Bayesian Statistics

Lecture Notes

Master M2 — 2025–2026

Yaë Ulrich Gaba

"Probability is the very guide of life."
— Marcus Tullius Cicero

March 25, 2026

Contents

Preface

Objectives of this course

Bayesian statistics is one of the two major paradigms of statistical inference, alongside the frequentist approach. Founded on Bayes' theorem, it provides a coherent framework for quantifying uncertainty, incorporating prior knowledge, and updating beliefs in light of data.

This course is aimed at Master-level students in applied mathematics, data science, and artificial intelligence. It assumes prior mastery of probability theory, elementary mathematical statistics, and linear algebra.

Course organization

The course is structured into eleven chapters, grouped into four thematic parts:

1. **Foundations** (Chapters 1–5): Bayesian paradigm, conjugate families, estimation, testing, and prediction.

2. **Advanced models** (Chapter 6): hierarchical models.

3. **Computational methods** (Chapters 7–9): MCMC (Metropolis–Hastings, Gibbs sampler) and variational inference.

4. **Extensions and applications** (Chapters 10–11): Bayesian nonparametrics and practical applications.

Pedagogical philosophy

Each chapter alternates between:

- rigorous **theoretical developments** (definitions, theorems, proofs),
- detailed **worked examples** with explicit calculations,
- **numerical implementations** in Python (PyMC, NumPy, SciPy),
- **exercises** of increasing difficulty.

Bayesian *versus* frequentist approaches

Before delving into the detailed study, let us contrast the two paradigms:

Aspect	**Frequentist**	**Bayesian**
Parameter θ	Fixed, unknown	Random variable
Probability	Limiting frequency	Degree of belief
Uncertainty	Confidence intervals	Credible intervals
Prior information	Not formally used	Integrated via prior
Main result	Estimator $\hat{\theta}(x)$	Posterior $\pi(\theta \mid x)$
Coherence	May violate likelihood principle	Coherence guaranteed
Small samples	Possible difficulties	Naturally suited
Computation	Often analytic	Often numerical (MCMC)

The founding theorem

The entire Bayesian paradigm rests on a single formula:

Bayes' Theorem

Let $\theta \in \Theta$ be a parameter with prior distribution $\pi(\theta)$ and $x = (x_1, \ldots, x_n)$ a sample with likelihood $L(\theta \mid x)$. The **posterior distribution** is:

$$\pi(\theta \mid x) = \frac{L(\theta \mid x)\, \pi(\theta)}{\int_\Theta L(\theta \mid x)\, \pi(\theta)\, d\theta} \propto L(\theta \mid x)\, \pi(\theta).$$

Notation and conventions

Throughout this course, we adopt the following conventions:

- X denotes a random variable, x an observed value.
- θ is the parameter of interest, Θ the parameter space.
- $\pi(\theta)$ is the prior distribution.
- $\pi(\theta \mid x)$ is the posterior distribution.
- $L(\theta \mid x) = f(x \mid \theta)$ is the likelihood.
- $m(x) = \int L(\theta \mid x)\, \pi(\theta)\, d\theta$ is the marginal likelihood (evidence).
- $\mathbb{E}[\cdot]$, $\mathrm{Var}[\cdot]$, $\mathrm{Cov}[\cdot]$ denote expectation, variance, and covariance.
- $\mathbb{P}(\cdot)$ denotes probability.
- $\mathrm{KL}(p \| q)$ denotes the Kullback–Leibler divergence.
- $\|\cdot\|$ denotes the Euclidean norm.

Commonly used distributions

Distribution	**Notation**	**Parameters**
Bernoulli	$\mathrm{Ber}(p)$	$p \in [0,1]$
Binomial	$\mathrm{Bin}(n,p)$	$n \in \mathbb{N}$, $p \in [0,1]$
Poisson	$\mathrm{Poi}(\lambda)$	$\lambda > 0$
Normal	$\mathcal{N}(\mu, \sigma^2)$	$\mu \in \mathbb{R}$, $\sigma^2 > 0$
Gamma	$\mathrm{Ga}(\alpha, \beta)$	$\alpha, \beta > 0$
Beta	$\mathrm{Be}(a, b)$	$a, b > 0$
Dirichlet	$\mathrm{Dir}(\boldsymbol{\alpha})$	$\alpha_k > 0$
Student-t	$t_\nu(\mu, \sigma^2)$	$\nu > 0$
Wishart	$\mathcal{W}_p(\nu, \boldsymbol{\Sigma})$	$\nu > p - 1$
Inverse-Gamma	$\mathrm{IG}(\alpha, \beta)$	$\alpha, \beta > 0$

Historical milestones

- **1763**: Posthumous publication of Thomas Bayes' theorem by Richard Price.
- **1812**: Pierre-Simon de Laplace formalizes and generalizes the method in *Théorie analytique des probabilités*.
- **1937**: Bruno de Finetti develops the notion of subjective probability and the exchangeability theorem.
- **1954**: Leonard Jimmie Savage publishes *The Foundations of Statistics*, a pillar of Bayesian decision theory.
- **1970s**: Dennis Lindley and others establish the modern foundations.
- **1990s**: The MCMC revolution — Metropolis–Hastings becomes the workhorse of Bayesian computation.
- **2000s–2020s**: Variational inference, Gaussian processes, Bayesian deep learning.

Software and libraries

The practical implementations in this course use:

- **Python 3.10+** as the primary language.
- **PyMC** (v5+) for Bayesian modeling and MCMC.
- **ArviZ** for chain diagnostics and visualization.
- **NumPy / SciPy** for numerical computation.
- **Matplotlib / Seaborn** for plotting.

Installing dependencies

```
# Recommended installation
# pip install pymc arviz numpy scipy matplotlib seaborn

import pymc as pm
import arviz as az
import numpy as np
import matplotlib.pyplot as plt

print(f"PyMC version: {pm.__version__}")
print(f"ArviZ version: {az.__version__}")
```

How to use this course

Important prerequisites

This course assumes that the student is proficient in:

1. Probability theory (random variables, classical distributions, modes of convergence).
2. Mathematical statistics (estimation, testing, likelihood).
3. Linear algebra (matrices, decompositions, quadratic forms).
4. Python programming (basic syntax, NumPy).

Guiding thread

The guiding idea of this course is simple: *all inference is a problem of updating a probability distribution.* We start from a prior belief $\pi(\theta)$, we observe data x, and we obtain a posterior belief $\pi(\theta \mid x)$. This unified vision underlies all the methods presented herein.

Main references

- A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, D.B. Rubin, *Bayesian Data Analysis* (3rd ed.), 2013.
- C.P. Robert, *The Bayesian Choice* (2nd ed.), 2007.
- J.K. Kruschke, *Doing Bayesian Data Analysis* (2nd ed.), 2015.
- D. Barber, *Bayesian Reasoning and Machine Learning*, 2012.
- K.P. Murphy, *Machine Learning: A Probabilistic Perspective*, 2012.
- P. Hoff, *A First Course in Bayesian Statistical Methods*, 2009.

The author
March 2026

Chapter 1

Bayesian Paradigm

In 1763, two years after his death, the Reverend Thomas Bayes saw his essay published posthumously by the Royal Society of London. The paper proposed a method for inverting probabilities: given observed data, what can we say about the cause that produced them? This seemingly innocent question would spark one of the longest-running debates in all of statistics. On one side, the *frequentists*, for whom parameters are fixed unknown constants. On the other, the *Bayesians*, for whom parameters are random variables endowed with distributions reflecting our state of knowledge. This chapter presents the Bayesian paradigm: its logic, its strengths, and the reasons behind its spectacular modern revival.

Central idea

The Bayesian paradigm treats the parameter θ as a random variable and encodes all uncertainty through probability distributions. Inference consists of moving from the prior $\pi(\theta)$ to the posterior $\pi(\theta \mid x)$ via Bayes' theorem.

1.1 Introduction and motivation

The frequentist approach regards the parameter θ as a fixed unknown constant and evaluates inference procedures by their repeated-sampling properties. The Bayesian approach, by contrast, models θ as a random variable whose distribution reflects our state of knowledge.

This philosophical difference has profound consequences:

- Uncertainty is directly quantified by the posterior distribution.
- Prior information is formally incorporated.
- Any optimal decision can be derived from the posterior.

1.2 Axiomatic foundations

1.2.1 Subjective probability

Definition 1.1 (Subjective probability)**.** A **subjective probability** is a function $\mathbb{P} : \mathcal{F} \to [0, 1]$ satisfying Kolmogorov's axioms, interpreted as the degree of belief of a rational

agent in the occurrence of an event.

