Lecture Notes: Statistical Estimation

MEAM/EE/CBE/MSE 4600: AI for Science and Engineering

Week 3: From Data to Discovery

LECTURE 1: Maximum Likelihood Estimation and Regularization

---

1. Introduction: The Parameter Estimation Problem

In previous lectures on probability, we assumed all parameters $\boldsymbol{\theta}$ of our probability models were known. In practice, we must **learn** these parameters from data $\mathcal{D}$. This process—called **model fitting** or **training**—is at the heart of machine learning.

Most estimation methods reduce to an optimization problem:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$$

where $\mathcal{L}(\boldsymbol{\theta})$ is some loss function or objective function.

Key Questions We'll Answer:

- How do we find the "best" parameters? → **Maximum Likelihood Estimation (MLE)**
- How do we prevent overfitting? → **Regularization & MAP Estimation**
- How do we quantify uncertainty in our estimates? → **Bayesian Inference**

Running Example: Throughout these lectures, we'll frequently return to coin flipping (Bernoulli distribution) and Gaussian distributions as our primary examples, since they illustrate all the key concepts while remaining analytically tractable.

---

2. Maximum Likelihood Estimation (MLE)

2.1 The Core Idea

The most common approach to parameter estimation is to pick parameters that assign the **highest probability** to the observed training data.

> **Intuition:** Good parameters should make the observed data "likely." If we saw a particular dataset, the best explanation is parameters under which that dataset would be probable.

Definition:

$$\boxed{\hat{\boldsymbol{\theta}}_{\text{MLE}} = \arg\max_{\boldsymbol{\theta}} p(\mathcal{D}|\boldsymbol{\theta})}$$

where $p(\mathcal{D}|\boldsymbol{\theta})$ is the **likelihood function**—the probability of observing data $\mathcal{D}$ given parameters $\boldsymbol{\theta}$.

Important Distinction: The likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ is a function of $\boldsymbol{\theta}$, not $\mathcal{D}$. The data is fixed (observed); we're searching over parameters.

2.2 The IID Assumption and Log-Likelihood

We typically assume training examples are **independently and identically distributed (iid)**. This means each data point is drawn from the same distribution, and the draws don't influence each other.

Under the iid assumption, the joint probability factorizes as a product:

$$p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{n=1}^{N} p(y_n|\mathbf{x}_n, \boldsymbol{\theta})$$

Why work with log-likelihood?

Taking the logarithm of the likelihood offers several advantages:

1. **Products become sums:** Derivatives of sums are much easier to compute than derivatives of products
2. **Numerical stability:** Products of many small probabilities can underflow to zero; sums of log-probabilities remain numerically stable
3. **Monotonicity:** Since $\log$ is monotonically increasing, maximizing $\log p(\mathcal{D}|\boldsymbol{\theta})$ gives the same answer as maximizing $p(\mathcal{D}|\boldsymbol{\theta})$

The **log-likelihood** is defined as:

$$\ell(\boldsymbol{\theta}) \triangleq \log p(\mathcal{D}|\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(y_n|\mathbf{x}_n, \boldsymbol{\theta})$$

Negative Log-Likelihood (NLL)

Since optimization algorithms are typically designed to **minimize** functions (gradient descent descends!), we

work with the negative log-likelihood:

$$\boxed{\text{NLL}(\boldsymbol{\theta}) = -\log p(\mathcal{D}|\boldsymbol{\theta}) = -\sum_{n=1}^{N} \log p(y_n|\mathbf{x}_n, \boldsymbol{\theta})}$$

MLE = Minimizing the NLL

2.3 Information-Theoretic Justification: MLE and KL Divergence

MLE has a beautiful information-theoretic interpretation that connects it to the goal of approximating the true data distribution.

The **empirical distribution** of observed data is:

$$p_{\mathcal{D}}(y) = \frac{1}{N}\sum_{n=1}^{N} \delta(y - y_n)$$

This is a sum of Dirac delta functions—it puts a "spike" of probability mass $1/N$ at each observed data point and zero probability everywhere else.

Our goal is to find a model $q(y) = p(y|\boldsymbol{\theta})$ that is "close" to $p_{\mathcal{D}}(y)$. A natural measure of dissimilarity between distributions is the **Kullback-Leibler (KL) divergence**:

$$D_{\text{KL}}(p_{\mathcal{D}} \| q) = \sum_{y} p_{\mathcal{D}}(y) \log \frac{p_{\mathcal{D}}(y)}{q(y)}$$

The KL divergence is always non-negative and equals zero if and only if $p = q$.

Expanding the KL divergence:

$$D_{\text{KL}}(p_{\mathcal{D}} \| q) = \underbrace{-H(p_{\mathcal{D}})}_{\text{entropy of empirical dist.}} \underbrace{- \sum_{y} p_{\mathcal{D}}(y) \log q(y)}_{\text{cross-entropy}}$$

The first term is the negative entropy of the empirical distribution, which is **constant** with respect to our model parameters $\boldsymbol{\theta}$. The second term, when we substitute in our empirical distribution, becomes:

$$-\sum_y p_{\mathcal{D}}(y) \log p(y|\boldsymbol{\theta}) = -\frac{1}{N}\sum_{n=1}^{N} \log p(y_n|\boldsymbol{\theta}) = \frac{1}{N} \cdot \text{NLL}(\boldsymbol{\theta})$$

Therefore:

$$D_{\text{KL}}(p_{\mathcal{D}} \| q) = \text{const} + \frac{1}{N} \cdot \text{NLL}(\boldsymbol{\theta})$$

> **Key Insight:** Minimizing KL divergence between the empirical distribution and our model distribution is equivalent to minimizing the NLL, which gives us the MLE! This tells us that MLE is finding the model that best approximates the observed data distribution.

2.4 MLE Example 1: Bernoulli Distribution (Coin Flipping)

Setup: Let $Y \in \{0, 1\}$ be a binary random variable with $Y \sim \text{Ber}(\theta)$, where $\theta = P(Y = 1)$ is the probability of heads.

