Motivation

Why model uncertainty?

- **Confidence calibration**: know how reliable a prediction is (e.g. so it can ask a human for clarification)
- **Regularization**: prevent your model from overfitting
- **Ensembling**: smooth your predictions by averaging them over multiple possible models
- **Model selection**: decide which of multiple plausible models best describes the data
- **Sparsification**: drop connections, encode them with fewer bits
- **Exploration**: decide which training examples are worth labeling (active learning), optimize an expensive black-box function (Bayesian optimization), estimating rewards from multi-armed bandits (reinforcement learning)
- **Robustness**: make good predictions when the data is either naturally perturbed or explicitly modified by an adversary
Motivating example: estimating the parameter of a biased coin
- You flip a coin 100 times. It lands heads $N_H = 55$ times and tails $N_T = 45$ times.
- What is the probability it will come up heads if we flip again?

Model: observations $x_i$ are independent and identically distributed (i.i.d.) Bernoulli random variables with parameter $\theta$.

The likelihood function is the probability of the observed data (the entire sequence of H’s and T’s) as a function of $\theta$: 

\[
L(\theta) = p(D) = \prod_{i=1}^{N} \theta^{x_i} (1 - \theta)^{1-x_i} = \theta^{N_H} (1 - \theta)^{N_T}
\]

$N_H$ and $N_T$ are sufficient statistics.
Bayesian Parameter Estimation: A Toy Example

- The likelihood is generally very small, so it’s often convenient to work with log-likelihoods.

\[ L(\theta) = \theta^{N_H}(1 - \theta)^{N_T} \]

\[ \ell(\theta) = \log L(\theta) = N_H \log \theta + N_T \log(1 - \theta) \]

\[ \approx 7.9 \times 10^{-31} \]

\[ \approx -69.31 \]
A Toy Example

- Good values of $\theta$ should assign high probability to the observed data. This motivates the maximum likelihood criterion.

- Solve by setting derivatives to zero:

  
  $$
  \frac{d \ell}{d \theta} = \frac{d}{d \theta} \left( N_H \log \theta + N_T \log(1 - \theta) \right) 
  = \frac{N_H}{\theta} - \frac{N_T}{1 - \theta}
  $$

- Setting this to zero gives the maximum likelihood estimate:

  $$
  \hat{\theta}_{ML} = \frac{N_H}{N_H + N_T},
  $$

- Normally there’s no analytic solution, and we need to solve an optimization problem (e.g. using gradient descent).
Bayesian Parameter Estimation: A Toy Example

• Maximum likelihood has a pitfall: if you have too little data, it can overfit.
• E.g., what if you flip the coin twice and get H both times?

\[ \theta_{\text{ML}} = \frac{N_H}{N_H + N_T} = \frac{2}{2 + 0} = 1 \]

• But even a fair coin has 25% chance of showing this result.
• Because it never observed T, it assigns this outcome probability 0. This problem is known as data sparsity.
• If you observe a single T in the test set, the likelihood is \(-\infty\).
A Toy Example

- In maximum likelihood, the observations are treated as random variables, but the parameters are not.
- The **Bayesian** approach treats the parameters as random variables as well.
- To define a Bayesian model, we need to specify two distributions:
  - The *prior distribution* \( p(\theta) \), which encodes our beliefs about the parameters *before* we observe the data
  - The *likelihood* \( p(D | \theta) \), same as in maximum likelihood
- When we update our beliefs based on the observations, we compute the *posterior distribution* using Bayes’ Rule:
  \[
  p(\theta | D) = \frac{p(\theta)p(D | \theta)}{\int p(\theta')p(D | \theta') d\theta'}.
  \]
- We rarely ever compute the denominator explicitly due to intractability.
Let's revisit the coin example. We already know the likelihood:

\[ L(\theta) = p(D) = \theta^N H (1 - \theta)^N T \]

It remains to specify the prior \( p(\theta) \).

- We can choose an uninformative prior, which assumes as little as possible. A reasonable choice is the uniform prior.
- But our experience tells us 0.5 is more likely than 0.99. One particularly useful prior that lets us specify this is the beta distribution:

\[
p(\theta; a, b) = \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)} \theta^{a-1} (1 - \theta)^{b-1}.
\]

This notation for proportionality lets us ignore the normalization constant:

\[
p(\theta; a, b) \propto \theta^{a-1} (1 - \theta)^{b-1}.
\]
Bayesian Parameter Estimation: A Toy Example

Beta distribution for various values of $a$, $b$:

- Some observations:
  - The expectation $\mathbb{E}[\theta] = a/(a + b)$.
  - The distribution gets more peaked when $a$ and $b$ are large.
  - The uniform distribution is the special case where $a = b = 1$.
- The main thing the beta distribution is used for is as a prior for the Bernoulli distribution.
Computing the posterior distribution:

\[ p(\theta \mid D) \propto p(\theta)p(D \mid \theta) \]
\[ \propto \left[ \theta^{a-1}(1-\theta)^{b-1} \right] \left[ \theta^{N_H}(1-\theta)^{N_T} \right] \]
\[ = \theta^{a-1+N_H}(1-\theta)^{b-1+N_T}. \]

This is just a beta distribution with parameters \( N_H + a \) and \( N_T + b \).

