Bayesian Inference

CMPUT 261: Introduction to Artificial Intelligence

P&M §10.4, §8.6

Logistics

- **Assignment #2** grades and feedback are available on eclass
- **Midterm** is **Thursday, March 2**
    - Coverage: Everything up to and including today (Bayesian Inference)
    - Logistics details on eclass

Recap: Linear Models

- **Linear regression** is a simple model for predicting real quantities
  - Can be used for classification too, either based on **sign** of prediction or using **logistic regression**
- **Gradient descent** is a general, widely-used training procedure (with several variants)
  - Linear models can be optimized in **closed form** for certain losses
  - In practice often optimized with gradient descent

Recap: Overfitting

- **Overfitting** is when a learned model fails to **generalize** due to **overconfidence** and/or learning **spurious regularities**
- **Causes of overfitting:**
  - Bias: Systematic choice of suboptimal hypotheses
  - Variance: Different training sets can yield very different hypotheses
  - Noise: Unpredictability that is inherent in the process
    (e.g., coin flips cannot be perfectly predicted, even by the "true" model)
- **Avoiding overfitting:**
  1. **Pseudocounts:** Add **imaginary** observations
  2. **Regularization: Penalize** model complexity
  3. **Cross-validation:** Reserve validation data to **estimate** overfitting / test error
     - Used to select values for **hyperparameters**

Lecture Outline

1. Recap & Logistics
2. Learning Model Probabilities
3. Using Model Probabilities
4. Prior Distributions as Bias

After this lecture, you should be able to:

- derive the posterior probability **of a model** using Bayes' rule
- explain how to use the Beta and Bernoulli distributions for Bayesian learning
- demonstrate model averaging

Learning Point Estimates

- So far, we have considered how to find the best **single** model (hypothesis), e.g.,
    - learn ***a*** classification function
    - optimize ***the*** weights of a linear or logistic regression
- The **predictions** might be a probability distribution, but they are coming out of a single **model**:

$$P(Y \mid X)$$ Probability of target Y given observation X

- We have been learning **point estimates** of our model

Learning Model Probabilities

- Instead, we could learn a distribution over **models**:
  - $\Pr(X, Y \mid \theta)$ Probability of target Y and features X given model θ
  - $\Pr(\theta \mid D)$ Probability of model θ given dataset D
- This is called **Bayesian learning**: we never discard any model, we only weight them differently depending upon their **posterior probability**
- **Question:** Why would we want to do that?

What is a Model?

- $\Pr(X, Y \mid \theta)$ Probability of target Y and features X given model θ
- $\Pr(\theta \mid D)$ Probability of model θ given dataset D

- We can do Bayesian learning over **finite** sets of models:
  - e.g., { rank by feature $\theta \mid \theta \in$ {height, weight, age} }
- We can do Bayesian learning over **parametric families** of models:
  - e.g., { regression with weights $w_0=\theta_1$, $w_1=\theta_2 \mid \theta \in \mathbb{R}^2$ }
- We can mix the two!
  - θ can encode choice of **model family and parameters**

What is the Dataset?

- $\Pr(X, Y \mid \theta)$ Probability of target Y and features X given model θ
- $\Pr(\theta \mid D)$ Probability of model θ given dataset D

- We have an expression for the probability of a single example given a model: $\Pr(X, Y \mid \theta)$
- **Question:** What is the expression for the probability of a dataset of observations $D = \{(X_1, Y_1), \ldots, (X_m, Y_m)\}$ given a model?
  - Assuming that the dataset are independent, identically distributed observations: $(X_i, Y_i) \sim P(X, Y \mid \theta)$

$$\Pr(D \mid \theta) = \Pr(X_1, Y_1 \mid \theta) \times \ldots \times \Pr(X_m, Y_m \mid \theta)$$

$$= \prod_{i=1}^{m} \Pr(X_i, Y_i \mid \theta)$$

What is the Posterior Model Probability?

