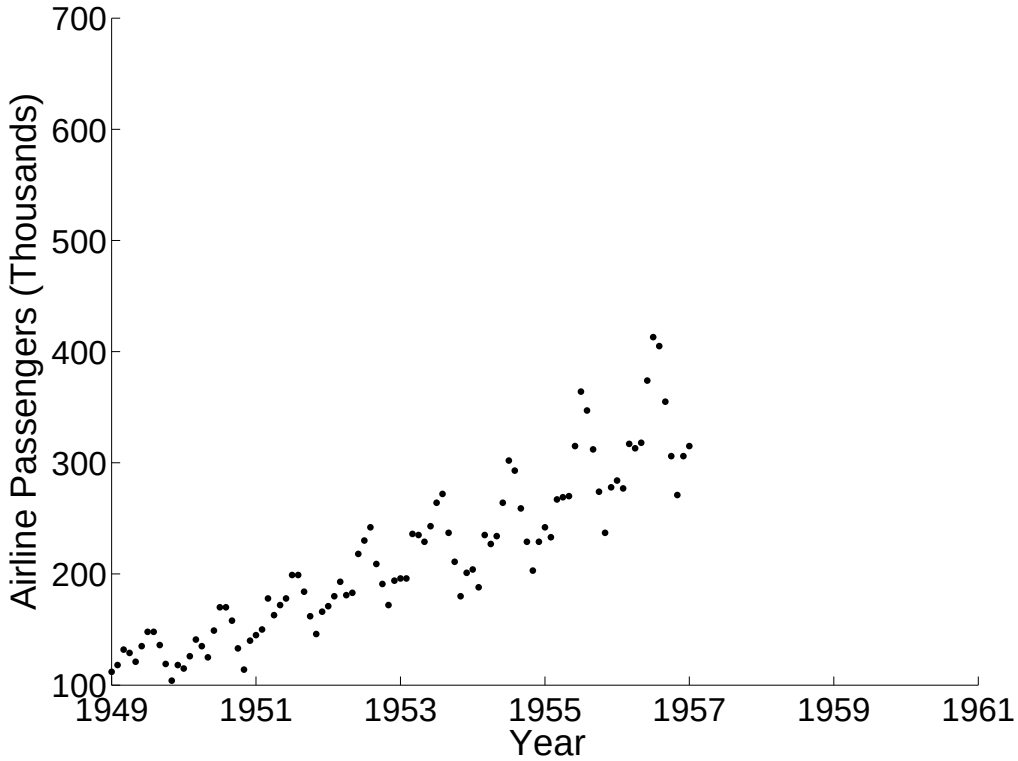
Probabilistic Model Construction

Andrew Gordon Wilson

https://cims.nyu.edu/~andrewgw
Courant Institute of Mathematical Sciences
Center for Data Science
New York University

Optimization, Big Data and Applications (OBA)
Veroli Summer School
July 4, 2022

Model Selection



Which model should we choose?

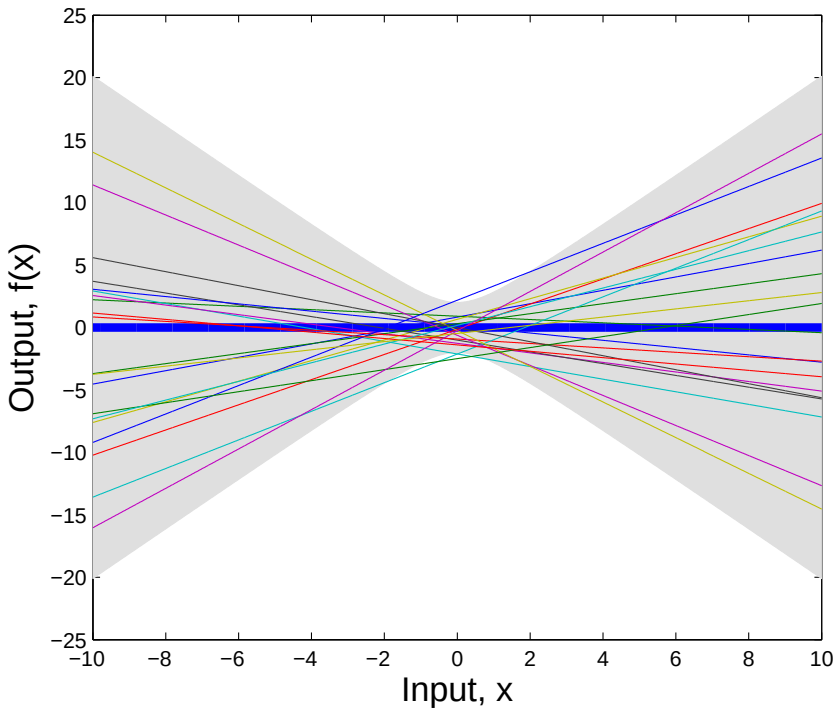
(1): $f_1(x) = w_0 + w_1 x$ **(2):** $f_2(x) = \sum_{j=0}^{3} w_j x^j$ **(3):** $f_3(x) = \sum_{j=0}^{10^4} w_j x^j$