The posterior expectation of \( \theta \) is:

\[ \mathbb{E}[\theta \mid D] = \frac{N_H + a}{N_H + N_T + a + b} \]

The parameters \( a \) and \( b \) of the prior can be thought of as pseudo-counts.

The reason this works is that the prior and likelihood have the same functional form. This phenomenon is known as conjugacy, and it’s very useful.
Bayesian Parameter Estimation: A Toy Example

Bayesian inference for the coin flip example:

**Small data setting**

\[ N_H = 2, \quad N_T = 0 \]

**Large data setting**

\[ N_H = 55, \quad N_T = 45 \]

When you have enough observations, the data overwhelm the prior.
Bayesian Parameter Estimation: A Toy Example

- What do we actually do with the posterior?
- The **posterior predictive distribution** is the distribution over future observables given the past observations. We compute this by marginalizing out the parameter(s):

\[
p(D' | D) = \int p(\theta | D)p(D' | \theta) d\theta.
\] (1)

- For the coin flip example:

\[
\theta_{\text{pred}} = \Pr(x' = H | D) = \int p(\theta | D)Pr(x' = H | \theta) d\theta
\]

\[
= \int \text{Beta}(\theta; N_H + a, N_T + b) \cdot \theta d\theta
\]

\[
= \mathbb{E}_{\text{Beta}(\theta; N_H + a, N_T + b)}[\theta]
\]

\[
= \frac{N_H + a}{N_H + N_T + a + b},
\] (2)
Bayesian Parameter Estimation: A Toy Example

- **Maximum a-posteriori (MAP) estimation**: find the most likely parameter settings under the posterior

- This converts the Bayesian parameter estimation problem into a maximization problem

\[
\hat{\theta}_{\text{MAP}} = \arg \max_{\theta} \ p(\theta | D)
\]

\[
= \arg \max_{\theta} \ p(\theta, D)
\]

\[
= \arg \max_{\theta} \ p(\theta) \ p(D | \theta)
\]

\[
= \arg \max_{\theta} \ \log p(\theta) + \log p(D | \theta)
\]
Bayesian Parameter Estimation: A Toy Example

- Joint probability in the coin flip example:

\[
\log p(\theta, D) = \log p(\theta) + \log p(D | \theta)
\]

\[
= \text{const} + (a - 1) \log \theta + (b - 1) \log(1 - \theta) + N_H \log \theta + N_T \log(1 - \theta)
\]

\[
= \text{const} + (N_H + a - 1) \log \theta + (N_T + b - 1) \log(1 - \theta)
\]

- Maximize by finding a critical point

\[
0 = \frac{d}{d\theta} \log p(\theta, D) = \frac{N_H + a - 1}{\theta} - \frac{N_T + b - 1}{1 - \theta}
\]

- Solving for \( \theta \),

\[
\hat{\theta}_{MAP} = \frac{N_H + a - 1}{N_H + N_T + a + b - 2}
\]
Bayesian Parameter Estimation: A Toy Example

Comparison of estimates in the coin flip example:

<table>
<thead>
<tr>
<th>Formula</th>
<th>$N_H = 2, N_T = 0$</th>
<th>$N_H = 55, N_T = 45$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\hat{\theta}_{ML}$</td>
<td>$\frac{N_H}{N_H + N_T}$</td>
<td>$\frac{55}{100} = 0.55$</td>
</tr>
<tr>
<td>$\theta_{pred}$</td>
<td>$\frac{N_H + a}{N_H + N_T + a + b}$</td>
<td>$\frac{4}{6} \approx 0.67$</td>
</tr>
<tr>
<td>$\hat{\theta}_{MAP}$</td>
<td>$\frac{N_H + a - 1}{N_H + N_T + a + b - 2}$</td>
<td>$\frac{3}{4} = 0.75$</td>
</tr>
</tbody>
</table>

How many samples do we need for $\hat{\theta}_{ML}$ to be a good estimate of $\theta$? Use Hoeffding’s Inequality for sampling complexity bound

$$p(|\hat{\theta}_{ML} - \theta| \geq \varepsilon) \leq 2e^{-2N\varepsilon^2}$$

where $N = N_H + N_T$. 
Lessons learned

- Bayesian parameter estimation is more robust to data sparsity.
- Maximum likelihood is about optimization, while Bayesian parameter estimation is about integration.
- The Bayesian solution converges to the maximum likelihood solution as we observe more data.
Linear Regression as Maximum Likelihood