Data: We observe $\mathcal{D} = \{y_1, \ldots, y_N\}$. Define:

- $N_1 = \sum_{n=1}^{N} \mathbb{I}(y_n = 1)$ = number of heads
- $N_0 = N - N_1$ = number of tails

These counts (N_1, N_0) are called **sufficient statistics**—they contain all the information needed to compute the MLE.

Likelihood:

$$p(\mathcal{D}|\theta) = \prod_{n=1}^{N} \theta^{y_n}(1-\theta)^{1-y_n} = \theta^{N_1}(1-\theta)^{N_0}$$

Negative Log-Likelihood:

$$\text{NLL}(\theta) = -[N_1 \log \theta + N_0 \log(1-\theta)]$$

Finding the MLE: We take the derivative and set it to zero:

$$\frac{d}{d\theta}\text{NLL}(\theta) = -\frac{N_1}{\theta} + \frac{N_0}{1-\theta} = 0$$

Solving for θ:

$$\frac{N_1}{\theta} = \frac{N_0}{1-\theta}$$

$$N_1(1-\theta) = N_0\theta$$

$$N_1 = \theta(N_0 + N_1)$$

$$\boxed{\hat{\theta}_{\text{MLE}} = \frac{N_1}{N_0 + N_1} = \frac{N_1}{N}}$$

> **Interpretation:** The MLE is simply the **empirical fraction** of heads—exactly what intuition suggests! If you flip a coin 100 times and see 60 heads, your best estimate is $\hat{\theta} = 0.6$.

ML Connection: This same principle underlies training binary classifiers. When we fit logistic regression, we're maximizing the likelihood of observed class labels under a Bernoulli model.

2.5 MLE Example 2: Categorical Distribution (Dice Rolling)

Setup: Let $Y \in \{1, 2, \ldots, K\}$ with $Y \sim \text{Cat}(\boldsymbol{\theta})$, where $\theta_k = P(Y = k)$ and $\sum_{k=1}^{K} \theta_k = 1$.

Data: Observe counts $N_k = \sum_{n=1}^{N} \mathbb{I}(y_n = k)$ for each category.

NLL:

$$\text{NLL}(\boldsymbol{\theta}) = -\sum_{k=1}^{K} N_k \log \theta_k$$

Constraint: We must have $\sum_k \theta_k = 1$. We use **Lagrange multipliers**:

$$\mathcal{L}(\boldsymbol{\theta}, \lambda) = -\sum_{k} N_k \log \theta_k - \lambda \left(1 - \sum_{k} \theta_k\right)$$

Taking derivatives:

$$\frac{\partial \mathcal{L}}{\partial \theta_k} = -\frac{N_k}{\theta_k} + \lambda = 0 \implies N_k = \lambda \theta_k$$

Summing over all k: $N = \lambda \sum_k \theta_k = \lambda$

Therefore:

$$\boxed{\hat{\theta}_k = \frac{N_k}{N}}$$

> **Interpretation:** The MLE for each category probability is the empirical fraction of observations in that category.

2.6 MLE Example 3: Univariate Gaussian

Setup: $Y \sim \mathcal{N}(\mu, \sigma^2)$, with parameters $\boldsymbol{\theta} = (\mu, \sigma^2)$.

Likelihood:

$$p(\mathcal{D}|\mu, \sigma^2) = \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_n - \mu)^2}{2\sigma^2}\right)$$

NLL:

$$\text{NLL}(\mu, \sigma^2) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mu)^2 + \frac{N}{2} \log(2\pi\sigma^2)$$

MLE for the mean: Taking $\frac{\partial}{\partial \mu}\text{NLL} = 0$:

$$\frac{\partial}{\partial \mu}\text{NLL} = -\frac{1}{\sigma^2} \sum_{n=1}^{N} (y_n - \mu) = 0$$

$$\sum_{n=1}^{N} (y_n - \mu) = 0 \implies N\mu = \sum_{n=1}^{N} y_n$$

$$\boxed{\hat{\mu}_{\text{MLE}} = \frac{1}{N}\sum_{n=1}^{N} y_n = \bar{y}}$$

MLE for the variance: Taking $\frac{\partial}{\partial \sigma^2}\text{NLL} = 0$:

$$\frac{\partial}{\partial \sigma^2}\text{NLL} = -\frac{1}{2(\sigma^2)^2}\sum_{n=1}^{N}(y_n - \mu)^2 + \frac{N}{2\sigma^2} = 0$$

$$\boxed{\hat{\sigma}^2_{\text{MLE}} = \frac{1}{N}\sum_{n=1}^{N}(y_n - \hat{\mu}_{\text{MLE}})^2}$$

> **Important Note on Bias:** The MLE for variance divides by N, not $N-1$. This makes it a **biased** estimator—it systematically underestimates the true variance. The **unbiased** estimator divides by $N-1$:
> $$\hat{\sigma}^2_{\text{unbiased}} = \frac{1}{N-1}\sum_{n=1}^{N}(y_n - \bar{y})^2$$
> Intuitively, we "use up" one degree of freedom estimating the mean, so we have only $N-1$ independent pieces of information for estimating variance. We'll see later why bias isn't always bad!

ML Connection: These formulas are used in Gaussian density estimation, Gaussian Naive Bayes classifiers, and as building blocks for Gaussian mixture models.

2.7 MLE Example 4: Multivariate Gaussian

Setup: $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\mathbf{y} \in \mathbb{R}^D$.

MLE Solutions:

$$\boxed{\hat{\boldsymbol{\mu}}_{\text{MLE}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{y}_n = \bar{\mathbf{y}}}$$

$$\boxed{\hat{\boldsymbol{\Sigma}}_{\text{MLE}} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}_n - \bar{\mathbf{y}})(\mathbf{y}_n - \bar{\mathbf{y}})^T}$$

> **Warning:** The covariance matrix has $O(D^2)$ parameters. If $N < D$, the MLE covariance matrix is **singular** (rank-deficient, non-invertible). Even when $N > D$ but N is not much larger than D, the

> estimate can be **ill-conditioned** (nearly singular, numerically unstable). This motivates regularization!

ML Connection: Covariance estimation appears throughout ML: PCA, Linear Discriminant Analysis (LDA), Gaussian Process regression, and the E-step of EM for Gaussian mixtures.