De Finetti's coherence axioms show that an agent whose bets are coherent (no *Dutch book*) necessarily behaves as if using probabilities satisfying Kolmogorov's axioms.

Theorem 1.2 (De Finetti's representation theorem)**.** *If $(X_1, X_2, \ldots)$ is an exchangeable sequence of $\{0,1\}$-valued random variables, then there exists a probability measure μ on $[0,1]$ such that:*

$$\mathbb{P}(X_1 = x_1, \ldots, X_n = x_n) = \int_0^1 \theta^{s_n}(1-\theta)^{n-s_n}\, d\mu(\theta),$$

where $s_n = \sum_{i=1}^n x_i$.

Remark 1.3. This theorem justifies the Bayesian model: exchangeability (a weaker assumption than independence) implies the existence of a latent parameter θ with a prior distribution μ.

1.2.2 Savage's axioms

Theorem 1.4 (Savage's theorem, 1954)**.** *Under Savage's seven axioms (ordering, sure-thing principle, etc.), there exist:*

1. *a subjective probability measure $\mathbb{P}$ on the state space,*

2. *a utility function u unique up to affine transformation,*

such that the agent prefers action a to action b if and only if $\mathbb{E}_{\mathbb{P}}[u(a)] > \mathbb{E}_{\mathbb{P}}[u(b)]$.

1.3 Bayes' theorem

Theorem 1.5 (Bayes' theorem)**.** *Let $\theta \in \Theta$ be a parameter with prior $\pi(\theta)$ and $x = (x_1, \ldots, x_n)$ a sample with density $f(x \mid \theta)$. The posterior distribution is:*

$$\pi(\theta \mid x) = \frac{f(x \mid \theta)\,\pi(\theta)}{m(x)}, \qquad m(x) = \int_\Theta f(x \mid \theta)\,\pi(\theta)\,d\theta.$$

Proof. By the rule for conditional densities:

$$\pi(\theta \mid x) = \frac{f(x,\theta)}{f(x)} = \frac{f(x \mid \theta)\,\pi(\theta)}{\int_\Theta f(x \mid \theta)\,\pi(\theta)\,d\theta}.$$

□

Components of Bayes' theorem

$$\begin{aligned} \text{Prior :} \quad & \pi(\theta) \\ \text{Likelihood :} \quad & L(\theta \mid x) = f(x \mid \theta) \\ \text{Evidence :} \quad & m(x) = \int_\Theta L(\theta \mid x)\, \pi(\theta)\, d\theta \\ \text{Posterior :} \quad & \pi(\theta \mid x) \propto L(\theta \mid x)\, \pi(\theta) \end{aligned}$$

1.4 Choice of the prior distribution

1.4.1 Types of priors

Definition 1.6 (Informative prior)**.** An **informative prior** is a distribution $\pi(\theta)$ that concentrates its mass on a restricted region of Θ, reflecting substantial prior knowledge.

Definition 1.7 (Vague (noninformative) prior)**.** A **vague** or **noninformative prior** is a distribution $\pi(\theta)$ that seeks to let the data dominate the inference. Examples: uniform prior, Jeffreys prior.

Definition 1.8 (Jeffreys prior)**.** The **Jeffreys prior** is defined by:

$$\pi_J(\theta) \propto \sqrt{\det I(\theta)},$$

where $I(\theta)$ is the Fisher information matrix.

Proposition 1.9 (Invariance of Jeffreys prior)**.** The Jeffreys prior is invariant under reparameterization: if $\phi = g(\theta)$ is a differentiable bijection, then $\pi_J(\phi) \propto \sqrt{\det I_\phi(\phi)}$.

Proof. Let $\phi = g(\theta)$. By the change-of-variable rule and the transformation of Fisher information:

$$I_\phi(\phi) = \left(\frac{d\theta}{d\phi}\right)^2 I(\theta),$$

whence $\sqrt{I_\phi(\phi)} = \left|\frac{d\theta}{d\phi}\right| \sqrt{I(\theta)}$, which is exactly the Jacobian of the change of variable. □

Example 1.10. For $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ with σ^2 known, the Fisher information for μ is $I(\mu) = n/\sigma^2$, a constant. The Jeffreys prior is therefore $\pi_J(\mu) \propto 1$, i.e., the (improper) uniform prior on $\mathbb{R}$.

1.4.2 Reference prior

Definition 1.11 (Bernardo's reference prior)**.** The **reference prior** maximizes the expected Kullback–Leibler information between the prior and the posterior:

$$\pi^*(\theta) = \arg\max_\pi \mathbb{E}_x \left[\mathrm{KL}\big(\pi(\theta \mid x) \| \pi(\theta)\big)\right].$$

1.5 Fundamental examples

Example 1.12. Bernoulli–Beta model. Let $X_1, \dots, X_n \overset{\text{iid}}{\sim} \text{Ber}(\theta)$ with $\theta \sim \text{Be}(a, b)$. Setting $s = \sum x_i$:

$$\begin{aligned}\pi(\theta \mid x) &\propto \theta^s(1-\theta)^{n-s} \cdot \theta^{a-1}(1-\theta)^{b-1} \\ &= \theta^{a+s-1}(1-\theta)^{b+n-s-1}.\end{aligned}$$

Hence $\theta \mid x \sim \text{Be}(a+s,\, b+n-s)$.

Example 1.13. Normal–Normal model (known variance). Let $X_1, \dots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ with σ^2 known, and $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$. The posterior is:

$$\mu \mid x \sim \mathcal{N}(\mu_n, \tau_n^2),$$

where

$$\tau_n^2 = \left(\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}\right)^{-1}, \qquad \mu_n = \tau_n^2\left(\frac{\mu_0}{\tau_0^2} + \frac{n\bar{x}}{\sigma^2}\right).$$

Posterior mean as weighted average

$$\mu_n = w \cdot \mu_0 + (1-w) \cdot \bar{x}, \qquad w = \frac{\sigma^2/n}{\tau_0^2 + \sigma^2/n}.$$

As $n \to \infty$, $w \to 0$ and $\mu_n \to \bar{x}$: the posterior converges to the maximum likelihood estimator.

1.6 Asymptotic properties

Theorem 1.14 (Bernstein–von Mises theorem)**.** *Under regularity conditions, the posterior converges in distribution to a Gaussian centered on the MLE:*

$$\pi(\theta \mid x_1, \dots, x_n) \xrightarrow{d} \mathcal{N}\left(\hat{\theta}_{MLE},\ \frac{1}{n}\, I(\theta_0)^{-1}\right),$$

where θ_0 is the true parameter value.

Remark 1.15. This result shows that, for large samples, the choice of prior has negligible impact. In other words, the Bayesian and frequentist approaches converge asymptotically.

Theorem 1.16 (Posterior consistency)**.** *Let θ_0 be the true value. Under regularity conditions (identifiability, prior support containing θ_0), for every neighborhood U of θ_0:*

$$\pi(\theta \in U \mid x_1, \dots, x_n) \xrightarrow{p} 1 \quad \text{as } n \to \infty.$$

1.7 Likelihood principle

Definition 1.17 (Likelihood principle)**.** Two experiments producing the same likelihood function $L(\theta \mid x)$ (up to a multiplicative constant) must lead to the same inference about θ.

Proposition 1.18. Bayesian inference automatically satisfies the likelihood principle, since the posterior depends on x only through $L(\theta \mid x)$.

Frequentist violation

Certain frequentist procedures (such as p-values) violate the likelihood principle because they depend on the sample space and experimental design, not just the observed data.