A Function-Space View

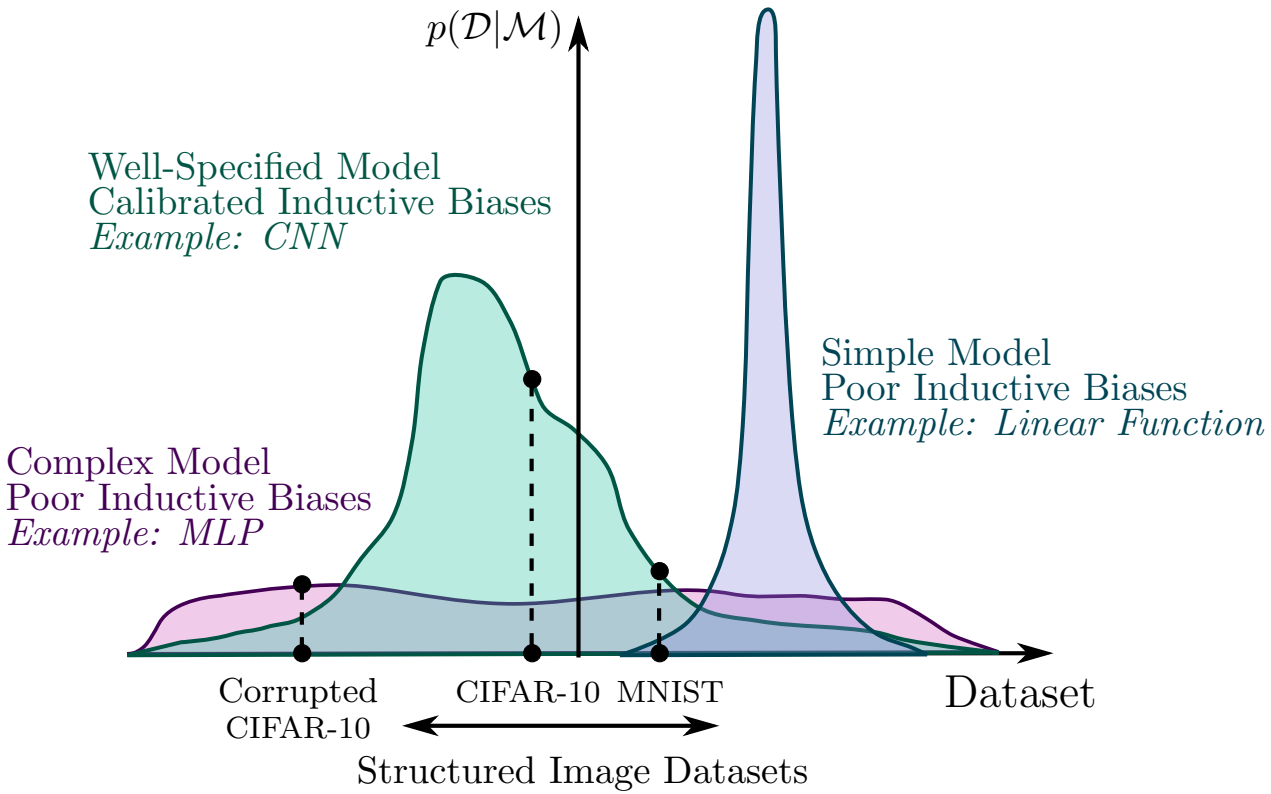
Consider the simple linear model,

$$f(x) = w_0 + w_1 x\,, \tag{1}$$

$$w_0, w_1 \sim \mathcal{N}(0, 1)\,. \tag{2}$$

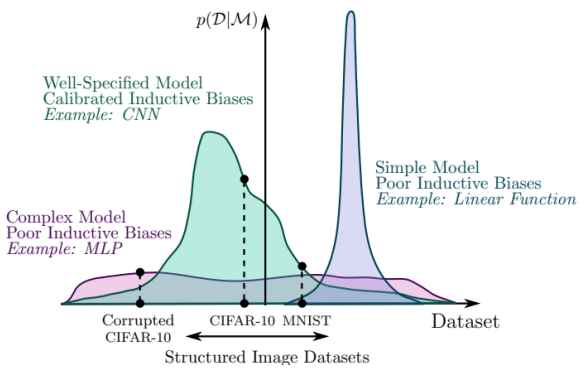


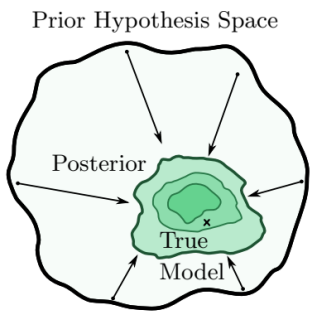
Model Construction and Generalization

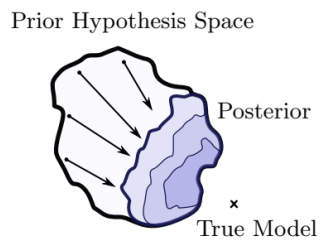


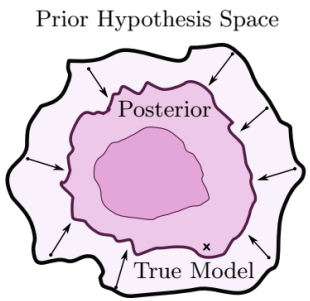
How do we learn?

- The ability for a system to learn is determined by its *support* (which solutions are a priori possible) and *inductive biases* (which solutions are a priori likely).