2.8 MLE Example 5: Linear Regression

Model: We assume outputs are linear functions of inputs plus Gaussian noise:

$$p(y|\mathbf{x};\boldsymbol{\theta}) = \mathcal{N}(y|\mathbf{w}^T\mathbf{x}, \sigma^2)$$

where $\boldsymbol{\theta} = (\mathbf{w}, \sigma^2)$.

NLL (treating σ^2 as fixed):

$$\text{NLL}(\mathbf{w}) = \frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \mathbf{w}^T\mathbf{x}_n)^2 + \text{const}$$

Ignoring constants and the scaling factor $\frac{1}{2\sigma^2}$, minimizing NLL is equivalent to minimizing the **Residual Sum of Squares (RSS)**:

$$\text{RSS}(\mathbf{w}) = \sum_{n=1}^{N}(y_n - \mathbf{w}^T\mathbf{x}_n)^2 = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

The **Mean Squared Error (MSE)** is just $\frac{1}{N}\text{RSS}(\mathbf{w})$.

MLE Solution (Ordinary Least Squares):

Setting the gradient to zero: $\nabla_{\mathbf{w}}\text{RSS} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = 0$

This gives the **normal equations**: $\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}$

$$\boxed{\hat{\mathbf{w}}_{\text{MLE}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}}$$

> **Key Insight:** Minimizing squared error in regression is equivalent to MLE under the assumption of Gaussian noise! The "least squares" method has a probabilistic interpretation.

2.9 Sufficient Statistics

Notice a pattern: for Bernoulli, we only need (N_1, N_0) to compute the MLE—not the full sequence of observations. For Gaussian, we need $(\bar{y}, s^2)$. These are **sufficient statistics**: they contain all information in the data that is relevant for estimating the parameters.

Distribution	Sufficient Statistics
Bernoulli	N_1, N_0 (counts of 1s and 0s)
Categorical	$N_1, N_2, \ldots, N_K$ (counts per class)
Gaussian	$\bar{y}, s^2$ (sample mean and variance)
Multivariate Gaussian	$\bar{\mathbf{y}}, \mathbf{S}$ (sample mean and scatter matrix)

The concept of sufficient statistics is central to understanding why certain summary statistics are "enough" for inference.

2.10 Empirical Risk Minimization (ERM)

MLE uses **log loss**: $\ell(y_n, \boldsymbol{\theta}; \mathbf{x}_n) = -\log p(y_n|\mathbf{x}_n, \boldsymbol{\theta})$

We can generalize to **any** loss function:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} \ell(y_n, \boldsymbol{\theta}; \mathbf{x}_n)$$

This is **Empirical Risk Minimization (ERM)**—minimizing average loss over training data.

Common Loss Functions:

Loss	Formula	Use Case
0-1 Loss	$\ell_{0-1}(y, \hat{y}) = \mathbb{I}(y \neq \hat{y})$	Classification error rate
Squared Loss	$\ell(y, \hat{y}) = (y - \hat{y})^2$	Regression
Log Loss	$\ell(y, \boldsymbol{\theta}) = -\log p(y\|\boldsymbol{\theta})$	Probabilistic models
Hinge Loss	$\ell(y, \eta) = \max(0, 1 - y\eta)$	SVM classification

> **Problem with 0-1 Loss:** It's non-differentiable (a step function), making gradient-based optimization impossible.
> **Solution:** Use **surrogate losses** that upper-bound the 0-1 loss and are differentiable. Log loss and hinge loss are both convex upper bounds on 0-1 loss, which is why logistic regression and SVMs work well for classification.

---

3. The Problem of Overfitting

3.1 When MLE Goes Wrong

A fundamental problem with MLE: it minimizes loss on **training data**, but this may not minimize loss on **future data**. This is called **overfitting**.

Motivating Example: Coin Flipping

Suppose we toss a coin $N = 3$ times and observe 3 heads. The MLE is:

$$\hat{\theta}_{\text{MLE}} = \frac{3}{3} = 1$$

This predicts that **all future tosses will be heads**—clearly unreasonable! We would assign probability zero to ever seeing tails.

What Went Wrong?

1. The model has enough flexibility to perfectly fit the observed data
2. Perfectly matching the empirical distribution leaves no probability mass for unseen events
3. The model memorizes the training data rather than learning generalizable patterns
4. With small samples, the empirical distribution is a poor approximation of the true distribution

This is related to the **black swan paradox** in philosophy: just because you've never seen a black swan doesn't mean they don't exist. A model that assigns zero probability to unobserved events will be badly surprised by reality.

3.2 Regularization: Adding a Complexity Penalty

The solution is to add a **penalty term** that discourages "extreme" or "complex" parameters:

$$\boxed{\mathcal{L}(\boldsymbol{\theta}; \lambda) = \underbrace{\frac{1}{N}\sum_{n=1}^{N} \ell(y_n, \boldsymbol{\theta}; \mathbf{x}_n)}_{\text{Data fit (empirical risk)}} + \underbrace{\lambda \cdot C(\boldsymbol{\theta})}_{\text{Complexity penalty}}}$$

where:

- $\lambda \geq 0$ is the **regularization strength** (hyperparameter)
- $C(\boldsymbol{\theta})$ is a **complexity penalty** (larger for "extreme" parameters)

The Regularization Tradeoff:

- $\lambda = 0$: Pure MLE, may overfit
- λ small: Slight regularization, reduced overfitting
- λ large: Strong regularization, may underfit (ignore data)
- $\lambda \to \infty$: Ignore data entirely, parameters determined only by the penalty

---

3.3 Common Regularizers

L2 Regularization (Weight Decay / Ridge):

$$C(\mathbf{w}) = \|\mathbf{w}\|_2^2 = \sum_{d=1}^{D} w_d^2$$

Effects:

- Encourages weights to be small in magnitude
- Spreads importance across all features (no single feature dominates)
- Leads to smooth solutions
- Has closed-form solutions when combined with linear models

Linear Regression + L2 = Ridge Regression:

$$\mathcal{L}(\mathbf{w}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$$

The closed-form solution is:

$$\hat{\mathbf{w}}_{\text{ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

Note: Adding $\lambda\mathbf{I}$ ensures the matrix is invertible even when $\mathbf{X}^T\mathbf{X}$ is singular!

