Uncertainty quantification

Brooks Paige

COMP0171 Week 7

Predictive uncertainty

Taking a probabilistic view, when we fit a deep learning model to data $\mathcal{D} = \{\mathbf{x}_{1:N}, y_{1:N}\}$ we

- Define a network architecture $f_{\boldsymbol{\theta}}(\mathbf{x}_i)$

Predictive uncertainty

Taking a probabilistic view, when we fit a deep learning model to data $\mathcal{D} = \{\mathbf{x}_{1:N}, y_{1:N}\}$ we

- Define a network architecture $f_{\boldsymbol{\theta}}(\mathbf{x}_i)$
- Define a **likelihood** $p(y_i|\mathbf{x}_i, \boldsymbol{\theta})$

Predictive uncertainty

Taking a probabilistic view, when we fit a deep learning model to data $\mathcal{D} = \{\mathbf{x}_{1:N}, y_{1:N}\}$ we

- Define a network architecture $f_{\boldsymbol{\theta}}(\mathbf{x}_i)$
- Define a **likelihood** $p(y_i|\mathbf{x}_i, \boldsymbol{\theta})$ *[e.g.* $\mathrm{Categorical}(y_i|f_{\boldsymbol{\theta}}(\mathbf{x}_i))$ *]*

Predictive uncertainty

Taking a probabilistic view, when we fit a deep learning model to data $\mathcal{D} = \{\mathbf{x}_{1:N}, y_{1:N}\}$ we

- Define a network architecture $f_{\boldsymbol{\theta}}(\mathbf{x}_i)$
- Define a **likelihood** $p(y_i|\mathbf{x}_i, \boldsymbol{\theta})$ *[e.g.* $\mathrm{Categorical}(y_i|f_{\boldsymbol{\theta}}(\mathbf{x}_i))$ *]*
- Run a stochastic gradient descent algorithm to find

$$\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta})$$

We then can make predictions for a new test point $\mathbf{x}$ by using the learned distribution

$$p(y|\mathbf{x}, \boldsymbol{\theta}^\star) = \mathrm{Categorical}(y|f_{\boldsymbol{\theta}^\star}(\mathbf{x})).$$

What are these probabilities?

$$p(y|\mathbf{x}, \boldsymbol{\theta}^\star) = \mathrm{Categorical}(y|f_{\boldsymbol{\theta}^\star}(\mathbf{x}))$$

For a **100-class classification problem**, suppose $\hat{\mathbf{y}} = f_{\boldsymbol{\theta}^\star}(\mathbf{x}) \in [0,1]^{100}$, with $\sum_{k=1}^{100} \hat{y}_k = 1$.

What are these probabilities?

$$p(y|\mathbf{x}, \boldsymbol{\theta}^\star) = \mathrm{Categorical}(y|f_{\boldsymbol{\theta}^\star}(\mathbf{x}))$$

For a **100-class classification problem**, suppose $\hat{\mathbf{y}} = f_{\boldsymbol{\theta}^\star}(\mathbf{x}) \in [0,1]^{100}$, with $\sum_{k=1}^{100} \hat{y}_k = 1$.

In this sense, $\hat{y}_k = \Pr[y = k]$ — the probability that the outcome of the random event "assign a label to $\mathbf{x}$" would assign the label k.

What are these probabilities?

$$p(y|\mathbf{x}, \boldsymbol{\theta}^\star) = \mathrm{Categorical}(y|f_{\boldsymbol{\theta}^\star}(\mathbf{x}))$$

For a **100-class classification problem**, suppose $\hat{\mathbf{y}} = f_{\boldsymbol{\theta}^\star}(\mathbf{x}) \in [0,1]^{100}$, with $\sum_{k=1}^{100} \hat{y}_k = 1$.

In this sense, $\hat{y}_k = \Pr[y = k]$ — the probability that the outcome of the random event "assign a label to $\mathbf{x}$" would assign the label k.

Think back to our definitions of probability. What should happen, then, if we repeat this "random event" many times, across many test points?

Uncertainty calibration

For a set of validation / test points $\mathbf{x}_{1:N}, y_{1:N}$, we can compute predicted class labels $\hat{y}_i$ and their corresponding probabilities $\hat{p}_i$.