1.8 Python implementation

Bernoulli–Beta model with PyMC

```
import pymc as pm
import arviz as az
import numpy as np

# Simulated data
np.random.seed(42)
n = 50
theta_true = 0.3
data = np.random.binomial(1, theta_true, size=n)

# Bayesian model
with pm.Model() as model_beta_binom:
    # Beta(2, 5) prior
    theta = pm.Beta("theta", alpha=2, beta=5)
    # Likelihood
    y_obs = pm.Bernoulli("y_obs", p=theta, observed=data)
    # MCMC sampling
    trace = pm.sample(2000, tune=1000, random_seed=42)

# Posterior summary
summary = az.summary(trace, var_names=["theta"],
                     hdi_prob=0.95)
print(summary)

# Analytic verification
a_post = 2 + data.sum()
b_post = 5 + n - data.sum()
print(f"Analytic posterior: Be({a_post}, {b_post})")
print(f"Analytic mean: {a_post/(a_post+b_post):.4f}")
```

Visualizing prior–posterior update

```
import matplotlib.pyplot as plt
from scipy import stats

fig, ax = plt.subplots(figsize=(8, 5))
theta_grid = np.linspace(0, 1, 500)
```

```
# Prior
ax.plot(theta_grid, stats.beta.pdf(theta_grid, 2, 5),
        label="Prior Be(2, 5)", linestyle="--")
# Analytic posterior
ax.plot(theta_grid, stats.beta.pdf(theta_grid, a_post, b_post),
        label=f"Posterior Be({a_post}, {b_post})")
# Normalized likelihood
s = data.sum()
like = theta_grid**s * (1-theta_grid)**(n-s)
like /= like.max() * 3
ax.plot(theta_grid, like, label="Likelihood (norm.)",
        linestyle=":")
ax.axvline(theta_true, color="red", alpha=0.5,
           label=f"True value ({theta_true})")
ax.set_xlabel(r"$\theta$")
ax.set_ylabel("Density")
ax.legend()
ax.set_title("Bayesian update: Bernoulli-Beta")
plt.tight_layout()
plt.savefig("bayes_update.pdf")
```

1.9 Exercises

Exercise 1.1. Let $X_1, \dots, X_n \overset{\text{iid}}{\sim} \text{Poi}(\lambda)$ with prior $\lambda \sim \text{Ga}(\alpha, \beta)$.

1. Determine the posterior distribution of λ.
2. Compute the posterior mean and variance.
3. Show that the posterior mean is a weighted average of the prior mean and the sample mean.

Exercise 1.2. Compute the Jeffreys prior for the $\text{Ber}(\theta)$ model and show that it is $\text{Be}(1/2, 1/2)$.

Exercise 1.3. Prove that in the Normal–Normal model, the posterior precision equals the sum of the prior precision and the data precision. Interpret this result.

Exercise 1.4. Consider two experiments:

1. A coin is tossed $n = 12$ times, yielding $s = 3$ successes.
2. A coin is tossed until the third success, requiring $n = 12$ tosses.

Show that the likelihoods are proportional. Deduce that Bayesian inference is identical, whereas the frequentist p-values differ.

Exercise 1.5. (Numerical) Consider the Normal–Normal model with $\sigma^2 = 1$, $\mu_0 = 0$, $\tau_0^2 = 10$, and a sample of size $n = 5$ drawn from $\mathcal{N}(2, 1)$.

1. Compute the posterior analytically.
2. Verify with PyMC.
3. Plot the prior, posterior, and normalized likelihood on the same graph.

Chapter 2

Prior and Posterior — Conjugate Families

Bayesian inference is elegant in principle: start with a prior, observe data, compute the posterior via Bayes' theorem. But in practice, the posterior can be an intractable integral, impossible to evaluate in closed form. This computational obstacle haunted Bayesian statistics for two centuries—until a beautiful structural shortcut was discovered. If the prior and the likelihood belong to compatible families, the posterior turns out to have the *same form* as the prior, with updated parameters. The prior and posterior are "conjugate" to each other.

This idea of conjugate families was systematised in the 1960s by Howard Raiffa and Robert Schlaifer at Harvard Business School, who catalogued the major conjugate pairs: Beta-Binomial, Gamma-Poisson, Normal-Normal, and others. Each pair provides a complete analytical solution to the Bayesian update, turning what could be an intractable computation into a simple parameter update. Before the advent of MCMC methods in the 1990s, conjugacy was essentially the only way to do Bayesian inference at scale. Even today, conjugate models remain the first tool a Bayesian statistician reaches for—and the lens through which the logic of Bayesian updating is most clearly understood.

Central idea

A conjugate family is a (likelihood, prior) pair such that the posterior belongs to the same parametric family as the prior. This guarantees analytic computations and a transparent interpretation of the Bayesian update.

2.1 Exponential family

Definition 2.1 (Natural exponential family)**.** A family of distributions $\{f(x \mid \theta) : \theta \in \Theta\}$ belongs to the **exponential family** if it can be written as:

$$f(x \mid \theta) = h(x) \exp\big(\eta(\theta)^\top T(x) - A(\theta)\big),$$

where $T(x)$ is the sufficient statistic, $\eta(\theta)$ the natural parameter, and $A(\theta)$ the log-normalizer.

Proposition 2.2 (Properties of the log-normalizer)**.**

1. $\mathbb{E}[T(X)] = \nabla_\eta A(\eta)$.
2. $\mathrm{Var}[T(X)] = \nabla^2_\eta A(\eta)$ (Hessian matrix).

3. $A(\eta)$ is convex.

Theorem 2.3 (Existence of the conjugate prior)**.** *Every exponential family admits a conjugate prior of the form:*

$$\pi(\theta \mid \chi_0, \nu_0) \propto \exp\big(\eta(\theta)^\top \chi_0 - \nu_0\, A(\theta)\big),$$

where χ_0 and ν_0 are hyperparameters. After observing $x_1, \dots, x_n$, the posterior has the same form with:

$$\chi_n = \chi_0 + \sum_{i=1}^{n} T(x_i), \qquad \nu_n = \nu_0 + n.$$

Proof. The sample likelihood is:

$$L(\theta \mid x) = \prod_{i=1}^{n} h(x_i) \cdot \exp\left(\eta(\theta)^\top \sum_{i=1}^{n} T(x_i) - n\, A(\theta)\right).$$

Multiplying by the conjugate prior:

$$\pi(\theta \mid x) \propto \exp\big(\eta(\theta)^\top (\chi_0 + \textstyle\sum T(x_i)) - (\nu_0 + n)\, A(\theta)\big),$$

which has the same form with (χ_n, ν_n). □

2.2 Table of conjugate families

Main conjugate families

Likelihood	Prior	Posterior	Update
$\mathrm{Ber}(\theta)$	$\mathrm{Be}(a,b)$	$\mathrm{Be}(a',b')$	$a' = a + s,\ b' = b + n - s$
$\mathrm{Bin}(m,\theta)$	$\mathrm{Be}(a,b)$	$\mathrm{Be}(a',b')$	$a' = a + \sum x_i,\ b' = b + nm - \sum x_i$
$\mathrm{Poi}(\lambda)$	$\mathrm{Ga}(\alpha,\beta)$	$\mathrm{Ga}(\alpha',\beta')$	$\alpha' = \alpha + \sum x_i,\ \beta' = \beta + n$
$\mathrm{Exp}(\lambda)$	$\mathrm{Ga}(\alpha,\beta)$	$\mathrm{Ga}(\alpha',\beta')$	$\alpha' = \alpha + n,\ \beta' = \beta + \sum x_i$
$\mathcal{N}(\mu, \sigma_0^2)$	$\mathcal{N}(\mu_0, \tau_0^2)$	$\mathcal{N}(\mu_n, \tau_n^2)$	see below
$\mathcal{N}(\mu_0, \sigma^2)$	$\mathrm{IG}(\alpha,\beta)$	$\mathrm{IG}(\alpha',\beta')$	see below
$\mathrm{Mult}(n, \mathbf{p})$	$\mathrm{Dir}(\boldsymbol{\alpha})$	$\mathrm{Dir}(\boldsymbol{\alpha}')$	$\alpha'_k = \alpha_k + x_k$
$\mathrm{Geom}(\theta)$	$\mathrm{Be}(a,b)$	$\mathrm{Be}(a',b')$	$a' = a + n,\ b' = b + \sum x_i$

2.3 Detailed examples

2.3.1 Bernoulli–Beta

Example 2.4. Let $X_1, \dots, X_n \overset{\text{iid}}{\sim} \mathrm{Ber}(\theta)$ with $\theta \sim \mathrm{Be}(a,b)$. The sufficient statistic is $s = \sum x_i$. The posterior is $\mathrm{Be}(a + s, b + n - s)$.

Interpretation: the prior $\mathrm{Be}(a, b)$ is equivalent to $a-1$ virtual "successes" and $b-1$ virtual "failures". The observation adds s successes and $n-s$ failures.