- We should not conflate *flexibility* and *complexity*.
- An influx of new *massive* datasets provide great opportunities to automatically learn rich statistical structure, leading to new scientific discoveries.





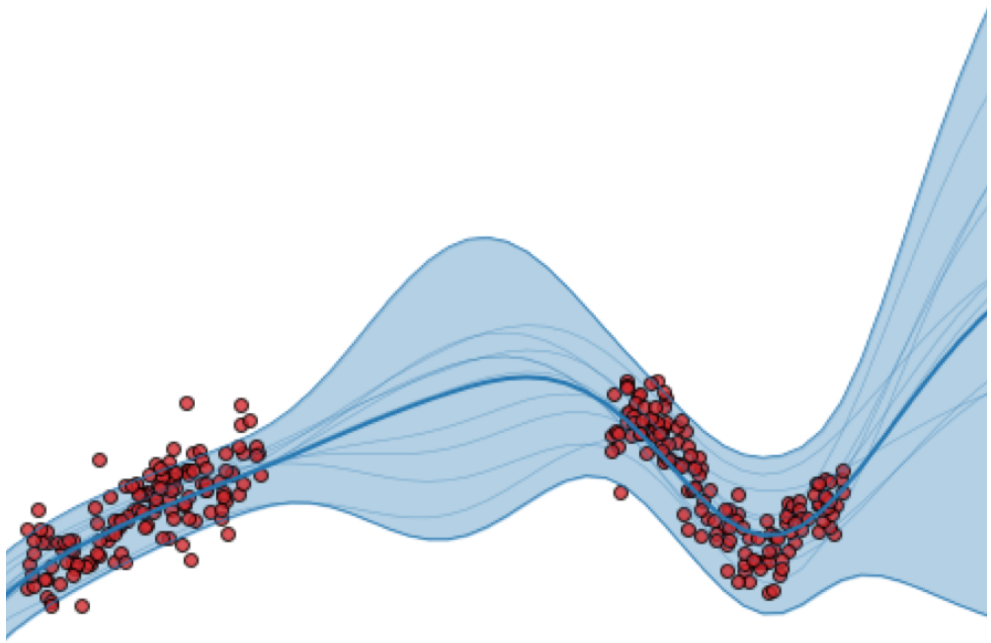


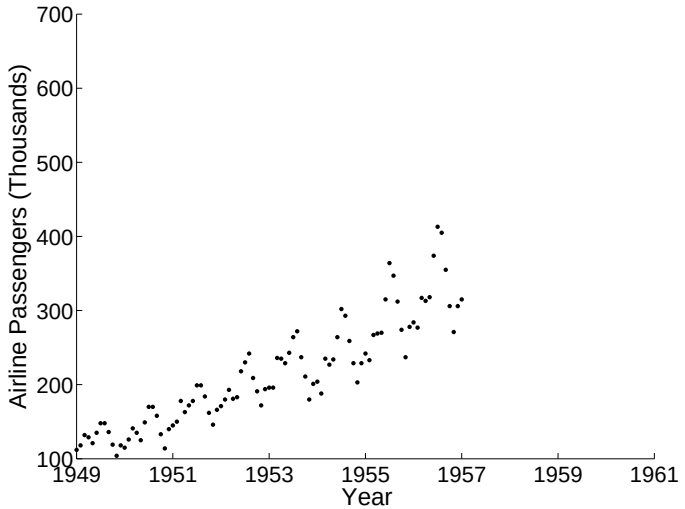


Bayesian Deep Learning and a Probabilistic Perspective of Generalization
Wilson and Izmailov, 2020
arXiv 2002.08791

What is Bayesian learning?