Intuitively, a **well-calibrated** classifier would get the "correct" label in proportion to the probability $\hat{p}$:

$$\Pr(\hat{y} = y | \hat{p} = p) = p$$

for all $p \in [0, 1]$.

Uncertainty calibration

For a set of validation / test points $\mathbf{x}_{1:N}, y_{1:N}$, we can compute predicted class labels $\hat{y}_i$ and their corresponding probabilities $\hat{p}_i$.

Intuitively, a **well-calibrated** classifier would get the "correct" label in proportion to the probability $\hat{p}$:

$$\Pr(\hat{y} = y | \hat{p} = p) = p$$

for all $p \in [0, 1]$.

Example: consider all test points i which the classifier had $\hat{p}_i \in [0.3, 0.4]$. We would expect between 30% and 40% of those points to have been classified correctly.

Estimating uncertainty calibration

To make this concrete: define M consecutive intervals $I_m = (\frac{m-1}{M}, \frac{m}{M}]$ which partition $[0,1]$, and let $B_m = \{i : \hat{p}_i \in I_m\}$, i.e. the bin of indices whose probabilities fall in the interval I_m. Then:

- The **accuracy** $\mathrm{acc}(B_m)$ is $\frac{1}{|B_m|}\sum_{i \in B_m} \mathbb{I}[\hat{y}_i = y_i]$.
- The average **confidence** $\mathrm{conf}(B_m)$ is $\frac{1}{|B_m|}\sum_{i \in B_m} \hat{p}_i$.

A well calibrated model would have $\mathrm{acc}(B_m) = \mathrm{conf}(B_m)$ for each m.

Estimating uncertainty calibration

To make this concrete: define M consecutive intervals $I_m = (\frac{m-1}{M}, \frac{m}{M}]$ which partition $[0,1]$, and let $B_m = \{i : \hat{p}_i \in I_m\}$, i.e. the bin of indices whose probabilities fall in the interval I_m. Then:

- The **accuracy** $\mathrm{acc}(B_m)$ is $\frac{1}{|B_m|}\sum_{i\in B_m} \mathbb{I}[\hat{y}_i = y_i]$.
- The average **confidence** $\mathrm{conf}(B_m)$ is $\frac{1}{|B_m|}\sum_{i\in B_m} \hat{p}_i$.

A well calibrated model would have $\mathrm{acc}(B_m) = \mathrm{conf}(B_m)$ for each m.

Since some bins may have more examples than other, a useful statistic is the **expected calibration error**,

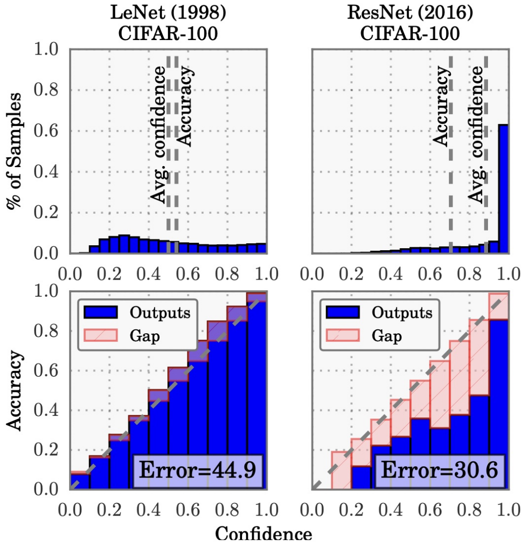
$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} \Big| \mathrm{acc}(B_m) - \mathrm{conf}(B_m) \Big|.$$

Deep network uncertainty calibration

Modern deep networks are **calibrated worse**.

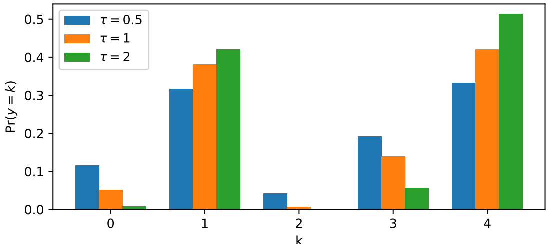
- **LeNet** is a 5-layer convolutional network
- **ResNet** is a 110-layer resnet

Observation: recent deeper networks have higher accuracy, but are more overconfident.