Posterior moments:

$$\mathbb{E}[\theta \mid x] = \frac{a+s}{a+b+n},$$
$$\mathrm{Var}[\theta \mid x] = \frac{(a+s)(b+n-s)}{(a+b+n)^2(a+b+n+1)}.$$

2.3.2 Poisson–Gamma

Example 2.5. Let $X_1, \dots, X_n \overset{\text{iid}}{\sim} \mathrm{Poi}(\lambda)$ with $\lambda \sim \mathrm{Ga}(\alpha, \beta)$. Then:

$$\begin{aligned}\pi(\lambda \mid x) &\propto \lambda^{\sum x_i} e^{-n\lambda} \cdot \lambda^{\alpha-1} e^{-\beta\lambda} \\ &= \lambda^{\alpha+\sum x_i - 1} e^{-(\beta+n)\lambda}.\end{aligned}$$

Hence $\lambda \mid x \sim \mathrm{Ga}(\alpha + \sum x_i, \beta + n)$.

The posterior mean is:

$$\mathbb{E}[\lambda \mid x] = \frac{\alpha + \sum x_i}{\beta + n} = \frac{\beta}{\beta+n} \cdot \frac{\alpha}{\beta} + \frac{n}{\beta+n} \cdot \bar{x}.$$

This is a weighted average of the prior mean α/β and the sample mean $\bar{x}$.

2.3.3 Normal–Normal (known variance)

Theorem 2.6 (Normal–Normal conjugacy)**.** *Let $X_1, \dots, X_n \overset{iid}{\sim} \mathcal{N}(\mu, \sigma^2)$ with σ^2 known and $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$. Then $\mu \mid x \sim \mathcal{N}(\mu_n, \tau_n^2)$ with:*

$$\frac{1}{\tau_n^2} = \frac{1}{\tau_0^2} + \frac{n}{\sigma^2}, \qquad \mu_n = \tau_n^2 \left(\frac{\mu_0}{\tau_0^2} + \frac{n\bar{x}}{\sigma^2} \right).$$

Proof. The log-posterior density (ignoring constants) is:

$$\begin{aligned}\log \pi(\mu \mid x) &\propto -\frac{1}{2\tau_0^2}(\mu - \mu_0)^2 - \frac{n}{2\sigma^2}(\mu - \bar{x})^2 \\ &= -\frac{1}{2}\left(\frac{1}{\tau_0^2} + \frac{n}{\sigma^2} \right) \mu^2 + \left(\frac{\mu_0}{\tau_0^2} + \frac{n\bar{x}}{\sigma^2} \right) \mu + C,\end{aligned}$$

which corresponds to a Gaussian with precision $1/\tau_n^2 = 1/\tau_0^2 + n/\sigma^2$ and mean μ_n. □

2.3.4 Normal–Inverse-Gamma (known mean)

Theorem 2.7 (Variance conjugacy)**.** *Let $X_1, \dots, X_n \overset{iid}{\sim} \mathcal{N}(\mu_0, \sigma^2)$ with μ_0 known and $\sigma^2 \sim IG(\alpha, \beta)$. Then:*

$$\sigma^2 \mid x \sim IG\left(\alpha + \frac{n}{2},\ \beta + \frac{1}{2}\sum_{i=1}^{n} (x_i - \mu_0)^2 \right).$$

2.3.5 Normal–Normal-Inverse-Gamma (general case)

Theorem 2.8 (NIG conjugacy)**.** *Let* $X_1, \dots, X_n \overset{iid}{\sim} \mathcal{N}(\mu, \sigma^2)$ *with joint prior* $(\mu, \sigma^2) \sim NIG(\mu_0, \kappa_0, \alpha_0, \beta_0)$*:*

$$\mu \mid \sigma^2 \sim \mathcal{N}(\mu_0, \sigma^2/\kappa_0), \qquad \sigma^2 \sim IG(\alpha_0, \beta_0).$$

The posterior is $NIG(\mu_n, \kappa_n, \alpha_n, \beta_n)$ *with:*

$$\begin{aligned}
\kappa_n &= \kappa_0 + n, \\
\mu_n &= \frac{\kappa_0 \mu_0 + n\bar{x}}{\kappa_n}, \\
\alpha_n &= \alpha_0 + n/2, \\
\beta_n &= \beta_0 + \tfrac{1}{2}\sum (x_i - \bar{x})^2 + \frac{\kappa_0 n (\bar{x} - \mu_0)^2}{2\kappa_n}.
\end{aligned}$$

2.3.6 Multinomial–Dirichlet

Example 2.9. Let $\mathbf{x} \sim \text{Mult}(n, \mathbf{p})$ with $\mathbf{p} \sim \text{Dir}(\boldsymbol{\alpha})$. The posterior is:

$$\mathbf{p} \mid \mathbf{x} \sim \text{Dir}(\alpha_1 + x_1, \dots, \alpha_K + x_K).$$

The posterior mean of p_k is:

$$\mathbb{E}[p_k \mid \mathbf{x}] = \frac{\alpha_k + x_k}{\sum_j (\alpha_j + x_j)}.$$

2.4 Non-conjugate priors

Remark 2.10. When the prior is not conjugate, the posterior has no closed form. One must then resort to:

- the Laplace approximation,
- MCMC methods (Chapters 7–8),
- variational inference (Chapter 9).

Definition 2.11 (Laplace approximation)**.** The Laplace approximation approximates the posterior by a Gaussian centered on the MAP (Maximum A Posteriori):

$$\pi(\theta \mid x) \approx \mathcal{N}\left(\hat{\theta}_{\text{MAP}},\ \left[-\nabla^2 \log \pi(\theta \mid x)\big|_{\hat{\theta}_{\text{MAP}}}\right]^{-1}\right).$$

2.5 Hyperpriors and hierarchical priors

Definition 2.12 (Hyperprior)**.** A **hyperprior** is a distribution placed on the hyperparameters of the prior. For instance, if $\theta \sim \text{Be}(a, b)$, one may set $a \sim \text{Ga}(1, 1)$ and $b \sim \text{Ga}(1, 1)$.

Remark 2.13. Hyperpriors lead to hierarchical models, discussed in detail in Chapter 6.

2.6 Prior sensitivity

Definition 2.14 (Sensitivity analysis)**. Sensitivity analysis** evaluates the robustness of Bayesian conclusions by varying the prior within a class $\Gamma = \{\pi_\gamma : \gamma \in \mathcal{G}\}$ and studying the range of posterior quantities.

Proposition 2.15 (Convergence to MLE)**.** For any prior $\pi(\theta)$ that is absolutely continuous with $\pi(\theta_0) > 0$, the posterior mean converges to the MLE as $n \to \infty$. Thus, the influence of the prior vanishes asymptotically.

2.7 Python implementation

Poisson–Gamma conjugacy

```python
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt

# Poisson data
np.random.seed(123)
n = 30
lam_true = 4.5
data = np.random.poisson(lam_true, size=n)

# Gamma(2, 0.5) prior => E[lambda] = 4
alpha_prior, beta_prior = 2, 0.5

# Analytic posterior
alpha_post = alpha_prior + data.sum()
beta_post = beta_prior + n
print(f"Prior: Ga({alpha_prior}, {beta_prior})")
print(f"Posterior: Ga({alpha_post}, {beta_post})")
print(f"Posterior mean: {alpha_post/beta_post:.3f}")
print(f"MLE: {data.mean():.3f}")

# Visualization
lam_grid = np.linspace(0, 10, 500)
fig, ax = plt.subplots(figsize=(8, 5))
ax.plot(lam_grid,
        stats.gamma.pdf(lam_grid, alpha_prior,
                        scale=1/beta_prior),
        label="Prior", linestyle="--")
ax.plot(lam_grid,
        stats.gamma.pdf(lam_grid, alpha_post,
                        scale=1/beta_post),
        label="Posterior")
ax.axvline(lam_true, color="red", alpha=0.5,
           label=f"True value ({lam_true})")
ax.set_xlabel(r"$\lambda$")
ax.set_ylabel("Density")
```

```
ax.legend()
ax.set_title("Poisson-Gamma conjugacy")
plt.tight_layout()
plt.savefig("poisson_gamma.pdf")
```

Multinomial–Dirichlet conjugacy

```
import pymc as pm
import arviz as az

# Categorical data (K=4 categories)
counts = np.array([45, 30, 15, 10])
K = len(counts)
alpha_prior = np.ones(K)  # Dirichlet(1,...,1) = uniform

# Analytic posterior
alpha_post = alpha_prior + counts
p_post_mean = alpha_post / alpha_post.sum()
print("Posterior Dirichlet:", alpha_post)
print("Posterior mean:", p_post_mean)

# PyMC verification
with pm.Model() as model_dir:
    p = pm.Dirichlet("p", a=alpha_prior)
    obs = pm.Multinomial("obs", n=counts.sum(),
                         p=p, observed=counts)
    trace = pm.sample(3000, random_seed=42)

print(az.summary(trace, var_names=["p"]))
```

2.8 Exercises

Exercise 2.1. Show that the exponential family $X \sim \mathrm{Exp}(\lambda)$ admits the conjugate prior $\mathrm{Ga}(\alpha, \beta)$ and determine the posterior.