- The key distinguishing property of a Bayesian approach is **marginalization** instead of optimization.
- Rather than use a single setting of parameters **w**, use all settings weighted by their posterior probabilities in a *Bayesian model average*.





Basic Regression Problem

- Training set of n targets (observations) $\mathbf{y} = (y(x_1), \ldots, y(x_n))^{\mathrm{T}}$.
- Observations evaluated at inputs $X = (x_1, \ldots, x_n)^{\mathrm{T}}$.
- Want to predict the value of $y(x_*)$ at a test input x_*.

For example: Given airline passenger numbers $\mathbf{y}$ measured at times X, what will be the number of passengers when $x_* = 1961$?

Just knowing high school math, what might you try?

Guess the parametric form of a function that could fit the data

- $f(x, \mathbf{w}) = \mathbf{w}^{\mathrm{T}} x$ [Linear function of $\mathbf{w}$ and x]
- $f(x, \mathbf{w}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(x)$ [Linear function of $\mathbf{w}$] (Linear Basis Function Model)
- $f(x, \mathbf{w}) = g(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(x))$ [Non-linear in x and $\mathbf{w}$] (E.g., Neural Network)

$\boldsymbol{\phi}(x)$ is a vector of basis functions. For example, if $\boldsymbol{\phi}(x) = (1, x, x^2)$ and $x \in \mathbb{R}^1$ then $f(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2$ is a quadratic function.

Choose an error measure $E(\mathbf{w})$, minimize with respect to $\mathbf{w}$

- $E(\mathbf{w}) = \sum_{i=1}^{n} [f(x_i, \mathbf{w}) - y(x_i)]^2$

A probabilistic approach

We could explicitly account for noise in our model.

- $y(x) = f(x, \mathbf{w}) + \epsilon(x)$, where $\epsilon(x)$ is a noise function.

One commonly takes $\epsilon(x) = \mathcal{N}(0, \sigma^2)$ for i.i.d. additive Gaussian noise, in which case

$$p(y(x)|x, \mathbf{w}, \sigma^2) = \mathcal{N}(y(x); f(x, \mathbf{w}), \sigma^2) \qquad \text{Observation Model} \qquad (3)$$

$$p(\mathbf{y}|x, \mathbf{w}, \sigma^2) = \prod_{i=1}^{n} \mathcal{N}(y(x_i); f(x_i, \mathbf{w}), \sigma^2) \qquad \text{Likelihood} \qquad (4)$$

- Maximize the likelihood of the data $p(\mathbf{y}|x, \mathbf{w}, \sigma^2)$ with respect to $\sigma^2, \mathbf{w}$.

For a Gaussian noise model, this approach will make the same predictions as using a squared loss error function:

$$\log p(\mathbf{y}|X, \mathbf{w}, \sigma^2) \propto -\frac{1}{2\sigma^2} \sum_{i=1}^{n} [f(x_i, \mathbf{w}) - y(x_i)]^2 \qquad (5)$$

- The probabilistic approach helps us interpret the error measure in a deterministic approach, and gives us a sense of the noise level σ^2.
- Both approaches are prone to *over-fitting* for flexible $f(x, \mathbf{w})$: low error on the training data, high error on the test set.

Regularization

- Use a penalized log likelihood (or error function), such as

$$E(\mathbf{w}) = \overbrace{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(f(x_i,\mathbf{w}) - y(x_i)^2)}^{\text{model fit}} \overbrace{-\lambda \mathbf{w}^{\mathrm{T}}\mathbf{w}}^{\text{complexity penalty}} \quad . \tag{6}$$

- **But how should we define and penalize complexity?**
- Can set λ using *cross-validation*.
- Same as maximizing a posterior $\log p(\mathbf{w}|\mathbf{y}, X) = \log p(\mathbf{y}|\mathbf{w}, X) + \log p(\mathbf{w}) + c$ with a Gaussian prior $p(\mathbf{w})$. **But this is not Bayesian!**

Bayesian Inference

Bayes' Rule

$$p(a|b) = p(b|a)p(a)/p(b)\,, \qquad p(a|b) \propto p(b|a)p(a)\,. \tag{7}$$

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{marginal likelihood}}, \quad p(\mathbf{w}|\mathbf{y}, X, \sigma^2) = \frac{p(\mathbf{y}|X, \mathbf{w}, \sigma^2)p(\mathbf{w})}{p(\mathbf{y}|X, \sigma^2)}\,. \tag{8}$$