L1 Regularization (Lasso):

$$C(\mathbf{w}) = \|\mathbf{w}\|_1 = \sum_{d=1}^{D} |w_d|$$

Effects:

- Encourages **sparsity**—drives some weights exactly to zero
- Performs automatic **feature selection**
- No closed-form solution (requires iterative optimization)
- Useful when you believe only a few features are relevant

3.4 The Bias-Variance Tradeoff

To understand regularization deeply, we need to understand the **bias-variance decomposition** of estimation error.

For any estimator $\hat{\theta}$ of a true parameter θ^*, the Mean Squared Error decomposes as:

$$\boxed{\text{MSE}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \theta^*)^2] = \underbrace{\text{Var}(\hat{\theta})}_{\text{Variance}} + \underbrace{(\mathbb{E}[\hat{\theta}] - \theta^*)^2}_{\text{Bias}^2}}$$

Definitions:

- **Bias:** $\text{bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta^*$ — systematic error, how far off we are on average
- **Variance:** $\text{Var}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2]$ — sensitivity to the particular training set

The Tradeoff:

Regularization	Bias	Variance	Typical Scenario
None (MLE)	Low	High	Large N, simple model
Weak (λ small)	Low	Medium	Moderate N
Strong (λ large)	High	Low	Small N, complex model

> **Key Insight:** A biased estimator can have **lower MSE** than an unbiased one if it sufficiently reduces variance! This is why regularization helps—we accept some bias in exchange for much lower variance.

Example: MLE Variance Estimator

The MLE $\hat{\sigma}^2_{\text{MLE}} = \frac{1}{N}\sum(y_n - \bar{y})^2$ is biased (underestimates by factor $\frac{N-1}{N}$).

The unbiased estimator $\hat{\sigma}^2_{\text{unb}} = \frac{1}{N-1}\sum(y_n - \bar{y})^2$ has zero bias.

However, one can show that the MSE of the MLE is actually **lower** than the unbiased estimator for certain parameter values! Being unbiased is not always optimal.

3.5 Choosing the Regularization Strength

How do we pick the hyperparameter λ? We cannot use training error—that would just select $\lambda = 0$.

Solution: Cross-Validation

1. **Holdout Method:** Split data into training set $\mathcal{D}_{\text{train}}$ (~80%) and validation set $\mathcal{D}_{\text{valid}}$ (~20%)
2. For each candidate λ:
   - Fit model on $\mathcal{D}_{\text{train}}$
   - Evaluate loss on $\mathcal{D}_{\text{valid}}$
3. Choose $\lambda^* = \arg\min_\lambda \mathcal{L}(\hat{\boldsymbol{\theta}}_\lambda, \mathcal{D}_{\text{valid}})$
4. Refit on all data with λ^*

K-Fold Cross-Validation: For small datasets, a single split may be unreliable.

1. Partition data into K equal folds
2. For each fold k: train on all other folds, validate on fold k
3. Average the K validation errors
4. Select λ minimizing average validation error

Common choices: $K = 5$ or $K = 10$. Setting $K = N$ gives **leave-one-out cross-validation**.

The One Standard Error Rule: A common heuristic is to choose the simplest model (largest λ) whose error is within one standard error of the minimum. This favors simpler models when differences are within noise.

3.6 Early Stopping: Implicit Regularization

For iterative optimization algorithms (gradient descent), there's another form of regularization: **early stopping**.

The idea: don't run optimization to convergence. Instead, monitor performance on a validation set and stop when validation error starts increasing.

Why it works: Iterative algorithms typically start from small/zero weights and gradually increase them. Stopping early keeps weights from growing too large—similar to explicit L2 regularization!

Early stopping is widely used in deep learning, where explicit regularization formulas may not be available.

Lecture 1 Summary

1. **MLE** finds parameters that maximize probability of observed data—equivalently, minimize NLL.
2. **MLE equals minimizing KL divergence** between empirical distribution and model distribution.
3. **MLE for common distributions:**
   - Bernoulli: $\hat{\theta} = N_1/N$ (sample proportion)
   - Gaussian: $\hat{\mu} = \bar{y}, \hat{\sigma}^2 = \frac{1}{N}\sum(y_n - \bar{y})^2$
   - Linear regression: $\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$
4. **MLE can overfit** with small samples—may assign zero probability to unseen events.
5. **Regularization** adds a complexity penalty: $\mathcal{L} = \text{NLL} + \lambda C(\boldsymbol{\theta})$
6. **Bias-variance tradeoff:** Regularization increases bias but decreases variance, often reducing overall MSE.
7. **Cross-validation** is used to select regularization strength λ.

LECTURE 2: MAP Estimation and Bayesian Inference

---

4. Maximum A Posteriori (MAP) Estimation

4.1 The Bayesian Interpretation of Regularization

In Lecture 1, we introduced regularization as adding a penalty $\lambda C(\boldsymbol{\theta})$. But where does this penalty come

from? The Bayesian perspective provides a principled answer.

Suppose we interpret the complexity penalty as the negative log of a **prior distribution**:

$$C(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta})$$

Then our regularized objective becomes:

$$\mathcal{L}(\boldsymbol{\theta}) = -\log p(\mathcal{D}|\boldsymbol{\theta}) - \log p(\boldsymbol{\theta})$$

Using the property $\log(ab) = \log a + \log b$:

$$\mathcal{L}(\boldsymbol{\theta}) = -\log \left[p(\mathcal{D}|\boldsymbol{\theta}) \cdot p(\boldsymbol{\theta})\right]$$

By Bayes' rule, $p(\boldsymbol{\theta}|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta}) \cdot p(\boldsymbol{\theta})$, so:

$$\mathcal{L}(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta}|\mathcal{D}) + \text{const}$$

Minimizing this is equivalent to **maximizing the log posterior**!

4.2 MAP Estimation Defined

Definition:

$$\boxed{\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\mathcal{D}) = \arg\max_{\boldsymbol{\theta}} \left[\log p(\mathcal{D}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\right]}$$

Using Bayes' rule:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathcal{D})} \propto p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

The denominator $p(\mathcal{D})$ is constant with respect to $\boldsymbol{\theta}$, so we can ignore it for optimization.