Exercise 2.2. Let $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ with both μ and σ^2 unknown. Use the NIG prior and compute the posterior. Show that the marginal posterior of μ is a Student-t distribution.

Exercise 2.3. For the geometric model $X \sim \mathrm{Geom}(\theta)$ with $\mathbb{P}(X = k) = \theta(1-\theta)^k$ for $k = 0, 1, 2, \ldots$:

1. Verify that this model belongs to the exponential family.

2. Determine the conjugate prior.

3. Compute the posterior mean.

Exercise 2.4. Perform a sensitivity analysis for the Bernoulli–Beta model with $n = 10$ and $s = 3$. Compare the posteriors obtained with priors $\text{Be}(1, 1)$, $\text{Be}(0.5, 0.5)$, $\text{Be}(2, 8)$, and $\text{Be}(5, 5)$. Plot all four posteriors on the same graph.

Exercise 2.5. (Theoretical) Prove that the conjugate prior for the general exponential family has the form given in the theorem of this section. Verify for the Bernoulli, Poisson, and Normal cases.

Chapter 3

Bayesian Estimation and Loss Functions

In frequentist statistics, an estimator is "good" if it is unbiased and has minimum variance. In Bayesian statistics, the question is framed differently: which summary of the posterior distribution should one choose, and at what cost? The answer depends on what one considers a serious error. Is being far off proportionally worse than being slightly off (quadratic loss), or is any error beyond a threshold equally unacceptable (0-1 loss)? Abraham Wald, in his foundational work on decision theory in the 1940s, showed that this question of the *loss function* is unavoidable: every estimator is, consciously or not, the solution to a decision problem.

This chapter explores the link between estimation and decision, and shows how each loss function leads to a different optimal estimator: the posterior mean, the posterior median, the posterior mode.

Central idea

Bayesian estimation is not limited to a single point estimator: it delivers the entire posterior distribution. Nevertheless, when a point summary is needed, the optimal estimator depends on the chosen **loss function**.

3.1 Bayesian decision theory

Definition 3.1 (Decision problem)**.** A **decision problem** is a triple $(\Theta, \mathcal{A}, L)$ where:

- Θ is the parameter space,
- $\mathcal{A}$ is the action (decision) space,
- $L : \Theta \times \mathcal{A} \to \mathbb{R}^+$ is the loss function.

Definition 3.2 (Posterior risk)**.** The **posterior risk** (or posterior expected loss) of an action $a \in \mathcal{A}$ is:

$$\rho(\pi, a \mid x) = \mathbb{E}_{\theta|x}[L(\theta, a)] = \int_{\Theta} L(\theta, a)\, \pi(\theta \mid x)\, d\theta.$$

Definition 3.3 (Bayes estimator)**.** The **Bayes estimator** under loss L is the action minimizing the posterior risk:

$$\hat{\theta}_B(x) = \arg\min_{a \in \mathcal{A}} \rho(\pi, a \mid x).$$

Definition 3.4 (Integrated Bayes risk)**.** The **Bayes risk** of an estimator $\delta(x)$ is:

$$r(\pi, \delta) = \mathbb{E}_x \left[\rho(\pi, \delta(x) \mid x) \right] = \int \int L(\theta, \delta(x))\, f(x \mid \theta)\, \pi(\theta)\, dx\, d\theta.$$

Theorem 3.5 (Optimality of the Bayes estimator)**.** *The Bayes estimator $\hat{\theta}_B$ minimizes the Bayes risk $r(\pi, \delta)$ among all estimators.*

Proof. The Bayes risk can be written as:

$$r(\pi, \delta) = \int \rho(\pi, \delta(x) \mid x)\, m(x)\, dx,$$

where $m(x)$ is the marginal likelihood. Since $m(x) \geq 0$, minimizing $r(\pi, \delta)$ amounts to minimizing $\rho(\pi, \delta(x) \mid x)$ for each x, yielding $\hat{\theta}_B(x)$. □

3.2 Classical loss functions

Bayes estimators by loss function

Loss function	Expression	Bayes estimator
Quadratic	$L(\theta, a) = (\theta - a)^2$	$\mathbb{E}[\theta \mid x]$ (posterior mean)
Absolute value	$L(\theta, a) = \lvert\theta - a\rvert$	$\mathrm{Med}[\theta \mid x]$ (posterior median)
0–1 (all-or-nothing)	$L(\theta, a) = \mathbf{1}(\lvert\theta - a\rvert > \varepsilon)$	$\mathrm{Mode}[\theta \mid x]$ (MAP)
LINEX	$L(\theta, a) = e^{c(\theta-a)} - c(\theta - a) - 1$	$\frac{1}{c} \log \mathbb{E}[e^{c\theta} \mid x]$
Weighted quadratic	$L(\theta, a) = w(\theta)(\theta - a)^2$	$\frac{\mathbb{E}[w(\theta)\theta \mid x]}{\mathbb{E}[w(\theta) \mid x]}$

3.2.1 Quadratic loss

Theorem 3.6 (Bayes estimator under quadratic loss)**.** *Under quadratic loss $L(\theta, a) = (\theta - a)^2$, the Bayes estimator is the posterior mean:*

$$\hat{\theta}_B(x) = \mathbb{E}[\theta \mid x].$$

Proof.

$$\frac{\partial}{\partial a} \mathbb{E}[(\theta - a)^2 \mid x] = -2\, \mathbb{E}[\theta - a \mid x] = -2\big(\mathbb{E}[\theta \mid x] - a\big).$$

This vanishes at $a = \mathbb{E}[\theta \mid x]$. The second derivative is $2 > 0$, confirming a minimum. □

Corollary 3.7. *The posterior risk under quadratic loss is the posterior variance:*

$$\rho(\pi, \hat{\theta}_B \mid x) = \mathrm{Var}[\theta \mid x].$$

3.2.2 Absolute value loss

Theorem 3.8 (Bayes estimator under absolute loss)**.** *Under $L(\theta, a) = |\theta - a|$, the Bayes estimator is the posterior median.*

Proof. The generalized derivative of the posterior risk is:

$$\frac{\partial}{\partial a}\mathbb{E}[|\theta - a| \mid x] = \mathbb{P}(\theta < a \mid x) - \mathbb{P}(\theta > a \mid x),$$

which vanishes when $\mathbb{P}(\theta < a \mid x) = 1/2$, i.e., a is the median of $\pi(\theta \mid x)$. □

3.2.3 Maximum a posteriori (MAP)

Definition 3.9 (MAP estimator)**.** The **maximum a posteriori** (MAP) estimator is:

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta} \pi(\theta \mid x) = \arg\max_{\theta} \big[\log L(\theta \mid x) + \log \pi(\theta)\big].$$

Remark 3.10. The MAP coincides with the MLE when the prior is uniform. With a Gaussian prior $\mathcal{N}(0, \tau^2)$, the MAP corresponds to Ridge regularization:

$$\hat{\theta}_{\text{MAP}} = \arg\min_{\theta} \left[-\log L(\theta \mid x) + \frac{\|\theta\|^2}{2\tau^2} \right].$$

Proposition 3.11 (MAP and Lasso regularization)**.** If the prior is a Laplace distribution $\pi(\theta) \propto e^{-\lambda|\theta|}$, the MAP corresponds to Lasso (ℓ_1) regularization.

3.3 Credible intervals

Definition 3.12 (Credible interval)**.** A $100(1-\alpha)\%$ **credible interval** is an interval $[a, b]$ such that:

$$\mathbb{P}(\theta \in [a, b] \mid x) = 1 - \alpha.$$

Definition 3.13 (HDI — Highest Density Interval)**.** The **highest density interval** (HDI) is the shortest credible interval:

$$\text{HDI}_{1-\alpha} = \{\theta : \pi(\theta \mid x) \geq c_\alpha\},$$

where c_α is chosen so that $\mathbb{P}(\pi(\theta \mid x) \geq c_\alpha \mid x) = 1 - \alpha$.

Credible vs. confidence

A 95% credible interval means that θ belongs to the interval with probability 0.95 *conditional on the observed data*. A 95% confidence interval means that the procedure contains θ in 95% of *hypothetical* repetitions.

