Uncertainty Quantification Notes

Noah Silverberg

1 Evidential Regression

Idea adapted from [2], which was UQ for classification networks. Then some papers extended it to regression, such as [1].

1.1 Background/Theory

1.1.1 Model Output and Interpretation

Let's say we want a normal distribution as our output for the brightness of a single pixel y, i.e.,

$$p(y \mid x) = \mathcal{N}(y \mid \mu, \sigma^2)$$

and we want to place a Gaussian prior over μ (we'd ideally like pixel values to be in the range $[0, 1]$, but this will be a good approximation) and an inverse gamma prior over σ^2 (we need to ensure $\sigma^2 > 0$):

$$\mu \sim \mathcal{N}(\gamma, \sigma^2/\nu), \quad \sigma^2 \sim \Gamma^{-1}(\alpha, \beta)$$

Then we can write the joint prior over the outputs as:

$$p(\mu, \sigma^2 \mid \gamma, \nu, \alpha, \beta) = \mathcal{N}(\mu \mid \gamma, \sigma^2/\nu) \cdot \Gamma^{-1}(\sigma^2 \mid \alpha, \beta) = \text{N-}\Gamma^{-1}(\mu, \sigma^2 \mid \gamma, \nu, \alpha, \beta)$$

where N-Γ^{-1} is the normal-inverse-gamma distribution.

So we can write the distribution over the pixel brightness as:

$$\begin{aligned}
&p(y \mid \gamma, \nu, \alpha, \beta) \\
&= \int_0^\infty \int_{-\infty}^\infty p(y \mid \mu, \sigma^2) p(\mu, \sigma^2 \mid \gamma, \nu, \alpha, \beta) \, \mathrm{d}\mu \, \mathrm{d}\sigma^2 \\
&= \int_0^\infty \int_{-\infty}^\infty \mathcal{N}(y \mid \mu, \sigma^2) \text{N-}\Gamma^{-1}(\mu, \sigma^2 \mid \gamma, \nu, \alpha, \beta) \, \mathrm{d}\mu \, \mathrm{d}\sigma^2 \\
&= \int_0^\infty \int_{-\infty}^\infty \mathcal{N}(y \mid \mu, \sigma^2) \cdot \mathcal{N}(\mu \mid \gamma, \sigma^2/\nu) \cdot \Gamma^{-1}(\sigma^2 \mid \alpha, \beta) \, \mathrm{d}\mu \, \mathrm{d}\sigma^2 \\
&= \int_0^\infty \Gamma^{-1}(\sigma^2 \mid \alpha, \beta) \left(\int_{-\infty}^\infty \mathcal{N}(y \mid \mu, \sigma^2) \cdot \mathcal{N}(\mu \mid \gamma, \sigma^2/\nu) \, \mathrm{d}\mu \right) \mathrm{d}\sigma^2
\end{aligned}$$

The inner integral is a pain to derive, but it can be shown that:

$$\int_{-\infty}^{\infty} \mathcal{N}(y \mid \mu, \sigma^2) \cdot \mathcal{N}(\mu \mid \gamma, \sigma^2/\nu) \, \mathrm{d}\mu = \mathcal{N}(y \mid \gamma, (1 + 1/\nu)\sigma^2)$$

So we can write the distribution over the pixel brightness as:

$$p(y \mid \gamma, \nu, \alpha, \beta) = \int_0^{\infty} \Gamma^{-1}(\sigma^2 \mid \alpha, \beta) \cdot \mathcal{N}(y \mid \gamma, (1 + 1/\nu)\sigma^2) \, \mathrm{d}\sigma^2$$

This is also annoying to derive, but it can be shown that

$$p(y \mid \gamma, \nu, \alpha, \beta) = \text{Student-}t_{2\alpha}\left(y \mid \text{loc} = \gamma, \text{scale}^2 = \frac{\beta(\nu + 1)}{\alpha\nu}\right)$$

where Student-$t_{2\alpha}$ is the Student's t distribution with 2α degrees of freedom.

So we get a prediction along with measures of both aleatoric and epistemic uncertainty [1]:

$$\underbrace{\mathbb{E}[\mu] = \gamma}_{\text{prediction}}, \quad \underbrace{\mathbb{E}[\sigma^2] = \frac{\beta}{\alpha - 1}}_{\text{aleatoric}}, \quad \underbrace{\text{Var}[\mu] = \frac{\beta}{\nu(\alpha - 1)}}_{\text{epistemic}}$$

1.1.2 Conjugate Priors

Let's say we get a bunch of samples $y_1, y_2, \ldots, y_n$, and we want to update our beliefs about μ and σ^2 given this new data. By Bayes' theorem:

$$\begin{aligned} p(\mu, \sigma^2 \mid y_1, y_2, \ldots, y_n) &\propto p(\mu, \sigma^2) \prod_{i=1}^{n} p(y_i \mid \mu, \sigma^2) \\ &= \text{N-}\Gamma^{-1}(\mu, \sigma^2 \mid \gamma, \nu, \alpha, \beta) \cdot \prod_{i=1}^{n} \mathcal{N}(y_i \mid \mu, \sigma^2) \\ &= \Gamma^{-1}(\sigma^2 \mid \alpha, \beta) \cdot \mathcal{N}(\mu \mid \gamma, \sigma^2/\nu) \cdot \prod_{i=1}^{n} \mathcal{N}(y_i \mid \mu, \sigma^2) \end{aligned}$$