Sum Rule

$$p(x) = \sum_y p(x, y) \tag{9}$$

Product Rule

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x) \tag{10}$$

Bayesian Predictive Distribution

Sum rule: $p(x) = \sum_x p(x, y)$. **Product rule:** $p(x, y) = p(x|y)p(y) = p(y|x)p(x)$.

$$p(y|x_*, \mathbf{y}, X) = \int p(y|x_*, \mathbf{w})p(\mathbf{w}|\mathbf{y}, X)d\mathbf{w}\,. \tag{11}$$

- Think of each setting of $\mathbf{w}$ as a different model. Eq. (32) is a *Bayesian model average*, an average of infinitely many models weighted by their posterior probabilities.
- Represents *epistemic uncertainty* over which $f(x, w)$ fits the data.
- Automatically calibrated complexity even with highly flexible models.
- Can view classical training as using an approximate posterior $q(\mathbf{w}|\mathbf{y}, X) = \delta(w = w_{\text{MAP}})$.

Bayesian Model Averaging is Not Model Combination

$$p(y|\mathcal{D}) = \sum_h p(y|h)p(h|\mathcal{D})$$

- The weights $p(h|\mathcal{D})$ only represent a statistical inability to distinguish between hypotheses.
- Assumes that combination models are not in the hypothesis space.
- In the limit of infinite data, $p(h|\mathcal{D})$ will collapse onto a point mass.

Bayesian model averaging is not model combination. Minka, T. Technical Report, 2000.

Example: Biased Coin

Suppose we flip a biased coin with probability λ of landing tails.

1. What is the likelihood of a set of data $\mathcal{D} = \{y_1, y_2, \ldots, y_n\}$?
2. What is the maximum likelihood solution for λ?
3. Suppose the first flip is tails. What is the probability that the next flip will be a tails, using maximum likelihood?

Example: Biased Coin

Likelihood of the data:

$$p(\{y_i\}_{i=1}^n) = \prod_{i=1}^{n} \lambda^{y_i}(1-\lambda)^{1-y_i} \tag{12}$$

where $y_i = 1$ if y_i is tails, and $y_i = 0$ if y_i is heads.

Likelihood of getting m tails is

$$p(\mathcal{D}|m,\lambda) = \binom{n}{m}\lambda^m(1-\lambda)^{n-m} \tag{13}$$

Example: Biased Coin

Likelihood of getting m tails is

$$p(\mathcal{D}|m,\lambda) = \binom{n}{m}\lambda^m(1-\lambda)^{n-m} \tag{14}$$

The maximum likelihood solution is

$$\hat{\lambda}_{\mathrm{ML}} = \mathrm{argmax}_\lambda p(\mathcal{D}|m,\lambda) = \frac{m}{n} \tag{15}$$

Example: Biased Coin

Likelihood of getting m tails is

$$p(\mathcal{D}|m,\lambda) = \binom{n}{m}\lambda^m(1-\lambda)^{n-m} \tag{16}$$

The maximum likelihood solution is

$$\hat{\lambda}_{\mathrm{ML}} = \mathrm{argmax}_\lambda p(\mathcal{D}|m,\lambda) = \frac{m}{n} \tag{17}$$

Do you believe this solution? Why or why not?

Example: Biased Coin

Likelihood of getting m tails is

$$p(\mathcal{D}|m,\lambda) = \binom{n}{m}\lambda^m(1-\lambda)^{n-m} \tag{18}$$

Example: Biased Coin

Likelihood of getting m tails is

$$p(\mathcal{D}|m,\lambda) = \binom{n}{m}\lambda^m(1-\lambda)^{n-m} \tag{19}$$

If we choose a prior $p(\lambda) \propto \lambda^\alpha(1-\lambda)^\beta$ then the posterior will have the same functional form as the prior.

