathcal{I}_T$,

Case 1: Detector is completely unreliable always producing useless estimates for uncertainty ($\mathcal{R}_{rand}$)

Case 2: Detector is perfect and always bigger for a misclassification than for a correct classification. In this case, the curve will represent the oracle curve (where all uncertain misclassifications are given to the oracle to classify) ($\mathcal{R}_{orc}$)

Any other detector would lie between the above two boundary cases and is denoted as $\mathcal{R}_{uns}$

$\mathcal{R}_{rand}$ is a straight line from e_{base} at 0% rejection to 0 at 100% rejection as the rejections are random:

$$\mathcal{R}_{rand} - 0 = \frac{e_{base} - 0}{0 - 1} \times (f - 1) \tag{18}$$

$$\mathcal{R}_{rand} = (1 - f) \times e_{base} \tag{19}$$

$\mathcal{R}_{rand}$ is a straight line from e_{base} at 0% rejection to 0 at the percentage of misclassification $\left(N_{misclassify} = \sum_{i=1}^{N(\mathcal{D}_{test})} \mathbb{1}[y \neq F(x)] \Rightarrow f' = \frac{\sum_{i=1}^{N(\mathcal{D}_{test})} \mathbb{1}[y\neq F(x)]}{N(\mathcal{D}_{test})}\right)$ as the "perfect" detector

will have no need to reject any correctly classfied datapoints:

$$\mathcal{R}_{orc} - 0 = \frac{e_{base} - 0}{0 - f'} \times (f - f') \tag{20}$$

$$\mathcal{R}_{rand} = \begin{cases} e_{base} \times (1 - \frac{f}{f'}) & , f < f' \\ 0 & , 1 \geq f \geq f' \end{cases} \tag{21}$$

The quality of th rejection curve is assesed by considering the *ratio* of the area between $\mathcal{R}_{uns}$ and $\mathcal{R}_{random}$ and is called the ***Prediction-Rejection-Ratio (PRR)*** as shown below,

$$PRR = \frac{AUC(R_{random}) - AUC(R_{orc})}{AUC(R_{uns}) - AUC(R_{orc})} \tag{22}$$

A higher value of PRR indicates that the detector is closer to the oracle in terms of classification error rate and is more certain about the misclassified labels.

5 Experimental Setup

6 Results and Discussion

7 Conclusions