Temperature scaling

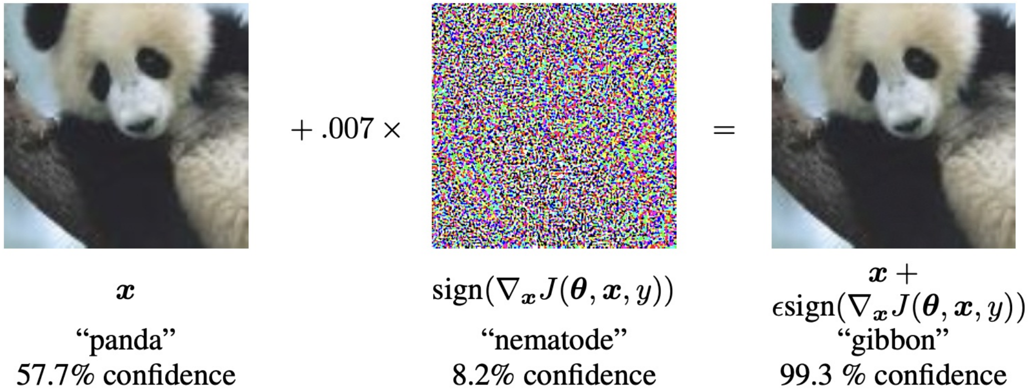
An easy "trick" to fix calibration error in classifiers is to scale the network outputs before the softmax.



This plot shows $\mathrm{softmax}(\tau \mathbf{z})$ for a fixed value $\mathbf{z}$ and a few different τ.

Adjusting the scalar τ (based on a validation set) won't change class predictions, but will affect uncertainties.

Adversarial perturbations


$+ .007 \times$
$=$
$\boldsymbol{x}$
"panda"
57.7% confidence
$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

[Szegedy et al., 2014]

Adversarial perturbations

There is really but one thing to say about **this** sorry movie It should never have been made The first one one of my favourites An American Werewolf in London is a great movie with a good plot good actors and good FX But this one It stinks to heaven with a cry of helplessness

There is really but one thing to say about **that** sorry movie It should never have been made The first one one of my favourites An American Werewolf in London is a great movie with a good plot good actors and good FX But this one It stinks to heaven with a cry of helplessness

The left example correctly is predicted to have "negative" sentiment.

Changing one word, on the right, changes the prediction to "positive".

[Sato et al., 2018]

Uncertainty in Bayesian deep learning

Now define a prior $p(\boldsymbol{\theta})$. Optimizing parameters $\boldsymbol{\theta}$ would find

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}),$$

and make predictions using the distribution $p(y|\mathbf{x}, \boldsymbol{\theta}^\star)$.

Uncertainty in Bayesian deep learning

Now define a prior $p(\boldsymbol{\theta})$. Optimizing parameters $\boldsymbol{\theta}$ would find

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} \log p(y_i|\mathbf{x}_i, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}),$$

and make predictions using the distribution $p(y|\mathbf{x}, \boldsymbol{\theta}^\star)$.

If we use a Bayesian approach and **marginalize** over the parameters $\boldsymbol{\theta}$, to average across uncertainty in the parameters, we will make predictions with

$$p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta},$$

effectively decomposing our predictive distribution uncertainty into uncertainty over $\boldsymbol{\theta}$ and uncertainty over $y|\boldsymbol{\theta}$.