Example: Biased Coin

Likelihood of getting m tails is

$$p(\mathcal{D}|m,\lambda) = \binom{n}{m}\lambda^m(1-\lambda)^{n-m} \tag{20}$$

If we choose a prior $p(\lambda) \propto \lambda^\alpha(1-\lambda)^\beta$ then the posterior will have the same functional form as the prior.
We can choose the beta distribution:

$$Beta(\lambda|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\lambda^{a-1}(1-\lambda)^{b-1} \tag{21}$$

The Gamma functions ensure that the distribution is normalized:

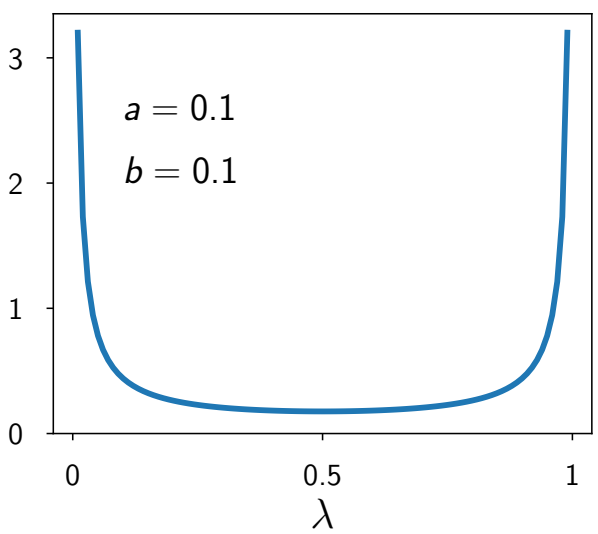
$$\int Beta(\lambda|a,b)d\lambda = 1 \tag{22}$$

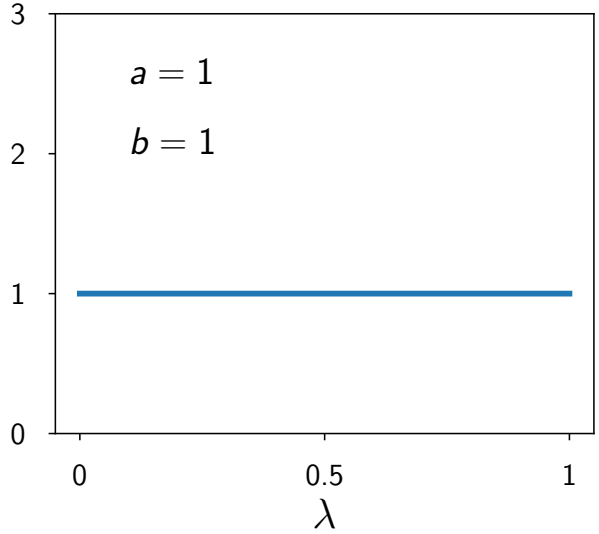
Moments:

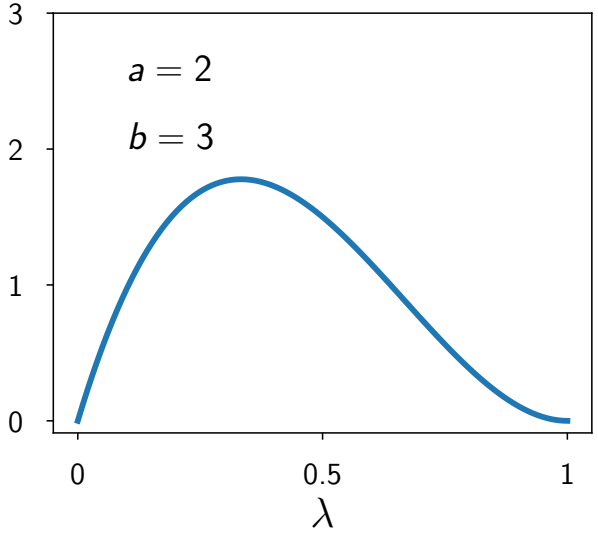
$$\mathbb{E}[\lambda] = \frac{a}{a+b} \tag{23}$$

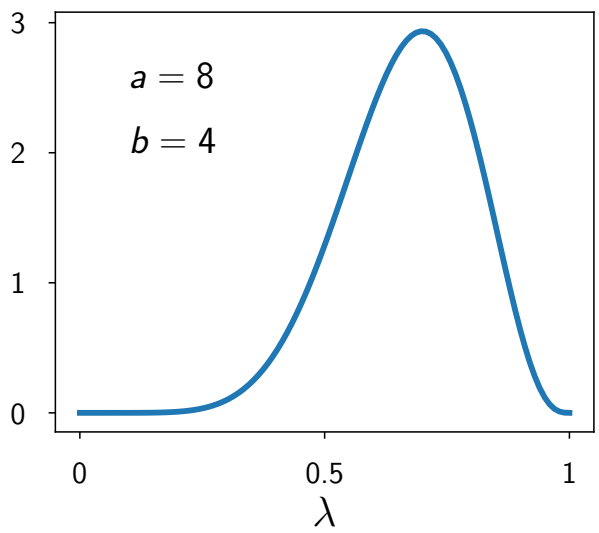
$$\text{var}[\lambda] = \frac{ab}{(a+b)^2(a+b-1)}\,. \tag{24}$$