Example 3.14. For the Bernoulli–Beta model with $n = 100$, $s = 30$, $\theta \sim \text{Be}(1, 1)$, the posterior is $\text{Be}(31, 71)$.

The equal-tailed 95% credible interval is:

$$[F^{-1}_{31,71}(0.025),\ F^{-1}_{31,71}(0.975)] \approx [0.216,\ 0.397].$$

3.4 Bayesian vs. frequentist comparison

Definition 3.15 (Frequentist risk)**.** The frequentist risk of an estimator $\delta(x)$ is:

$$R(\theta, \delta) = \mathbb{E}_{x|\theta}[L(\theta, \delta(x))].$$

Definition 3.16 (Admissibility)**.** An estimator δ is **admissible** if there is no estimator δ' such that $R(\theta, \delta') \leq R(\theta, \delta)$ for all θ, with strict inequality for at least one θ.

Theorem 3.17 (Admissibility of Bayes estimators)**.** *Under regularity conditions, every Bayes estimator with finite Bayes risk is admissible.*

Theorem 3.18 (Complete class)**.** *Under general conditions, the class of Bayes estimators (proper and limits) forms a* ***complete class****: every admissible estimator is a Bayes estimator (possibly generalized).*

Example 3.19. James–Stein estimator. For $X \sim \mathcal{N}_p(\theta, I_p)$ with $p \geq 3$, the estimator $\hat{\theta}_{\text{JS}} = (1 - \frac{p-2}{\|X\|^2})X$ dominates the MLE $\hat{\theta} = X$ under quadratic loss. It is a generalized Bayes estimator.

3.5 Point estimation: examples

Example 3.20. Poisson–Gamma model. With $X_i \sim \text{Poi}(\lambda)$ and $\lambda \sim \text{Ga}(\alpha, \beta)$, the posterior is $\text{Ga}(\alpha + \sum x_i, \beta + n)$. The point estimators are:

$$\begin{aligned}
\hat{\lambda}_{\text{mean}} &= \frac{\alpha + \sum x_i}{\beta + n}, \\
\hat{\lambda}_{\text{MAP}} &= \frac{\alpha + \sum x_i - 1}{\beta + n} \quad (\text{if } \alpha + \sum x_i > 1), \\
\hat{\lambda}_{\text{median}} &\approx \hat{\lambda}_{\text{mean}} - \frac{1}{3(\beta + n)} \quad (\text{approx. for large } \alpha').
\end{aligned}$$

Example 3.21. Normal–Normal model. With $X_i \sim \mathcal{N}(\mu, \sigma^2)$, σ^2 known, and $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$: the posterior is Gaussian hence symmetric, and all three estimators coincide:

$$\hat{\mu}_{\text{mean}} = \hat{\mu}_{\text{MAP}} = \hat{\mu}_{\text{median}} = \mu_n = \tau_n^2 \left(\frac{\mu_0}{\tau_0^2} + \frac{n\bar{x}}{\sigma^2} \right).$$

3.6 LINEX loss and asymmetric estimation

Definition 3.22 (LINEX loss)**.** The LINEX (*LINear-EXponential*) loss is:

$$L(\theta, a) = e^{c(\theta - a)} - c(\theta - a) - 1, \qquad c \neq 0.$$

For $c > 0$, overestimation is penalized exponentially. For $c < 0$, underestimation is penalized.

Theorem 3.23 (Bayes estimator under LINEX loss)**.** *The Bayes estimator under LINEX loss is:*

$$\hat{\theta}_{LINEX} = -\frac{1}{c} \log \mathbb{E}[e^{-c\theta} \mid x].$$

Proof. The posterior risk is:

$$\rho(a) = \mathbb{E}[e^{c(\theta-a)} \mid x] - c\,\mathbb{E}[\theta - a \mid x] - 1.$$

Differentiating with respect to a and setting to zero:

$$-c\,e^{-ca}\,\mathbb{E}[e^{c\theta} \mid x] + c = 0 \implies e^{ca} = \mathbb{E}[e^{c\theta} \mid x],$$

whence $a = \frac{1}{c}\log \mathbb{E}[e^{c\theta} \mid x]$. □

3.7 Bayesian shrinkage

Proposition 3.24 (Shrinkage toward the prior mean)**.** In the Normal–Normal model, the posterior mean can be written as:

$$\mathbb{E}[\mu \mid x] = (1 - B)\bar{x} + B\mu_0, \qquad B = \frac{\sigma^2/n}{\tau_0^2 + \sigma^2/n},$$

where $B \in (0, 1)$ is the **shrinkage factor**. The Bayesian estimator is shrunk toward the prior mean μ_0.

3.8 Python implementation

Bayesian estimation and credible intervals

```python
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt

# Bernoulli data
np.random.seed(42)
n, s = 100, 30

# Posterior Beta(31, 71)
a_post, b_post = 1 + s, 1 + n - s
posterior = stats.beta(a_post, b_post)

# Point estimators
mean_post = posterior.mean()
median_post = posterior.median()
mode_post = (a_post - 1) / (a_post + b_post - 2)

print(f"Posterior mean: {mean_post:.4f}")
print(f"Posterior median: {median_post:.4f}")
print(f"MAP: {mode_post:.4f}")
print(f"MLE: {s/n:.4f}")

# Credible intervals
ci_equi = posterior.ppf([0.025, 0.975])
print(f"Equal-tailed 95% CI: [{ci_equi[0]:.4f}, {ci_equi[1]:.4f}]")
```

```python
# HDI (via ArviZ)
import arviz as az
samples = posterior.rvs(100000)
hdi = az.hdi(samples, hdi_prob=0.95)
print(f"95% HDI: [{hdi[0]:.4f}, {hdi[1]:.4f}]")
```

Comparing loss functions

```python
delta = np.linspace(-3, 3, 500)

fig, axes = plt.subplots(1, 3, figsize=(14, 4))

# Quadratic loss
axes[0].plot(delta, delta**2)
axes[0].set_title("Quadratic loss")
axes[0].set_xlabel(r"$\theta - a$")

# Absolute value loss
axes[1].plot(delta, np.abs(delta))
axes[1].set_title("Absolute value loss")
axes[1].set_xlabel(r"$\theta - a$")

# LINEX loss (c=1 and c=-1)
for c in [1, -1]:
    loss = np.exp(c*delta) - c*delta - 1
    axes[2].plot(delta, loss, label=f"c={c}")
axes[2].set_title("LINEX loss")
axes[2].set_xlabel(r"$\theta - a$")
axes[2].legend()

for ax in axes:
    ax.set_ylabel("Loss")

plt.tight_layout()
plt.savefig("loss_functions.pdf")
```

Bayesian shrinkage effect

```python
import pymc as pm
import arviz as az

# Simulation: J groups, simultaneous estimation
np.random.seed(0)
J = 8
true_mu = np.array([-2, -1, 0, 0.5, 1, 1.5, 2, 3])
sigma = 1.0
n_per_group = 5
```

```
data = [np.random.normal(mu, sigma, n_per_group)
        for mu in true_mu]
x_bar = np.array([d.mean() for d in data])

# MLE vs Bayes estimators (prior N(0, 4))
tau0 = 2.0
B = (sigma**2 / n_per_group) / (tau0**2 + sigma**2 / n_per_group)
bayes_est = (1 - B) * x_bar + B * 0  # mu_0 = 0

fig, ax = plt.subplots(figsize=(8, 6))
ax.scatter(range(J), true_mu, marker="*", s=150,
           label=r"True $\mu_j$", zorder=3)
ax.scatter(range(J), x_bar, marker="o",
           label=r"MLE $\bar{x}_j$")
ax.scatter(range(J), bayes_est, marker="s",
           label="Bayes (shrinkage)")
ax.axhline(0, color="gray", linestyle=":", alpha=0.5)
ax.set_xlabel("Group $j$")
ax.set_ylabel(r"$\mu_j$")
ax.legend()
ax.set_title("Bayesian shrinkage")
plt.tight_layout()
plt.savefig("shrinkage.pdf")
```

3.9 Exercises

Exercise 3.1. Prove that the Bayes estimator under the weighted loss $L(\theta, a) = w(\theta)(\theta - a)^2$ is:

$$\hat{\theta}_B = \frac{\mathbb{E}[w(\theta)\theta \mid x]}{\mathbb{E}[w(\theta) \mid x]}.$$

Exercise 3.2. For the Poisson–Gamma model, compute the Bayes estimator under LINEX loss with $c = 1$ and compare with the posterior mean.