- $\Pr(X, Y \mid \theta)$ Probability of target Y and features X given model θ
- $\Pr(\theta \mid D)$ Probability of model θ given dataset D

Now we can use **Bayes' Rule** to compute the posterior probability of a model θ:

Prior probability of model θ

$$\Pr(\theta \mid D) = \frac{\boxed{\Pr(D \mid \theta)}\,\boxed{\Pr(\theta)}}{\Pr(D)}$$

$$= \frac{\prod_i \Pr(X_i, Y_i \mid \theta)\Pr(\theta)}{\Pr(D)}$$

$$= \frac{\prod_i \Pr(X_i, Y_i \mid \theta)\Pr(\theta)}{\sum_{\theta'} \Pr(D \mid \theta')\Pr(\theta')}$$

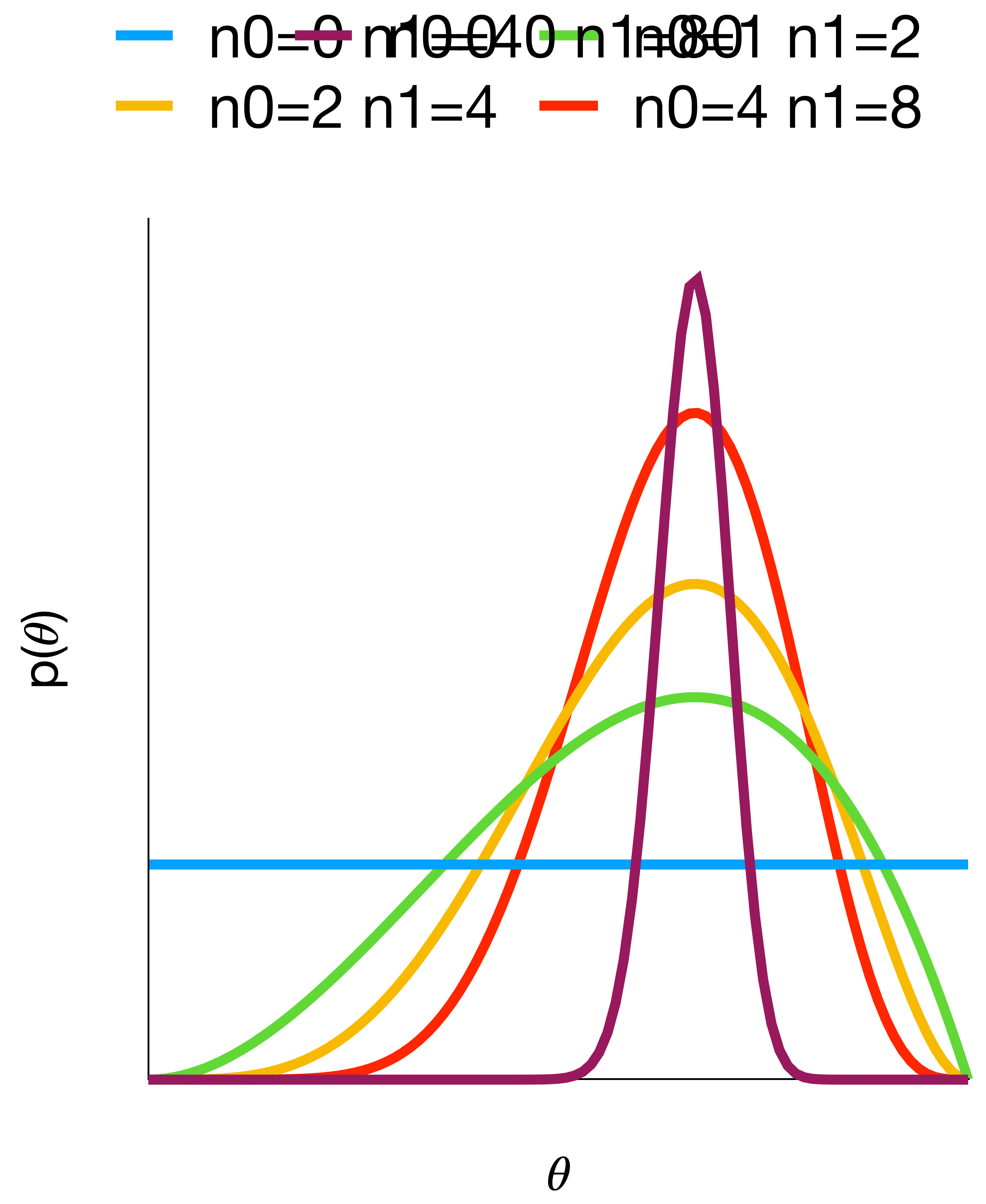
Likelihood of data D given model θ

Example: Biased Coin

- Back to coin flipping! We can flip a coin and observe heads or tails, but we don't know the coin's bias
- Model: **Binomial observations**
  - Observations: $Y \in \{h, t\}$
  - Bias: $\theta \in [0,1]$
  - Likelihood: $\Pr(H \mid \theta) = \theta$
  - **Question:** What should the **prior** $\Pr(\theta)$ be?