Beta Distribution

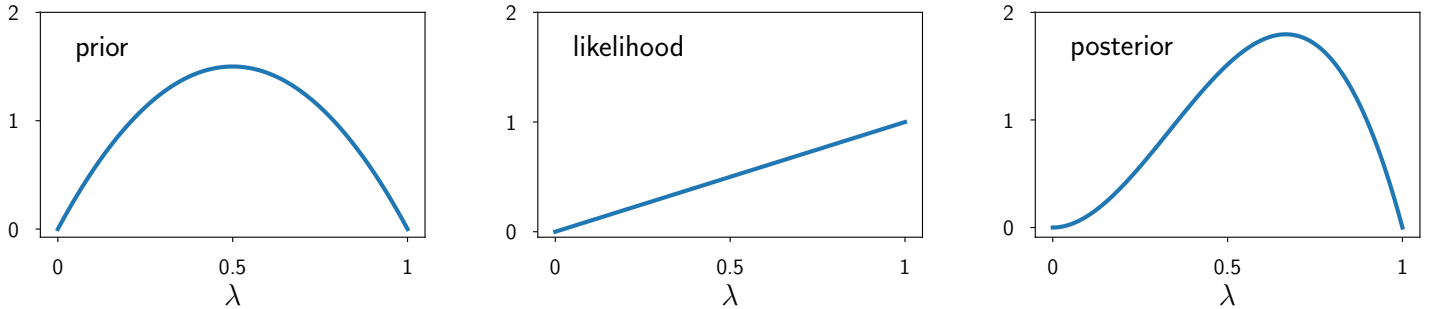








Example: Biased Coin



Applying Bayes theorem, we find:

$$p(\lambda|\mathcal{D}) \propto p(\mathcal{D}|\lambda)p(\lambda) \tag{25}$$

$$= Beta(\lambda; m+a, n-m+b) \tag{26}$$

We can view a and b as pseudo-observations!

$$\mathbb{E}[\lambda|\mathcal{D}] = \frac{m+a}{n+a+b} \tag{27}$$

1. What is the probability that the next flip is tails?
2. What happens in the limits of a, b?
3. What happens in the limit of infinite data?

Example: Biased Coin

Applying Bayes theorem, we find:

$$p(\lambda|\mathcal{D}) \propto p(\mathcal{D}|\lambda)p(\lambda) \tag{28}$$

$$= Beta(\lambda; m+a, n-m+b) \tag{29}$$

We can view a and b as pseudo-observations!

$$\mathbb{E}[\lambda|\mathcal{D}] = \frac{m+a}{n+a+b} \tag{30}$$

1. What is the probability that the next flip is tails?
2. What happens in the limits of a, b?
3. What happens in the limit of infinite data?
4. **Does the MAP estimate**

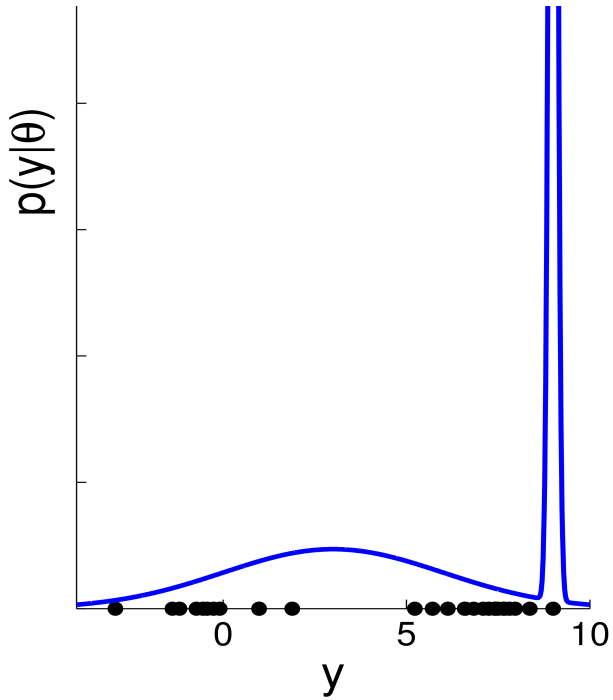
$$\hat{\lambda}_{MAP} = \textbf{argmax}_\lambda \log p(\lambda|\mathcal{D}) = \textbf{argmax}_\lambda \log p(\mathcal{D}|\lambda) + \log p(\lambda) \tag{31}$$

with a uniform prior $p(\lambda)$ give the same answer as Bayesian marginalization to find $p(\text{tails next flip}|\mathcal{D}) = \int \lambda p(\lambda|\mathcal{D})d\lambda = \mathbb{E}[\lambda|\mathcal{D}]$ with a uniform prior $p(\lambda)$?