Note:

$$\prod_{i=1}^{n} \mathcal{N}(y_i \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \mu)^2\right)$$

and

$$\Gamma^{-1}(\sigma^2 \mid \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} (\sigma^2)^{-\alpha - 1} \exp\left(-\frac{\beta}{\sigma^2}\right)$$

and

$$\mathcal{N}(\mu \mid \gamma, \sigma^2/\nu) = \frac{1}{\sqrt{2\pi\sigma^2/\nu}} \exp\left(-\frac{\nu}{2\sigma^2}(\mu - \gamma)^2\right)$$

So we can write (omitting intermediate steps):

$$p(\mu, \sigma^2 \mid \mathbf{y}) = \mathcal{N}(\mu \mid \tilde{\gamma}, \sigma^2/\tilde{\nu}) \cdot \Gamma^{-1}(\sigma^2 \mid \tilde{\alpha}, \tilde{\beta}) = \text{N-}\Gamma^{-1}(\mu, \sigma^2 \mid \tilde{\gamma}, \tilde{\nu}, \tilde{\alpha}, \tilde{\beta})$$

where

$$\begin{aligned}
\tilde{\gamma} &= \frac{\nu\gamma + \sum_{i=1}^{n} y_i}{\nu + n} \\
\tilde{\nu} &= \nu + n \\
\tilde{\alpha} &= \alpha + \frac{n}{2} \\
\tilde{\beta} &= \beta + \frac{1}{2}\sum_{i=1}^{n}(y_i - \bar{y})^2 + \frac{\nu n}{2(\nu + n)}\left(\gamma - \bar{y}\right)^2
\end{aligned}$$

So the posterior is also a normal-inverse-gamma distribution, which is nice because we can use the same model architecture for training and inference. This is called a conjugate prior.

1.2 Training

We need the model to now output the parameters of the prior distribution, i.e., γ, ν, α, and β, at each pixel. We will want our loss to be the negative log-likelihood, which is:

$$\mathcal{L}_{\text{NLL}}(\gamma, \nu, \alpha, \beta \mid y_1, y_2, \ldots, y_n) = -\log p(y_1, y_2, \ldots, y_n \mid \gamma, \nu, \alpha, \beta)$$

This can be computed as:

$$\begin{aligned}
&\mathcal{L}_{\text{NLL}}(\gamma, \nu, \alpha, \beta \mid y) \\
&= \frac{1}{2}\log\left(\frac{\pi}{\nu}\right) - \alpha\log(\Omega) + \left(\alpha + \frac{1}{2}\right)\log((y-\gamma)^2\nu + \Omega) + \log\left(\frac{\Gamma(\alpha)}{\Gamma\left(\alpha + \frac{1}{2}\right)}\right)
\end{aligned}$$

where $\Omega = 2\beta(1+\nu)$. The model may become overconfident by driving $\alpha, \nu \to \infty$, so we can add a regularization term to the loss:

$$\mathcal{L}_{\text{reg}}(\gamma, \nu, \alpha, \beta \mid y) = |y - \gamma| \cdot (2\nu + \alpha)$$

to penalize high confidence when the prediction is far from the mean.

So the total loss is:

$$\mathcal{L}(\gamma, \nu, \alpha, \beta \mid y) = \mathcal{L}_{\text{NLL}}(\gamma, \nu, \alpha, \beta \mid y) + \lambda\mathcal{L}_{\text{reg}}(\gamma, \nu, \alpha, \beta \mid y)$$

1.3 Pros/Cons

1.3.1 Pros

- Requires little change from our current DDCNN architecuture. The only differences are (1) we need to change the loss function, and (2) we need to output 4 channels instead of 1.

- We only need to do one forward pass to get the prediction and uncertainty estimates.
- We only need to train one model.
- Provides a measure of aleatoric and epistemic uncertainty [1].

1.3.2 Cons

- We don't necessarily know if the current size of the DDCNN would be sufficient to learn 4 parameters per pixel, it might require a larger model.
- It isn't a super popular method (although there are some papers that use it).
- It is not Bayesian, but I don't think that really matters to us.
- It doesn't necessarily know how to handle out-of-distribution data (this is kind of important actually since the uncertainty values we get from it will only really be valid if the data is similar to the training data).

2 MC Dropout

[[TODO – general idea is that this requires minimal changes to DDCNN architecture (just need to turn on dropout during training and inference), but the main issue is that it takes multiple forward passes to get the uncertainty estimates, which is not practical in a clinical setting. This does seem to have good performance empirically, though.]]

3 Bayesian By Backprop (BBB)

[[TODO – general idea is that this requires a lot of changes to the DDCNN architecuture, but it is fully Bayesian. Similar to MC Dropout, though, we need to do multiple forward passes to get the uncertainty estimates, which is not practical in a clinical setting. Also it doesn't seem to have super good performance from what I've read.]]

4 Deep Ensembles

[[TODO – this requires no changes at all which is really nice. But it suffers again from the same issue as MCD and BBB which is multiple forward passes. However this emprirically does have good results.]]

References

[1] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression, 2020.

[2] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty, 2018.