Exercise 3.3. Show that in the univariate Gaussian case, the HDI coincides with the equal-tailed interval.

Exercise 3.4. Let $X \sim \mathcal{N}_p(\theta, I_p)$ with $\theta \sim \mathcal{N}_p(0, \tau^2 I_p)$.

1. Compute the Bayes estimator under quadratic loss.
2. Show that it is a shrinkage estimator of the form $c \cdot X$ with $c \in (0, 1)$.
3. Compare with the James–Stein estimator.

Exercise 3.5. (Numerical) Simulate $n = 50$ observations from $\mathcal{N}(3, 4)$. Compare the Bayesian estimators of μ (mean, median, MAP) with the MLE for different priors: $\mathcal{N}(0, 100)$ (vague), $\mathcal{N}(0, 1)$ (informative, centered), $\mathcal{N}(10, 1)$ (informative, off-center).

Chapter 4

Bayesian Hypothesis Testing

The frequentist hypothesis test relies on the p-value: the probability of observing data at least as extreme as those obtained, under the null hypothesis. But this quantity answers a question that nobody really asks: what we want to know is the probability that the hypothesis is true *given the data*, not the other way round. The Bayesian test answers this directly through the *Bayes factor*: the ratio of marginal likelihoods under each hypothesis. Harold Jeffreys, in his 1961 treatise, systematised this approach and proposed the scale that bears his name for interpreting the strength of evidence.

Central idea

Bayesian testing compares hypotheses via their posterior probabilities or the **Bayes factor**, which measures the relative strength of evidence for each hypothesis without depending on the stopping rule or sample space.

4.1 Bayesian formulation of testing

Definition 4.1 (Bayesian test)**.** Consider two hypotheses $H_0 : \theta \in \Theta_0$ and $H_1 : \theta \in \Theta_1$ with $\Theta_0 \cup \Theta_1 = \Theta$ and $\Theta_0 \cap \Theta_1 = \emptyset$. Assign prior probabilities $\mathbb{P}(H_0) = \pi_0$ and $\mathbb{P}(H_1) = \pi_1 = 1 - \pi_0$. The posterior probabilities are:

$$\mathbb{P}(H_k \mid x) = \frac{m_k(x)\,\pi_k}{m(x)}, \quad k = 0, 1,$$

where $m_k(x) = \int_{\Theta_k} f(x \mid \theta)\,\pi(\theta \mid H_k)\,d\theta$.

Definition 4.2 (Bayes factor)**.** The **Bayes factor** in favor of H_0 over H_1 is:

$$B_{01} = \frac{m_0(x)}{m_1(x)} = \frac{\mathbb{P}(H_0 \mid x)/\mathbb{P}(H_1 \mid x)}{\mathbb{P}(H_0)/\mathbb{P}(H_1)}.$$

Posterior odds ratio

$$\underbrace{\frac{\mathbb{P}(H_0 \mid x)}{\mathbb{P}(H_1 \mid x)}}_{\text{posterior odds}} = \underbrace{B_{01}}_{\text{Bayes factor}} \times \underbrace{\frac{\pi_0}{\pi_1}}_{\text{prior odds}}.$$

4.2 Interpreting the Bayes factor

Jeffreys' scale for B_{01}

$\log_{10} B_{01}$	**Evidence in favor of H_0**
0 to 1/2	Weak
1/2 to 1	Substantial
1 to 2	Strong
> 2	Decisive

Remark 4.3. The Bayes factor is symmetric: $B_{10} = 1/B_{01}$. If $B_{01} > 1$, the data support H_0; if $B_{01} < 1$, they support H_1.

4.3 Testing a point hypothesis

Definition 4.4 (Point hypothesis)**.** We test $H_0 : \theta = \theta_0$ against $H_1 : \theta \neq \theta_0$. We assign a point mass $\pi_0 = \mathbb{P}(\theta = \theta_0) > 0$ and a continuous density $\pi_1(\theta)$ on Θ_1.

Theorem 4.5 (Bayes factor for point H_0)**.**

$$B_{01} = \frac{f(x \mid \theta_0)}{\int_{\Theta_1} f(x \mid \theta)\, \pi_1(\theta)\, d\theta} = \frac{L(\theta_0 \mid x)}{m_1(x)}.$$

Jeffreys–Lindley paradox

With a vague prior ($\tau_0^2 \to \infty$) under H_1, the Bayes factor $B_{01} \to \infty$, always favoring H_0, even when the frequentist p-value is very small. This paradox illustrates the importance of the prior choice under H_1.

Example 4.6. Testing $\mu = 0$ in the Gaussian model. Let $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ with σ^2 known. Under $H_0 : \mu = 0$, under $H_1 : \mu \sim \mathcal{N}(0, \tau^2)$.

The Bayes factor is:

$$B_{01} = \sqrt{\frac{\tau^2 + \sigma^2/n}{\sigma^2/n}} \exp\left(-\frac{n\bar{x}^2}{2\sigma^2} \cdot \frac{\tau^2}{\tau^2 + \sigma^2/n} \right).$$

4.4 Bayesian decision rule

Definition 4.7 (Optimal decision rule)**.** Under 0–1 loss:

$$L(H_k, d) = \begin{cases} 0 & \text{if } d = k, \\ 1 & \text{if } d \neq k, \end{cases}$$

the optimal decision is: accept H_0 if $\mathbb{P}(H_0 \mid x) > \mathbb{P}(H_1 \mid x)$, i.e., $B_{01} > \pi_1/\pi_0$.

With $\pi_0 = \pi_1 = 1/2$, this reduces to accepting H_0 if $B_{01} > 1$.

Theorem 4.8 (Decision rule under asymmetric loss)**.** *Under the loss $L(H_0, d = 1) = c_0$ (type I error) and $L(H_1, d = 0) = c_1$ (type II error), the optimal rule is: accept H_0 if*

$$B_{01} > \frac{c_0}{c_1} \cdot \frac{\pi_1}{\pi_0}.$$

4.5 Computing the Bayes factor

4.5.1 Conjugate case

Proposition 4.9 (Bayes factor via conjugacy)**.** For a conjugate model, the marginal likelihood is often available analytically. For the Bernoulli–Beta model with $\theta \sim \mathrm{Be}(a, b)$:

$$m(x) = \frac{B(a + s, b + n - s)}{B(a, b)},$$

where $B(\cdot, \cdot)$ is the beta function.

4.5.2 Savage–Dickey method

Theorem 4.10 (Savage–Dickey density ratio)**.** *To test $H_0 : \theta = \theta_0$ against $H_1 : \theta \sim \pi_1(\theta)$, if π_1 is the prior under H_1 and $\pi_1(\theta \mid x)$ the corresponding posterior:*

$$B_{01} = \frac{\pi_1(\theta_0 \mid x)}{\pi_1(\theta_0)}.$$

Proof. By definition:

$$B_{01} = \frac{f(x \mid \theta_0)}{m_1(x)}.$$

But $\pi_1(\theta_0 \mid x) = f(x \mid \theta_0)\pi_1(\theta_0)/m_1(x)$, so $f(x \mid \theta_0)/m_1(x) = \pi_1(\theta_0 \mid x)/\pi_1(\theta_0)$. □

4.5.3 Numerical approximations

Definition 4.11 (Harmonic mean estimator)**.** The marginal likelihood can be estimated by the harmonic mean of likelihoods evaluated on MCMC samples:

$$\hat{m}(x) = \left(\frac{1}{T} \sum_{t=1}^{T} \frac{1}{L(\theta^{(t)} \mid x)} \right)^{-1}.$$

Instability of the harmonic mean

This estimator has potentially infinite variance and should be used with great caution. More stable methods such as *bridge sampling* or WAIC are preferred.

4.6 Bayesian information criteria

Definition 4.12 (BIC — Bayesian Information Criterion)**.**

$$\mathrm{BIC} = -2 \log L(\hat{\theta}_{\mathrm{MLE}} \mid x) + k \log n,$$

where k is the number of parameters. BIC approximates the log Bayes factor: $-2 \log B_{01} \approx \mathrm{BIC}_0 - \mathrm{BIC}_1$.

Definition 4.13 (WAIC — Widely Applicable IC)**.**

$$\text{WAIC} = -2 \sum_{i=1}^{n} \log \mathbb{E}_{\theta|x}[f(x_i \mid \theta)] + 2\, p_{\text{WAIC}},$$

where $p_{\text{WAIC}} = \sum_{i=1}^{n} \text{Var}_{\theta|x}[\log f(x_i \mid \theta)]$ is the effective number of parameters.