Example: Density Estimation

- Observations $y_1, \ldots, y_n$ from unknown density $p(y)$.
- Specify an observation model. For example, we can let the points be drawn from a mixture of Gaussians:
  $p(y|\theta) = w_1 \mathcal{N}(y|\mu_1, \sigma_1^2) + w_2 \mathcal{N}(y|\mu_2, \sigma_2^2)$,
  $\theta = \{w_1, w_2, \mu_1, \mu_2, \sigma_1, \sigma_2\}$.
- Likelihood $p(\mathbf{y}|\theta) = \prod_{i=1}^{n} p(y_i|\theta)$.

Can learn all free parameters θ using maximum likelihood...



$$\hat{\theta}_{\mathrm{ML}} = \mathrm{argmax}_\theta \prod_{i=1}^{n} \frac{w_1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{1}{2\sigma_1^2}(y_i - \mu_1)^2\right) + \frac{w_2}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{1}{2\sigma_2^2}(y_i - \mu_2)^2\right)$$

Can you look at this equation and see what θ achieves maximum likelihood?

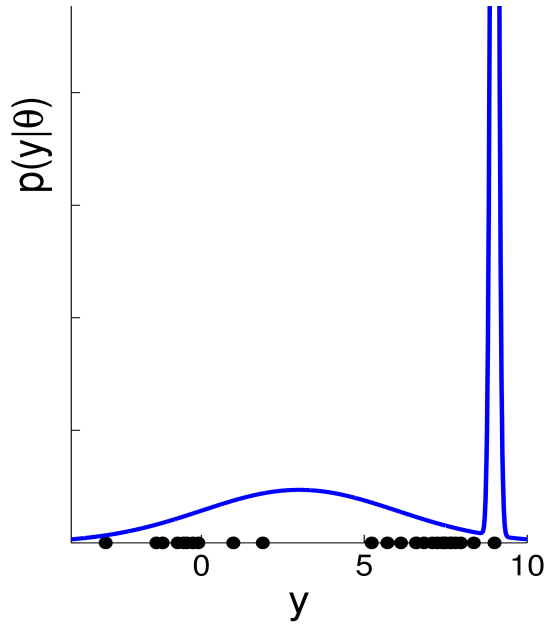
Regularization = MAP $\neq$ Bayesian Inference

Regularization or MAP

- Find

$$\text{argmax}_\theta \log p(\theta|\mathbf{y}) \stackrel{c}{=} \overbrace{\log p(\mathbf{y}|\theta)}^{\text{model fit}} + \overbrace{\log p(\theta)}^{\text{complexity penalty}}$$

- Choose $p(\theta)$ such that $p(\theta) \to 0$ faster than $p(\mathbf{y}|\theta) \to \infty$ as σ_1 or $\sigma_2 \to 0$.



Bayesian Inference

- Predictive Distribution: $p(y_*|\mathbf{y}) = \int p(y_*|\theta)p(\theta|\mathbf{y})d\theta$.
- Parameter Posterior: $p(\theta|\mathbf{y}) \propto p(\mathbf{y}|\theta)p(\theta)$.

Approximate Inference

Ultimately we wish to compute a Bayesian model average:

$$p(y|x_*, \mathcal{D}) = \int p(y|x_*, w)p(w|\mathcal{D})dw\,. \tag{32}$$

For most models, including Bayesian neural networks, this integral is *not analytic*. It is common to use a *Simple Monte Carlo* approximation:

$$p(y|x_*, \mathcal{D}) \approx \frac{1}{J}\sum_j p(y|x_*, w_j)\,, \quad w_j \sim q(w|\mathcal{D})\,. \tag{33}$$

w_j are samples from an approximate posterior $q(w|\mathcal{D})$ typically found by:

1. **Deterministic Methods:** Approximate $p(w|\mathcal{D})$ with convenient $q(w|\mathcal{D}, \theta)$, often Gaussian. θ (e.g. mean of q) chosen to make q close to p. Variational methods find $\text{argmin}_\theta \mathcal{KL}(q||p)$. Classical training: $q(w|\mathcal{D}) = \delta(w = w_{\text{MAP}})$.
   E.g.: Laplace, Expectation Propagation, Variational, Standard Training.
2. **MCMC:** Form a Markov chain of approximate (but asymptotically exact) samples from $p(w|\mathcal{D})$.
   E.g.: Metropolis-Hastings, Hamiltonian Monte Carlo, SGLD, SGHMC.

- Later we will argue we may want to avoid the simple MC perspective.