Uncertainty quantification

$$p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \mathcal{M})p(\mathcal{M}|\mathcal{D})d\mathcal{M}$$

- **Aleatoric uncertainty**: $p(y|\mathbf{x}, \mathcal{M})$ captures irreducible uncertainty in the measurement process
- **Epistemic uncertainty**: $p(\mathcal{M}|\mathcal{D})$ captures our uncertainty about the correct "model" given limited data
  - Represents our uncertainty over multiple different hypothetical models $\mathcal{M}$
  - More data means less epistemic uncertainty

Uncertainty quantification

$$p(y|\mathbf{x}, \mathcal{D}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta}$$

- **Aleatoric uncertainty**: $p(y|\mathbf{x}, \boldsymbol{\theta})$ captures irreducible uncertainty in the measurement process
- **Epistemic uncertainty**: $p(\boldsymbol{\theta}|\mathcal{D})$ captures our uncertainty about the correct "model" given limited data
  - Represents our uncertainty over multiple different hypothetical values of $\boldsymbol{\theta}$
  - More data means less epistemic uncertainty

Hypothetical example

- Suppose we have a binary classification problem, and a test data point $\mathbf{x}$.
- We could have large amounts of uncertainty in $p(y|\mathbf{x}, \mathcal{D})$ in two distinct ways:

Hypothetical example

- Suppose we have a binary classification problem, and a test data point $\mathbf{x}$.
- We could have large amounts of uncertainty in $p(y|\mathbf{x}, \mathcal{D})$ in two distinct ways:
  1. The posterior over $\boldsymbol{\theta}$ is concentrated around a particular value $\boldsymbol{\theta}^{\star}$, and $p(y|\mathbf{x}, \boldsymbol{\theta}^{\star}) \approx 0.5$

Hypothetical example

- Suppose we have a binary classification problem, and a test data point $\mathbf{x}$.
- We could have large amounts of uncertainty in $p(y|\mathbf{x}, \mathcal{D})$ in two distinct ways:
  1. The posterior over $\boldsymbol{\theta}$ is concentrated around a particular value $\boldsymbol{\theta}^\star$, and $p(y|\mathbf{x}, \boldsymbol{\theta}^\star) \approx 0.5$
  2. The posterior over $\boldsymbol{\theta}$ places high probability over two values $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, with

$$p(y|\mathbf{x}, \boldsymbol{\theta}_1) \approx 1$$
$$p(y|\mathbf{x}, \boldsymbol{\theta}_2) \approx 0$$

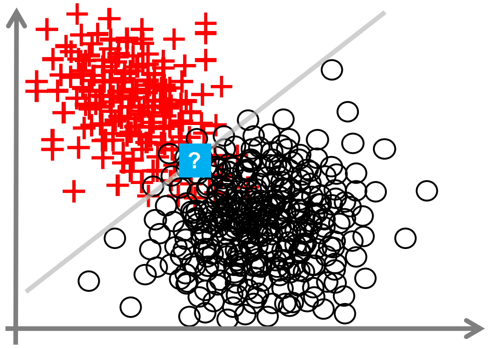
Hypothetical example

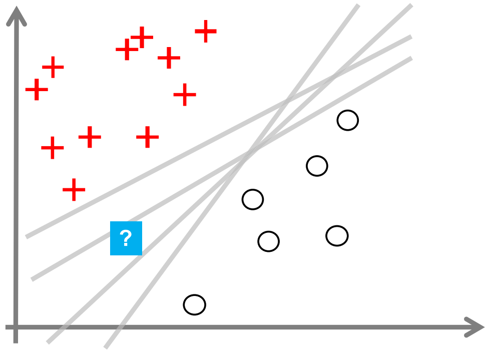
- Suppose we have a binary classification problem, and a test data point $\mathbf{x}$.
- We could have large amounts of uncertainty in $p(y|\mathbf{x}, \mathcal{D})$ in two distinct ways:
  1. The posterior over $\boldsymbol{\theta}$ is concentrated around a particular value $\boldsymbol{\theta}^\star$, and $p(y|\mathbf{x}, \boldsymbol{\theta}^\star) \approx 0.5$
  2. The posterior over $\boldsymbol{\theta}$ places high probability over two values $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, with

$$\begin{aligned} p(y|\mathbf{x}, \boldsymbol{\theta}_1) &\approx 1 \\ p(y|\mathbf{x}, \boldsymbol{\theta}_2) &\approx 0 \end{aligned}$$

- In one of these cases, collecting more data might resolve the ambiguity. In the other, it's just label noise!

