Raphaël Pellegrin, Gael Ancel, David Assaraf - 12/11/20

BaCOUn: Bayesian Classifiers with Out-of-Distribution Uncertainty

As we have seen in homeworks, some classifiers yield models that make confident predictions for OOD points. We partially resolved this issue in Homework 7 by working with non-linear classifiers. It is important to get a notion of the uncertainty associated to a prediction if the classifier is used for some real-world application (in the medical sector, for an air plane system, etc). If the uncertainty associated to the prediction is too high, it might be necessary for a human supervisor to take the decision. Let's say that we have n classes of data points. The main point of the paper is to add a $n+1$th class that surrounds the data-points.

1 Epistemic and aleatoric uncertainty

Epidemic uncertainty is uncertainty that is due to a lack of observations. It can be reduced with further observations. aleatoric uncertainty is uncertainty that is due to the intrinsic uncertainty in the data, and can't be reduced with further observations.

In the case of classifiers, the epistemic uncertainty can have two causes: being far away from training points, and lying close to boundaries. Points lying far away from training points are called OOD points.

2 Previous techniques to estimate the uncertainty in classification

Gaussian Processes (GPs) have been the gold standard to estimate the uncertainty in classification. However, they become computationally intractable as the number of parameters grow.

2.1 Neural Networks

It is possible to use Bayesian Neural Networks (BNN) by placing priors on all the weights of a neural net. It is important to note that a result of Neal (1996) shows that BNN are equivalent to GP in the infinite width limit. However, this is computationally intractable. Furthermore, BNN are not great for estimating the OOD uncertainty when their size is finite and small. Even then, it can be computationally expensive to train the model with priors on all the weights.

Thus, Neural Linear Models (NLM), where priors are put on the weights in the last hidden layers were used instead of BNN. The remaining weights are learnt. Of course, this is not great for estimating the OOD uncertainty (it is more restrictive than BNN, which are not satisfactory already).

2.2 Nitty gritty of NLMs

The data is $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$. We have N data points, $\{x_i\}_{i=1}^N$, that each belong to $\mathbb{R}^D$ and come with a label $y_i \in [n] := \{1, \dots, n\}$. It is assumed that the labels come from a categorical distribution:

$$y|x \sim \text{Cat}(\text{softmax}(W^T \phi_\theta(x))), W \sim p(W) \tag{1}$$

where, given $\{p_1, \dots, p_n\}$ the categorical random variable has support $[n]$ and probability mass function:

$$p(x) = \prod_{i=1}^n p_i^{[x=i]} \tag{2}$$

and where, given $\mathbf{z} = (z_1, \dots, z_n)$, the softmax function is defined as:

$$\sigma(\mathbf{z})_j = \frac{\mathrm{e}^{z_j}}{\sum_{k=1}^K \mathrm{e}^{z_k}} \tag{3}$$

for all $j \in \{1, \dots, n\}$. Thus, the softmax function is a function from $\mathbb{R}^n$ to $\mathbb{R}^n$.

ϕ_θ is called the feature map because it extracts information from the data in order to proceed to classification. The feature map is trained to maximize the observed data log-likelihood:

$$\theta^* = \arg\max_{\theta, W} p(y_1, \dots, Y_N | x_1, \dots, x_N, \theta, W) \tag{4}$$

Then, the posterior for the weights $p(W|\theta^*, \mathcal{D})$ can be inferred using $\phi_{\theta*}$. The posterior is untractable so Hamiltonian Monte Carlo (HMC) or mean-field Gaussian variational inference is used.

It is then possible to make new predictions using:

$$p(y_{\text{new}}|x_{\text{new}}, \mathcal{D}, \theta^*) = \int p(y_{\text{new}}, W|x_{\text{new}}, \mathcal{D}, , \theta^*) dW \tag{5}$$

$$= \int p(y_{\text{new}}|x_{\text{new}}, \mathcal{D}, \theta^*, W) p(W|x_{\text{new}}, \mathcal{D}, \theta^*) dW \tag{6}$$

$$= \int p(y_{\text{new}}|x_{\text{new}}, \theta^*, W) p(W|\mathcal{D}, \theta^*) dW \tag{7}$$

2.3 Case study

Let us consider the case $\{(x_i, y_i, b_i)\}_{i=1}^N$ where x_i is a data pint, y_i is its label and b_i is a binary indicator indicating whether x_i is OOD or not. It seems the trick is to predict y, b given x rather than y given x (because then predicting b given the features learnt when predicting $y|x$ is not possible).

3 The authors solution

The authors show that BNN and NLM are not good at dealing with OOD uncertainty because the decision boundaries would need to bound the data. This is not encouraged when training BNN or NLM.

BaCOUn is a framework of training (that is here applied to NLM, but could also be applied to BNN), that starts by generating an $n+1$th class of boundary points.

In details, the solution proposed by the authors of BaCOUn: Bayesian Classifiers with Out-of-Distribution Uncertainty is to add another class of points that lie at the boundary of the data. Then, to classify the augmented data (the n original classes and the $n+1$ boundary class, decision boundaries that properly bound the original data will have to be learnt).

The authors show that the OOD uncertainty estimates are comparable to the estimates produced by GPs.

3.1 BaCOUn

To reiterate, the goal of BaCOUn is to use NLM that learn decision boundaries that properly bound the data. This is done by adding OOD samples at the boundary of the data (the $n+1$th class). The classifier is then trained to distinguish between the training data and OOD points. Using the features learnt by this classifier, we fit a Bayesian logistic regression model on these features.

The method proposed to generate OOD points was to use normalizing flows. In the simpler examples (data lying in an ambient space of dimensions 2 and 3), the OOD points were generated directly by the authors because they could visualize what the OOD points should be. Normalizing flows involve a latent space, which is easier to work with. The OOD points in the latent space correspond to boundary points of the latent space. They are then sent back to the original space.

4 Conclusions: summary of experiments and results

The authors performed experiments (on MNIST, a wine dataset, and on synthetic data: the moons datasets, as well as the Gaussian mixture model). BaCOUn is able to capture OOD uncertainty when NLM and BNN cannot.

Furthermore, BaCOUn is able to provide a decomposition of the uncertainty into epistemic uncertainty and aleatoric uncertainty. It was checked on real data-sets to show that this decomposition is interpretable. For other techniques, this is not the case. Even the gold standard GPs (for giving correct OOD uncertainties) are not able to distinguish between epistemic and aleatoric uncertainty.