Biased Coin: Posterior Probabilities

- Before we see any flips, all biases are equally probable (according to our prior)
- After more and more flips, we become more confident in θ
- θ with **highest probability** is 2/3
  - **Expected** value of θ is less! (**why**?)
  - But with more observations, mode and expected value get **closer**



Beta-Binomial Models

- Likelihood: $P(h \mid \theta) = \theta$
  - aka **Bernoulli**$(h \mid \theta)$
  - Dataset likelihood: $\theta^{n_1} \times (1 - \theta)^{n_0}$
  - aka **Binomial**(n_1, n_0)
- Prior: $P(\theta) \propto 1$
  - aka **Beta**$(1,1)$
- Models of this kind are called **Beta-Binomial models**
- They can be solved analytically: $Pr(\theta \mid D) = \text{Beta}(1 + n_1, 1 + n_0)$

Conjugate Priors

- The beta distribution is a **conjgate prior** for the binomial distribution:
  - Updating a beta prior with a binomial likelihood gives a **beta posterior**
- Other distributions have this property:
  - Gaussian-Gaussian (for means)
  - Dirichlet-Multinomial (generalization of Beta-Binomial for multiple values)

Using Model Probabilities

So we can estimate $\Pr(\theta \mid D)$. What can we do with it?

1. Parameter estimates
2. Target predictions (model averaging)
3. Target predictions (point estimates)

1. Parameter Estimates

- Sometimes, we really want to know the **parameters** of a model itself
- E.g., maybe I don't care about predicting the next coin flip, but I do want to know whether the coin is fair
- Can use $\Pr(\theta \mid D)$ to make statements like

$$\Pr(0.49 \leq \theta \leq 0.51) > 0.9$$

2. Model Averaging

- Sometimes we do want to make **predictions**:

$$\Pr(Y \mid D) = \sum_{\theta} \Pr(Y \mid \theta) \Pr(\theta \mid D)$$

- This is called the **posterior predictive distribution**
- **Question:** How is this different from just learning a point estimate of a model, and then predicting with that model?

3. Maximum A Posteriori

- Sometimes we do want to make predictions, **but...**

$$\Pr(Y \mid D) = \int_0^1 \Pr(Y \mid \theta) \Pr(\theta \mid D) d\theta$$

- the posterior predictive distribution may be **expensive** to compute (or even **intractable**)
- One possible solution is to use the **maximum a posterior** model as a point estimate:

$$\Pr(Y \mid D) \simeq \Pr(Y \mid \hat{\theta}) \qquad \text{where } \hat{\theta} = \arg\max_{\theta} \Pr(\theta \mid D)$$

- **Question:** Why would you do this instead of just using a point estimate that was computed in the usual way?

Prior Distributions as Bias

- Suppose I'm comparing two models, θ_1 and θ_2 such that

$$\Pr(D \mid \theta_1) = Pr(D \mid \theta_2)$$

- **Question:** Which model has higher **posterior probability**?
- Priors are a way of encoding **bias**: they tell use which models to prefer when the data doesn't