> **Interpretation:** MAP estimation finds the **mode** (peak) of the posterior distribution—the single most probable parameter value given both the data and our prior beliefs.

MLE as a Special Case: If we use a uniform (flat) prior $p(\boldsymbol{\theta}) \propto 1$, then $\log p(\boldsymbol{\theta}) = \text{const}$, and MAP reduces to MLE:

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \hat{\boldsymbol{\theta}}_{\text{MLE}} \quad \text{when } p(\boldsymbol{\theta}) \propto 1$$

4.3 Connecting Priors to Regularizers

Different prior distributions correspond to different regularization penalties:

Prior $p(\boldsymbol{\theta})$	$-\log p(\boldsymbol{\theta})$	**Regularizer**	**Method Name**
Uniform	const	None	MLE
$\mathcal{N}(\mathbf{0}, \frac{1}{\lambda}\mathbf{I})$	$\frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \text{const}$	L2	Ridge Regression
Laplace$(0, \frac{1}{\lambda})$	$\lambda\|\mathbf{w}\|_1 + \text{const}$	L1	Lasso
Beta(α, β)	$-(\alpha-1)\log\theta - (\beta-1)\log(1-\theta)$	—	Smoothed Bernoulli

> **Key Insight:** Regularization is not an ad-hoc trick—it corresponds to encoding prior beliefs about parameters! L2 regularization says "I believe weights are probably small, drawn from a Gaussian centered at zero."

4.4 MAP Example: Bernoulli with Beta Prior

Setup:

- Likelihood: $p(\mathcal{D}|\theta) = \theta^{N_1}(1-\theta)^{N_0}$
- Prior: $p(\theta) = \text{Beta}(\theta|\alpha,\beta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$

The Beta distribution is defined on $[0,1]$, making it a natural prior for probabilities. The parameters α and β control the shape:

- $\alpha = \beta = 1$: Uniform distribution
- $\alpha = \beta > 1$: Peaked at 0.5 (favors fair coins)
- $\alpha > \beta$: Peaked toward 1 (favors heads)
- $\alpha, \beta < 1$: U-shaped (favors extreme values)

Posterior:

$$p(\theta|\mathcal{D}) \propto \theta^{N_1}(1-\theta)^{N_0} \cdot \theta^{\alpha-1}(1-\theta)^{\beta-1} = \theta^{N_1+\alpha-1}(1-\theta)^{N_0+\beta-1}$$

This is another Beta distribution:

$$p(\theta|\mathcal{D}) = \text{Beta}(\theta|N_1 + \alpha, N_0 + \beta)$$

MAP Estimate:

The mode of $\text{Beta}(a, b)$ is $\frac{a-1}{a+b-2}$ (for $a, b > 1$), so:

$$\boxed{\hat{\theta}_{\text{MAP}} = \frac{N_1 + \alpha - 1}{N_1 + N_0 + \alpha + \beta - 2}}$$

Special Case: Add-One Smoothing

Setting $\alpha = \beta = 2$ gives:

$$\hat{\theta}_{\text{MAP}} = \frac{N_1 + 1}{N + 2}$$

This is called **Laplace smoothing** or **add-one smoothing**—a simple but widely used technique.

Numerical Example:

Observe 3 heads in 3 tosses:

- MLE: $\hat{\theta}_{\text{MLE}} = \frac{3}{3} = 1.0$ (predicts only heads forever!)
- MAP with Beta(2,2): $\hat{\theta}_{\text{MAP}} = \frac{3+1}{3+2} = \frac{4}{5} = 0.8$ (reasonable!)

The prior "pulls" the estimate away from the extreme value of 1.

---

5. Full Bayesian Inference

5.1 Beyond Point Estimates

Both MLE and MAP give **point estimates**—single "best" parameter values. But they discard information about **uncertainty** in our estimates.

Consider two scenarios after observing 6 heads in 10 flips:

1. Strong prior knowledge that the coin is fair
2. No prior knowledge about the coin

Both might give similar point estimates, but our **confidence** in those estimates should differ!

The Bayesian Solution: Instead of finding a single best $\boldsymbol{\theta}$, maintain the entire **posterior distribution** $p(\boldsymbol{\theta}|\mathcal{D})$. This distribution encodes:

- The most likely parameter values (the peak/mode)
- Our uncertainty about parameters (the spread)
- The probability of any particular parameter value

5.2 Bayes' Rule for Parameter Inference

$$\boxed{p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})}{p(\mathcal{D})} = \frac{p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})}{\int p(\boldsymbol{\theta}')p(\mathcal{D}|\boldsymbol{\theta}')d\boldsymbol{\theta}'}}$$

Term	Name	Role
$p(\boldsymbol{\theta})$	**Prior**	Our beliefs before seeing data
$p(\mathcal{D}\|\|\boldsymbol{\theta})$	**Likelihood**	How probable is data given parameters
$p(\boldsymbol{\theta}\|\|\mathcal{D})$	**Posterior**	Updated beliefs after seeing data
$p(\mathcal{D})$	**Marginal likelihood** (evidence)	Normalizing constant; probability of data under the model

The marginal likelihood $p(\mathcal{D}) = \int p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})d\boldsymbol{\theta}$ is often the computational bottleneck—it requires integrating over all possible parameter values.

5.3 Conjugate Priors

A prior $p(\boldsymbol{\theta}) \in \mathcal{F}$ is **conjugate** to a likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ if the posterior is in the same family: $p(\boldsymbol{\theta}|\mathcal{D}) \in \mathcal{F}$.

Why use conjugate priors?

1. **Closed-form posteriors:** No numerical integration needed
2. **Interpretable updates:** Prior parameters often act as "pseudo-observations"
3. **Sequential updating:** Easy to incorporate new data incrementally
4. **Analytical marginal likelihood:** The normalizing constant $p(\mathcal{D})$ has a closed form

Common Conjugate Pairs:

Likelihood	Conjugate Prior	Posterior
Bernoulli/Binomial	Beta	Beta
Categorical/Multinomial	Dirichlet	Dirichlet
Gaussian (known σ^2)	Gaussian	Gaussian
Gaussian (known μ)	Inverse-Gamma	Inverse-Gamma
Poisson	Gamma	Gamma
Exponential	Gamma	Gamma

5.4 The Beta-Bernoulli Model in Detail

This is the canonical example of Bayesian inference with conjugate priors.