- We can give linear regression a probabilistic interpretation by assuming a Gaussian noise model:

\[ t \mid x \sim \mathcal{N}(w^\top x + b, \sigma^2) \]

- Linear regression is just maximum likelihood under this model:

\[
\frac{1}{N} \sum_{i=1}^{N} \log p(t^{(i)} \mid x^{(i)}; w, b) = \frac{1}{N} \sum_{i=1}^{N} \log \mathcal{N}(t^{(i)}; w^\top x + b, \sigma^2) \\
= \frac{1}{N} \sum_{i=1}^{N} \log \left[ \frac{1}{\sqrt{2\pi}\sigma} \exp \left( -\frac{(t^{(i)} - w^\top x - b)^2}{2\sigma^2} \right) \right] \\
= \text{const} - \frac{1}{2N\sigma^2} \sum_{i=1}^{N} (t^{(i)} - w^\top x - b)^2
\]
Bayesian linear regression considers various plausible explanations for how the data points were generated.

- It makes predictions using all possible regression weights, weighted by their posterior probability.
Bayesian Linear Regression

• Leave out the bias for simplicity

• Prior distribution: a broad, spherical (multivariate) Gaussian centered at zero:

\[ w \sim \mathcal{N}(0, \nu^2 I) \]

• Likelihood: same as in the maximum likelihood formulation:

\[ t \mid x, w \sim \mathcal{N}(w^T x, \sigma^2) \]

• Posterior:

\[ w \mid \mathcal{D} \sim \mathcal{N}(\mu, \Sigma) \]

\[ \mu = \sigma^{-2} \Sigma X^T t \]

\[ \Sigma^{-1} = \nu^{-2} I + \sigma^{-2} X^T X \]

— Bishop, Pattern Recognition and Machine Learning
Bayesian Linear Regression

Posterior predictive distribution:

\[ p(t \mid x, D) = \int p(t \mid x, w)p(w \mid D) \, dw \]

\[ = \mathcal{N}(t \mid \mu^\top x, \sigma^2_{\text{pred}}(x)) \]

\[ \sigma^2_{\text{pred}}(x) = \sigma^2 + x^\top \Sigma x, \]

where \( \mu \) and \( \Sigma \) are the posterior mean and covariance of \( \Sigma \).
Bayesian Linear Regression

- We can turn this into nonlinear regression using basis functions.

\[ \phi_j(x) = x^j \]

\[ \phi_j(x) = \exp \left( -\frac{(x - \mu_j)^2}{2s^2} \right) \]

\[ \phi_j(x) = \sigma \left( \frac{x - \mu_j}{s} \right) \]

— Bishop, Pattern Recognition and Machine Learning
Bayesian Linear Regression

— Bishop, Pattern Recognition and Machine Learning
Calibration

- **Calibration**: of the times your model predicts something with 90% confidence, is it right 90% of the time?
- **Example**: calibration of weather forecasts

The Weather Channel  
Local Weather Station
• Most of our neural nets output probability distributions, e.g. over object categories. Are these calibrated?
• While more accurate, modern neural networks are overconfident in their decisions.
Calibration

• Suppose an algorithm outputs a probability distribution over targets, and gets a loss based on this distribution and the true target.

• A scoring rule is a numerical quantization of the calibration of a predictive distribution $p(y|x)$. If the underlying true distribution over data points is denoted $q(x,y)$, then the expected scoring rule is defined as $S(p, q) = \mathbb{E}_q[S(p, (x, y))]$ for a scoring function $S(p, (x, y))$.

• A proper scoring rule is a rule which ensures that $S(p, q) \leq S(q, q)$ with equality iff $p(y|x) = q(y|x)$.

• The canonical example is negative log-likelihood (NLL). If $k$ is the category label, $t$ is the indicator vector for the label, and $y$ are the predicted probabilities,

$$L(y, t) = - \log y_k = - t^\top (\log y)$$
Calibration

- Calibration failures show up in the test NLL scores:

---

Guo et al., 2017, On calibration of modern neural networks
Guo et al. explored 7 different calibration methods, but the one that worked the best was also the simplest: **temperature scaling**.

A classification network typically predicts $\sigma(z)$, where $\sigma$ is the softmax function

$$
\sigma(z)_k = \frac{\exp(z_k)}{\sum_{k'} \exp(z_{k'})}
$$

and $z$ are called the **logits**.

They replace this with

$$
\sigma(z/T),
$$

where $T$ is a scalar called the **temperature**.

$T$ is tuned to minimize the NLL on a validation set.

Intuitively, because NLL is a proper scoring rule, the algorithm is incentivized to match the true probabilities as closely as possible.
Calibration

- Before and after temperature scaling:

![Graph showing calibration before and after temperature scaling](image)
References I