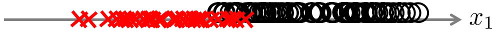
Epistemic vs Aleatoric Uncertainty

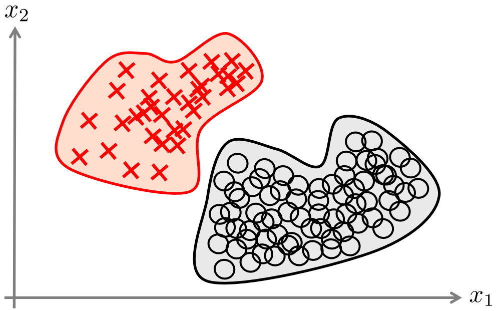




[Hüllermeier & Waegeman, 2021]

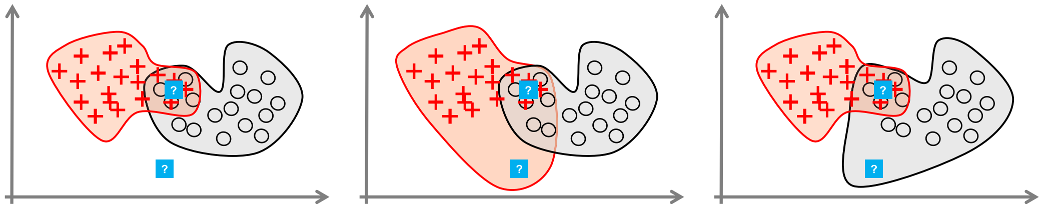
Epistemic vs Aleatoric Uncertainty





Sometimes, aleatoric uncertainty is indicative of missing features.

Epistemic vs Aleatoric Uncertainty



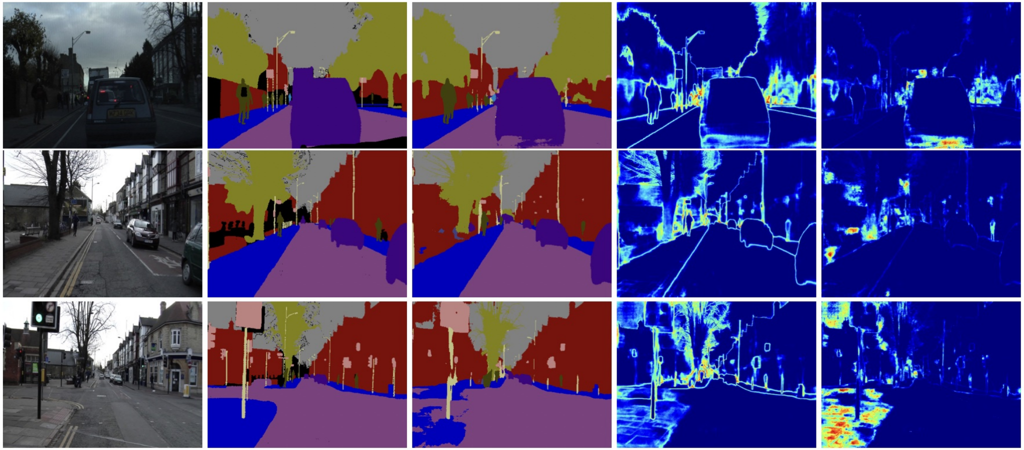
With high-complexity models, points far away from training data could be any class.

[Hüllermeier & Waegeman, 2021]

Accurate uncertainty estimates are useful

- Essential for many downstream tasks (which may rely on epistemic uncertainty):
    - Bayesian optimization
    - Out-of-distribution detection
    - Active learning
- Useful whenever using a model to make a **real-world decision, with varying costs** to different failures

Epistemic vs Aleatoric Uncertainty


(a) Input Image (b) Ground Truth (c) Semantic Segmentation (d) Aleatoric Uncertainty (e) Epistemic Uncertainty

[Kendal & Gal, 2017]

How do we actually compute this?

Generally the idea is to **decompose the predictive variance**. This is more clear in a regression setting:

Suppose we have a network $\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x})$ which outputs $\mathbf{z} = \{\boldsymbol{\mu}, \sigma\}$ that define a likelihood $\mathcal{N}(\mathbf{y}|\boldsymbol{\mu}, \sigma^2\mathbf{I})$ over targets $\mathbf{y}$.

How do we actually compute this?