Setup:

- Data: N coin flips yielding N_1 heads, N_0 tails
- Likelihood: $p(\mathcal{D}|\theta) = \theta^{N_1}(1-\theta)^{N_0}$
- Prior: $p(\theta) = \text{Beta}(\theta|\tilde{\alpha}, \tilde{\beta})$

Posterior:

$$\boxed{p(\theta|\mathcal{D}) = \text{Beta}(\theta|\hat{\alpha}, \hat{\beta}) \quad \text{where} \quad \hat{\alpha} = \tilde{\alpha} + N_1, \quad \hat{\beta} = \tilde{\beta} + N_0}$$

> **Interpretation of Prior Parameters:** The prior parameters $(\tilde{\alpha}, \tilde{\beta})$ act as **pseudo-counts**—as if we had already observed $\tilde{\alpha} - 1$ heads and $\tilde{\beta} - 1$ tails before seeing the actual data. The prior "strength" or **equivalent sample size** is $\tilde{N} = \tilde{\alpha} + \tilde{\beta}$.

Posterior Mean:

$$\bar{\theta} = \mathbb{E}[\theta|\mathcal{D}] = \frac{\hat{\alpha}}{\hat{\alpha} + \hat{\beta}} = \frac{\tilde{\alpha} + N_1}{\tilde{\alpha} + \tilde{\beta} + N}$$

We can rewrite this as a weighted average:

$$\bar{\theta} = \underbrace{\frac{\tilde{N}}{\tilde{N}+N}}_{\text{weight on prior}} \cdot \underbrace{\frac{\tilde{\alpha}}{\tilde{N}}}_{\text{prior mean}} + \underbrace{\frac{N}{\tilde{N}+N}}_{\text{weight on data}} \cdot \underbrace{\frac{N_1}{N}}_{\text{MLE}}$$

As $N \to \infty$, the weight on the prior goes to zero and $\bar{\theta} \to \hat{\theta}_{\text{MLE}}$. The data eventually overwhelms any prior.

Posterior Variance:

$$\text{Var}[\theta|\mathcal{D}] = \frac{\hat{\alpha}\hat{\beta}}{(\hat{\alpha}+\hat{\beta})^2(\hat{\alpha}+\hat{\beta}+1)}$$

For large N, this simplifies to approximately:

$$\text{Var}[\theta|\mathcal{D}] \approx \frac{\hat{\theta}(1-\hat{\theta})}{N}$$

The **standard error** is $\text{se}(\theta) = \sqrt{\text{Var}[\theta|\mathcal{D}]} \approx \sqrt{\frac{\hat{\theta}(1-\hat{\theta})}{N}}$

> Uncertainty decreases at rate $1/\sqrt{N}$—we need 4x more data to halve our uncertainty.

5.5 The Gaussian-Gaussian Model

Setup:

- Data: $y_1, \ldots, y_N \sim \mathcal{N}(\mu, \sigma^2)$ with σ^2 known
- Prior: $p(\mu) = \mathcal{N}(\mu|\tilde{m}, \tilde{\tau}^2)$

Posterior:

$$p(\mu|\mathcal{D}) = \mathcal{N}(\mu|\hat{m}, \hat{\tau}^2)$$

It's cleaner to work with **precision** (inverse variance). Let $\lambda = 1/\sigma^2$ be the data precision and $\tilde{\lambda} = 1/\tilde{\tau}^2$ be the prior precision.

Posterior precision:

$$\boxed{\hat{\lambda} = \tilde{\lambda} + N\lambda}$$

Posterior mean:

$$\boxed{\hat{m} = \frac{\tilde{\lambda}\tilde{m} + N\lambda\bar{y}}{\hat{\lambda}} = \frac{\tilde{\lambda}}{\hat{\lambda}}\tilde{m} + \frac{N\lambda}{\hat{\lambda}}\bar{y}}$$

> **Interpretation:**
>
> - Posterior precision = prior precision + data precision (precisions add!)
> - Posterior mean = precision-weighted average of prior mean and sample mean
> - As $N \to \infty$, posterior concentrates on MLE: $\hat{m} \to \bar{y}$

Example: If prior is $\mathcal{N}(0, 1)$ and we observe one data point $y = 3$ with noise variance $\sigma^2 = 1$:

- Prior precision: $\tilde{\lambda} = 1$
- Data precision: $\lambda = 1$, so $N\lambda = 1$
- Posterior precision: $\hat{\lambda} = 1 + 1 = 2$, so $\hat{\tau}^2 = 0.5$
- Posterior mean: $\hat{m} = \frac{1 \cdot 0 + 1 \cdot 3}{2} = 1.5$

The posterior mean is pulled toward zero by the prior, and uncertainty is reduced.

5.6 Posterior Predictive Distribution

A key advantage of full Bayesian inference: instead of predicting with a single point estimate, we **marginalize out** parameter uncertainty:

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

This is called the **posterior predictive distribution**. It accounts for two sources of uncertainty:

1. **Aleatoric uncertainty:** Inherent randomness in the data-generating process (even if we knew $\boldsymbol{\theta}$ exactly)
2. **Epistemic uncertainty:** Our uncertainty about $\boldsymbol{\theta}$ (reducible with more data)

The posterior predictive is sometimes called **Bayes Model Averaging (BMA)**—we average predictions over all possible parameter values, weighted by their posterior probability.

Contrast with Plug-in Prediction:

The plug-in approach uses $p(y|\hat{\boldsymbol{\theta}})$, which ignores epistemic uncertainty. This leads to overconfident predictions, especially with small samples.