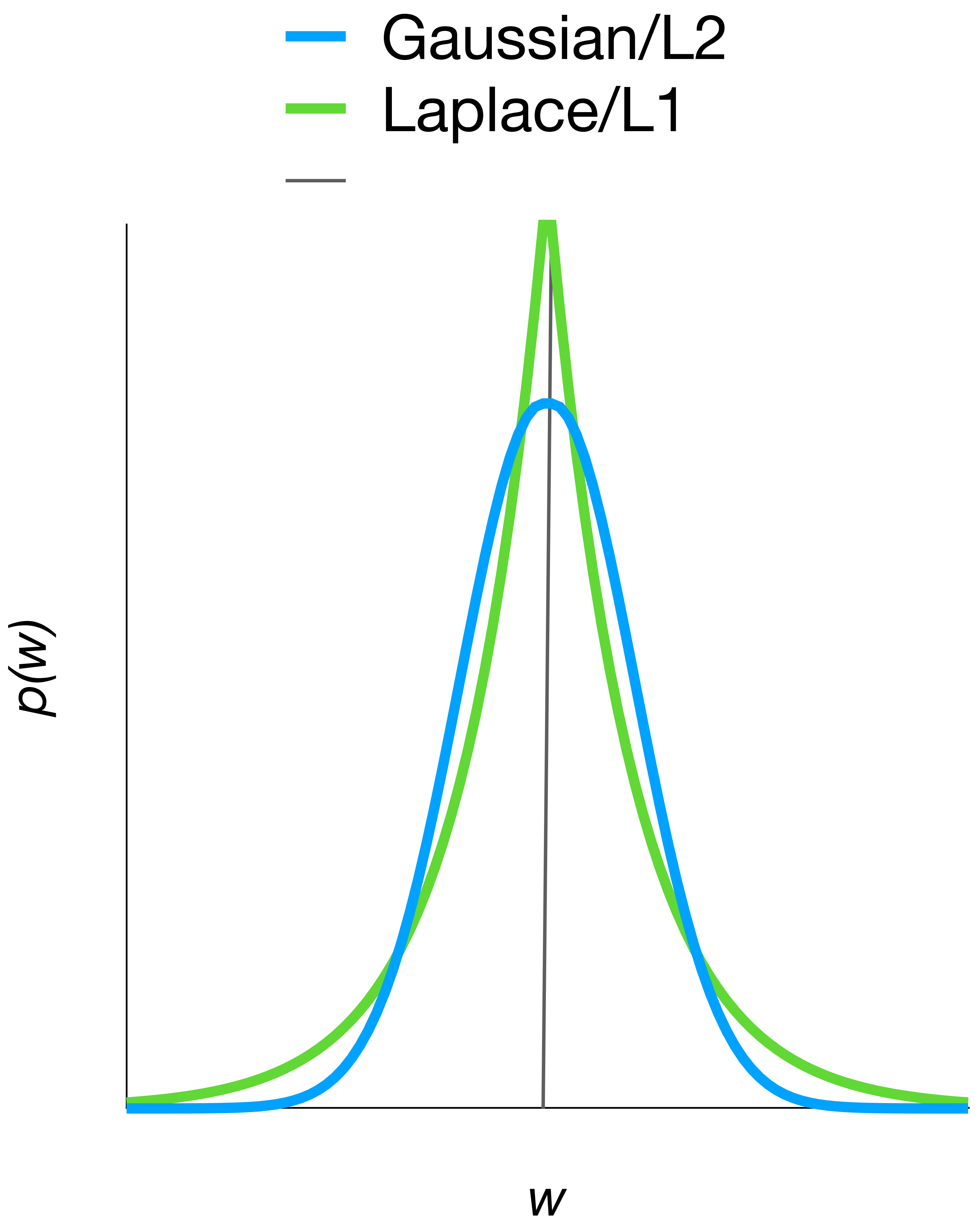
Priors for Pseudocounts

- We can straightforwardly encode pseudocounts as prior information in Beta-Binomial and Dirichlet-Multinomial models
- E.g., for pseudocounts k_1 and k_0,

$$p(\theta) = \text{Beta}(1 + k_1, 1 + k_0)$$

Priors for Regularization

- Some **regularizers** can be encoded as priors also
- **L2 regularization** is equivalent to a **Gaussian** prior on the weights: $p(w) = \mathcal{N}(w \mid m, s)$
- **L1 regularization** is equivalent to a **Laplacian** prior on the weights: $p(w) = \exp(|w|)/2$



Summary

- **Cross-validation** is a powerful technique for selecting hyperparameters based on data
- In Bayesian Learning, we learn a **distribution** over models instead of a **single model**
- When the model is **conjugate**, posterior probabilities can be computed **analytically**
- We can make predictions by **model averaging** to compute the **posterior predictive distribution**
- The **prior** can encode **bias over models**, much the same as **regularization**