Generally the idea is to **decompose the predictive variance**. This is more clear in a regression setting:

Suppose we have a network $\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x})$ which outputs $\mathbf{z} = \{\boldsymbol{\mu}, \sigma\}$ that define a likelihood $\mathcal{N}(\mathbf{y}|\boldsymbol{\mu}, \sigma^2\mathbf{I})$ over targets $\mathbf{y}$.

$$\mathrm{Var}(\mathbf{y}) = \mathrm{Var}[\mathbb{E}[\mathbf{y}|\mathbf{z}]] + \mathbb{E}[\mathrm{Var}[\mathbf{y}|\mathbf{z}]]$$

How do we actually compute this?

Generally the idea is to **decompose the predictive variance**. This is more clear in a regression setting:

Suppose we have a network $\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x})$ which outputs $\mathbf{z} = \{\boldsymbol{\mu}, \sigma\}$ that define a likelihood $\mathcal{N}(\mathbf{y}|\boldsymbol{\mu}, \sigma^2\mathbf{I})$ over targets $\mathbf{y}$.

$$\begin{aligned}\mathrm{Var}(\mathbf{y}) &= \mathrm{Var}[\mathbb{E}[\mathbf{y}|\mathbf{z}]] + \mathbb{E}[\mathrm{Var}[\mathbf{y}|\mathbf{z}]] \\ &= \underbrace{\mathrm{Var}[\boldsymbol{\mu}]}_{\text{epistemic}} + \underbrace{\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\sigma^2\mathbf{I}]}_{\text{aleatoric}}.\end{aligned}$$

How do we actually compute this?

Generally the idea is to **decompose the predictive variance**. This is more clear in a regression setting:

Suppose we have a network $\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x})$ which outputs $\mathbf{z} = \{\boldsymbol{\mu}, \sigma\}$ that define a likelihood $\mathcal{N}(\mathbf{y}|\boldsymbol{\mu}, \sigma^2\mathbf{I})$ over targets $\mathbf{y}$.

$$\begin{aligned}\mathrm{Var}(\mathbf{y}) &= \mathrm{Var}[\mathbb{E}[\mathbf{y}|\mathbf{z}]] + \mathbb{E}[\mathrm{Var}[\mathbf{y}|\mathbf{z}]] \\ &= \underbrace{\mathrm{Var}[\boldsymbol{\mu}]}_{\text{epistemic}} + \underbrace{\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\sigma^2\mathbf{I}]}_{\text{aleatoric}} .\end{aligned}$$

Here the first line is the **law of total variance**, and always holds; the second line is model-specific. Generally we will construct Monte Carlo estimates for both those terms.

How do we actually compute this?

Generally the idea is to **decompose the predictive variance**. This is more clear in a regression setting:

Suppose we have a network $\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x})$ which outputs $\mathbf{z} = \{\boldsymbol{\mu}, \sigma\}$ that define a likelihood $\mathcal{N}(\mathbf{y}|\boldsymbol{\mu}, \sigma^2\mathbf{I})$ over targets $\mathbf{y}$.

$$\begin{aligned}\mathrm{Var}(\mathbf{y}) &= \mathrm{Var}[\mathbb{E}[\mathbf{y}|\mathbf{z}]] + \mathbb{E}[\mathrm{Var}[\mathbf{y}|\mathbf{z}]] \\ &= \underbrace{\mathrm{Var}[\boldsymbol{\mu}]}_{\text{epistemic}} + \underbrace{\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\sigma^2\mathbf{I}]}_{\text{aleatoric}}.\end{aligned}$$

Here the first line is the **law of total variance**, and always holds; the second line is model-specific. Generally we will construct Monte Carlo estimates for both those terms.

For classification likelihoods, the “aleatoric” term is a bit more complicated to write out, and there are various estimators.

Summary

- Calibrated uncertainty (and understanding sources of uncertainty) is probably necessary for safe deployment of real-world ML systems
- Calibration (and mis-calibration) can be measured, and there are some simple "tricks" to calibrate models after they are trained
- Uncertainty can be conceptually decomposed into aleatoric and epistemic uncertainty
- In Bayesian models it can be (relatively) easy to isolate these contributions to predictive uncertainty