5.7 Posterior Predictive for Beta-Bernoulli

Predicting the next coin flip:

$$p(y=1|\mathcal{D}) = \int_0^1 \theta \cdot \text{Beta}(\theta|\hat{\alpha},\hat{\beta})d\theta = \mathbb{E}[\theta|\mathcal{D}] = \frac{\hat{\alpha}}{\hat{\alpha}+\hat{\beta}}$$

With a uniform prior Beta(1,1):

$$\boxed{p(y=1|\mathcal{D}) = \frac{N_1+1}{N+2}}$$

This is **Laplace's rule of succession**. After seeing 3 heads in 3 tosses:

- Plug-in (MLE): $P(y=1) = 1.0$ (certain heads)
- Posterior predictive: $P(y=1) = \frac{4}{5} = 0.8$ (reasonable uncertainty)

The Bayesian approach automatically provides smoothing—it never predicts probability 0 or 1 for events that haven't been conclusively ruled out.

Predicting multiple future flips:

If we want to predict M future flips, the posterior predictive follows the **Beta-Binomial distribution**:

$$p(y|\mathcal{D},M) = \text{BetaBin}(y|M,\hat{\alpha},\hat{\beta}) = \binom{M}{y}\frac{B(y+\hat{\alpha},M-y+\hat{\beta})}{B(\hat{\alpha},\hat{\beta})}$$

where $B(\cdot,\cdot)$ is the Beta function. This distribution has heavier tails than a Binomial with fixed θ, reflecting our parameter uncertainty.

5.8 Credible Intervals

A **100(1-α)% credible interval** is a region $\mathcal{C} = (\ell, u)$ containing $1-\alpha$ of the posterior probability:

$$P(\ell \leq \theta \leq u|\mathcal{D}) = 1-\alpha$$

Types of credible intervals:

1. **Central/Equal-tailed interval:** Puts probability $\alpha/2$ in each tail. Computed using quantiles: $\ell = F^{-1}(\alpha/2), u = F^{-1}(1-\alpha/2)$ where F is the posterior CDF.

2. **Highest Posterior Density (HPD) interval:** The narrowest interval containing $1 - \alpha$ probability. Every point inside has higher posterior density than any point outside. Preferred for skewed posteriors.

Example: For a Gaussian posterior, the 95% central credible interval is approximately $\hat{m} \pm 1.96 \cdot \hat{\tau}$ (often approximated as $\hat{m} \pm 2\hat{\tau}$).

> **Bayesian Credible Interval vs. Frequentist Confidence Interval:**
> A 95% credible interval means: "Given the observed data, there is 95% probability that θ lies in this interval."
> A 95% confidence interval means: "If we repeated the experiment many times and computed an interval each time, 95% of those intervals would contain the true θ."
> The Bayesian interpretation is usually what practitioners actually want!

5.9 The Dirichlet-Multinomial Model

This generalizes Beta-Bernoulli from binary to K-ary outcomes.

Setup:

- Data: Counts $N_1, \ldots, N_K$ for K categories
- Likelihood: $p(\mathcal{D}|\boldsymbol{\theta}) = \prod_{k=1}^{K} \theta_k^{N_k}$
- Prior: $p(\boldsymbol{\theta}) = \text{Dir}(\boldsymbol{\theta}|\tilde{\boldsymbol{\alpha}}) \propto \prod_{k=1}^{K} \theta_k^{\tilde{\alpha}_k - 1}$

The Dirichlet distribution is defined on the probability simplex $\{\boldsymbol{\theta} : \theta_k \geq 0, \sum_k \theta_k = 1\}$.

Posterior:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \text{Dir}(\boldsymbol{\theta}|\hat{\boldsymbol{\alpha}}) \quad \text{where} \quad \hat{\alpha}_k = \tilde{\alpha}_k + N_k$$

Posterior mean:

$$\bar{\theta}_k = \frac{\hat{\alpha}_k}{\sum_{k'} \hat{\alpha}_{k'}} = \frac{\tilde{\alpha}_k + N_k}{\tilde{\alpha}_0 + N}$$

where $\tilde{\alpha}_0 = \sum_k \tilde{\alpha}_k$ is the prior strength.

ML Application: Dirichlet-Multinomial is used in:

- Naive Bayes text classification (word count smoothing)
- Topic models (LDA)
- Any model with categorical distributions

5.10 Comparison: MLE vs MAP vs Full Bayes

Aspect	MLE	MAP	Full Bayes
Output	Point estimate $\hat{\boldsymbol{\theta}}$	Point estimate $\hat{\boldsymbol{\theta}}$	Distribution $p(\boldsymbol{\theta} \| \mathcal{D})$
Uses prior?	No	Yes	Yes
Quantifies uncertainty?	No	No	Yes
Prediction	$p(y \| \hat{\boldsymbol{\theta}})$	$p(y \| \hat{\boldsymbol{\theta}})$	$\int p(y \| \boldsymbol{\theta}) p(\boldsymbol{\theta} \| \mathcal{D}) d\boldsymbol{\theta}$
Overfitting risk	High	Reduced	Further reduced
Computation	Usually easy	Usually easy	Often hard
Interpretation	Most likely data	Most probable parameter	Full uncertainty

When to use which:

- **MLE:** Large datasets, simple models, when prior information is unavailable
- **MAP:** Moderate datasets, when regularization is needed, when you have prior information
- **Full Bayes:** Small datasets, when uncertainty quantification is important, for model comparison

---

5.11 Computational Challenges in Bayesian Inference

Computing the posterior requires the marginal likelihood:

$$p(\mathcal{D}) = \int p(\boldsymbol{\theta}) p(\mathcal{D} | \boldsymbol{\theta}) d\boldsymbol{\theta}$$

This integral is tractable only in special cases:

- Conjugate priors with closed-form posteriors
- Discrete parameters with small state spaces

Approximation methods (covered in later lectures):

1. **Grid approximation:** Discretize parameter space, evaluate on a grid. Only practical for 1-3 dimensions.

2. **Laplace approximation:** Approximate posterior as Gaussian centered at the MAP estimate: $$p(\boldsymbol{\theta}|\mathcal{D}) \approx \mathcal{N}(\boldsymbol{\theta}|\hat{\boldsymbol{\theta}}_{\text{MAP}}, \mathbf{H}^{-1})$$ where $\mathbf{H}$ is the Hessian of the negative log-posterior at the mode.
3. **Variational Inference (Week 5):** Approximate posterior with a tractable distribution $q(\boldsymbol{\theta})$ by minimizing KL divergence.
4. **Markov Chain Monte Carlo (Week 6):** Generate samples $\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \ldots$ from the posterior without computing $p(\mathcal{D})$.

---

6. Key Formulas Reference

MLE

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \arg\max_{\boldsymbol{\theta}} \sum_{n=1}^{N} \log p(y_n|\mathbf{x}_n, \boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} \text{NLL}(\boldsymbol{\theta})$$

MAP

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} \left[\sum_{n=1}^{N} \log p(y_n|\mathbf{x}_n, \boldsymbol{\theta}) + \log p(\boldsymbol{\theta}) \right]$$

Posterior (Bayes' Rule)

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})}{\int p(\boldsymbol{\theta}')p(\mathcal{D}|\boldsymbol{\theta}')d\boldsymbol{\theta}'} \propto p(\boldsymbol{\theta})p(\mathcal{D}|\boldsymbol{\theta})$$

Posterior Predictive

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

Beta-Bernoulli

- Prior: $\text{Beta}(\theta|\tilde{\alpha}, \tilde{\beta})$
- Posterior: $\text{Beta}(\theta|\tilde{\alpha} + N_1, \tilde{\beta} + N_0)$
- Posterior mean: $\frac{\tilde{\alpha}+N_1}{\tilde{\alpha}+\tilde{\beta}+N}$

- Posterior predictive: $\frac{\tilde{\alpha}+N_1}{\tilde{\alpha}+\tilde{\beta}+N}$

Gaussian-Gaussian (known variance)

- Posterior precision: $\hat{\lambda} = \tilde{\lambda} + N\lambda$
- Posterior mean: $\hat{m} = \frac{\tilde{\lambda}\tilde{m}+N\lambda\bar{y}}{\hat{\lambda}}$

Ridge Regression

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

---

7. Practice Problems

Problem 1: MLE Computation

You flip a coin 20 times and observe 14 heads.

(a) What is the MLE for $\theta = P(\text{heads})$?

(b) Using a $\text{Beta}(2, 2)$ prior, what is the MAP estimate?

(c) Using the same prior, what is the posterior mean?

(d) Compute the posterior variance. How does it compare to the variance of the MLE (treating MLE as an estimator with variance $\frac{\theta(1-\theta)}{N}$)?

Problem 2: Gaussian MLE

Given data points $\{1, 3, 5, 7, 9\}$:

(a) Compute the MLE estimates $\hat{\mu}$ and $\hat{\sigma}^2$

(b) Compute the unbiased variance estimate $\hat{\sigma}^2_{\text{unb}}$

(c) What is the ratio $\hat{\sigma}^2_{\text{MLE}} / \hat{\sigma}^2_{\text{unb}}$? How does this ratio depend on N?

Problem 3: Regularization and Priors

Consider ridge regression with objective:

$$\mathcal{L}(\mathbf{w}) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$$

(a) What prior on $\mathbf{w}$ does this correspond to? Write out the prior distribution explicitly.

(b) Derive the closed-form solution for $\hat{\mathbf{w}}$.

(c) What happens to $\hat{\mathbf{w}}$ as $\lambda \to 0$? As $\lambda \to \infty$?

(d) If $\mathbf{X}^T\mathbf{X}$ is singular, can you still compute the ridge solution? Explain.

Problem 4: Posterior Predictive

Using a $\text{Beta}(1, 1)$ prior (uniform), you observe 8 heads in 10 coin flips.

(a) What is the posterior distribution?

(b) What is $P(y = 1|\mathcal{D})$ for the next flip?

(c) Compare this to the MLE plug-in prediction.

(d) What is the 95% credible interval for θ? (You may state this in terms of Beta quantiles.)

Problem 5: Sequential Updating

One advantage of conjugate priors is easy sequential updating. Suppose you start with a Beta(1,1) prior on a coin's bias.

(a) After seeing 3 heads, what is your posterior?

(b) Now you see 2 more tails. What is your new posterior?

(c) Show that this gives the same answer as if you had processed all 5 observations at once.

---

8. Key Takeaways

From Lecture 1:

1. **MLE** maximizes $p(\mathcal{D}|\boldsymbol{\theta})$, equivalently minimizes NLL, equivalently minimizes KL divergence from empirical distribution.
2. **MLE formulas** for common distributions:
   - Bernoulli: sample proportion
   - Gaussian: sample mean and variance
   - Linear regression: OLS formula
3. **Overfitting** occurs when MLE assigns zero probability to unseen events.
4. **Regularization** $= \text{NLL} + \lambda C(\boldsymbol{\theta})$ trades data fit for simplicity.
5. **Bias-variance tradeoff**: regularization increases bias but decreases variance.

From Lecture 2:

6. **MAP estimation** = mode of posterior = regularized MLE with $C(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta})$.

7. **Prior-regularizer correspondence**: Gaussian prior ↔ L2, Laplace prior ↔ L1.
8. **Conjugate priors** give closed-form posteriors with interpretable pseudo-count updates.
9. **Full Bayesian inference** maintains entire posterior, enabling uncertainty quantification.
10. **Posterior predictive** marginalizes over parameters, avoiding overconfident predictions.
11. **Credible intervals** give direct probability statements about parameters.
12. **As** $N \to \infty$: prior influence vanishes, all methods converge to MLE.

---

9. Looking Ahead

Next Lecture (Week 4): Linear and Logistic Regression

- Apply MLE/MAP framework to supervised learning
- Maximum likelihood for linear regression = least squares
- Maximum likelihood for logistic regression = cross-entropy loss
- Regularized regression: Ridge and Lasso

Week 5: Variational Inference

- Approximate intractable posteriors with tractable distributions
- The Evidence Lower Bound (ELBO)
- Mean-field approximations

Week 6: Monte Carlo Sampling

- MCMC methods for sampling from posteriors
- Metropolis-Hastings algorithm
- Hamiltonian Monte Carlo

These notes draw from Chapter 4 of Murphy, K. P. (2022). Probabilistic Machine Learning: An Introduction. MIT Press.