Definition 4.14 (LOO-CV — Leave-One-Out Cross-Validation)**.**

$$\text{LOO} = -2 \sum_{i=1}^{n} \log f(x_i \mid x_{-i}),$$

where $f(x_i \mid x_{-i}) = \int f(x_i \mid \theta)\, \pi(\theta \mid x_{-i})\, d\theta$ is the leave-one-out predictive density.

4.7 ROPE and practical equivalence

Definition 4.15 (ROPE — Region of Practical Equivalence)**.** The **ROPE** is an interval $[\theta_0 - \varepsilon, \theta_0 + \varepsilon]$ defining practical equivalence to θ_0. The decision is:

- Accept H_0 if HDI $\subset$ ROPE.
- Reject H_0 if HDI $\cap$ ROPE $= \emptyset$.
- Undecided otherwise.

4.8 Comparison with p-values

Proposition 4.16 (Bayes–p-value calibration)**.** Sellke, Bayarri, and Berger (2001) show that:

$$B_{01} \geq -\frac{1}{e \cdot p \cdot \log p} \quad \text{for } p < 1/e,$$

where p is the p-value. Thus, a p-value of 0.05 corresponds to $B_{01} \geq 2.5$ at most, representing weak to moderate evidence for H_0.

4.9 Python implementation

Bayes factor for a Gaussian test

```python
import numpy as np
from scipy import stats

# Data
np.random.seed(42)
n = 25
sigma = 1.0
mu_true = 0.3
data = np.random.normal(mu_true, sigma, n)
x_bar = data.mean()
```

```python
# H0: mu = 0, H1: mu ~ N(0, tau^2)
tau = 1.0

# Analytic Bayes factor
var_ratio = tau**2 / (tau**2 + sigma**2/n)
log_BF01 = 0.5 * np.log(1 + n*tau**2/sigma**2) \
           - 0.5 * n * x_bar**2 / sigma**2 * var_ratio
BF01 = np.exp(log_BF01)

print(f"x_bar = {x_bar:.4f}")
print(f"BF01 = {BF01:.4f}")
print(f"log10(BF01) = {np.log10(BF01):.4f}")

if BF01 > 1:
    print("Data favor H0")
else:
    print(f"Data favor H1 (BF10 = {1/BF01:.4f})")

# Comparison with p-value
t_stat = x_bar / (sigma / np.sqrt(n))
p_value = 2 * (1 - stats.norm.cdf(abs(t_stat)))
print(f"Two-sided p-value: {p_value:.4f}")
```

ROPE and HDI with PyMC

```python
import pymc as pm
import arviz as az
import matplotlib.pyplot as plt

# Bayesian model
with pm.Model() as model_test:
    mu = pm.Normal("mu", mu=0, sigma=10)
    y = pm.Normal("y", mu=mu, sigma=sigma,
                  observed=data)
    trace = pm.sample(3000, random_seed=42)

# HDI and ROPE
hdi = az.hdi(trace, var_names=["mu"], hdi_prob=0.95)
print("95% HDI for mu:", hdi)

# ROPE test
rope = [-0.1, 0.1]
az.plot_posterior(trace, var_names=["mu"],
                  rope=rope, ref_val=0,
                  figsize=(8, 4))
plt.savefig("rope_test.pdf")

# Proportion of posterior in ROPE
mu_samples = trace.posterior["mu"].values.flatten()
```

```
in_rope = np.mean((mu_samples >= rope[0]) &
                  (mu_samples <= rope[1]))
print(f"P(mu in ROPE | data) = {in_rope:.4f}")
```

Model comparison with WAIC

```
# Two competing models
with pm.Model() as model_linear:
    a = pm.Normal("a", 0, 10)
    b = pm.Normal("b", 0, 10)
    sigma_m = pm.HalfNormal("sigma", 5)
    x_pred = np.linspace(0, 10, n)
    mu_pred = a + b * x_pred
    y_obs = pm.Normal("y", mu=mu_pred, sigma=sigma_m,
                      observed=data)
    trace_lin = pm.sample(2000, random_seed=42)

with pm.Model() as model_null:
    a = pm.Normal("a", 0, 10)
    sigma_m = pm.HalfNormal("sigma", 5)
    y_obs = pm.Normal("y", mu=a, sigma=sigma_m,
                      observed=data)
    trace_null = pm.sample(2000, random_seed=42)

# WAIC comparison
comp = az.compare({"linear": trace_lin,
                   "null": trace_null})
print(comp)
```

4.10 Exercises

Exercise 4.1. Compute the Bayes factor for the test $H_0 : \theta = 1/2$ against $H_1 : \theta \sim \text{Be}(1, 1)$ in the Bernoulli model with $n = 20$ and $s = 14$.

Exercise 4.2. Prove the Savage–Dickey density ratio. Verify numerically on the Normal–Normal model.

Exercise 4.3. Illustrate the Jeffreys–Lindley paradox: for $n = 100$, $\bar{x} = 0.2$, $\sigma = 1$, compute the p-value and the Bayes factor B_{01} for different values of $\tau \in \{1, 10, 100\}$. Comment.

Exercise 4.4. (Numerical) Compare WAIC and LOO-CV for selecting between a linear and a quadratic model on simulated data.

Exercise 4.5. Derive the BIC approximation to the Bayes factor from the Laplace approximation of the marginal likelihood.

Chapter 5

Bayesian Prediction

In frequentist statistics, prediction uses a point estimate: one estimates θ, then predicts $\tilde{y}$ as if $\hat{\theta}$ were the true value. This approach ignores parameter uncertainty and produces prediction intervals that are too narrow. The Bayesian approach is fundamentally different: instead of fixing θ, one *integrates* over all possible values of θ, weighted by the posterior distribution. The resulting predictive distribution naturally incorporates two sources of uncertainty—the intrinsic randomness of the data and the uncertainty about the model—and produces better-calibrated forecasts.

Central idea

Bayesian prediction integrates parameter uncertainty by averaging the predictive distribution over the posterior. This produces well-calibrated prediction intervals and avoids the overfitting inherent in using point estimates.

5.1 Prior predictive distribution

Definition 5.1 (Prior predictive distribution)**.** The **prior predictive** (or marginal) distribution of a new observation $\tilde{x}$ is:

$$m(\tilde{x}) = \int_{\Theta} f(\tilde{x} \mid \theta)\, \pi(\theta)\, d\theta.$$

It does not depend on the data and represents the prediction before any observation.

Example 5.2. Bernoulli–Beta model. With $\theta \sim \mathrm{Be}(a, b)$ and $\tilde{X} \mid \theta \sim \mathrm{Ber}(\theta)$:

$$m(\tilde{x} = 1) = \mathbb{E}[\theta] = \frac{a}{a+b}.$$

For the Laplace prior $\mathrm{Be}(1, 1)$: $m(\tilde{x} = 1) = 1/2$.

Example 5.3. Normal–Normal model. With $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$ and $\tilde{X} \mid \mu \sim \mathcal{N}(\mu, \sigma^2)$:

$$m(\tilde{x}) = \mathcal{N}(\tilde{x} \mid \mu_0, \sigma^2 + \tau_0^2).$$

The predictive variance $\sigma^2 + \tau_0^2$ combines aleatoric and epistemic uncertainty.

5.2 Posterior predictive distribution

Definition 5.4 (Posterior predictive distribution)**.** The **posterior predictive distribution** of $\tilde{x}$ given $x = (x_1, \dots, x_n)$ is:

$$f(\tilde{x} \mid x) = \int_\Theta f(\tilde{x} \mid \theta)\, \pi(\theta \mid x)\, d\theta.$$

Predictive variance decomposition

$$\operatorname{Var}[\tilde{X} \mid x] = \underbrace{\mathbb{E}_{\theta|x}[\operatorname{Var}[\tilde{X} \mid \theta]]}_{\text{aleatoric uncertainty}} + \underbrace{\operatorname{Var}_{\theta|x}[\mathbb{E}[\tilde{X} \mid \theta]]}_{\text{epistemic uncertainty}}.$$

This decomposition is the **law of total variance**.

Theorem 5.5 (Normal–Normal posterior predictive)**.** *If* $X_i \overset{iid}{\sim} \mathcal{N}(\mu, \sigma^2)$ *with* $\mu \mid x \sim \mathcal{N}(\mu_n, \tau_n^2)$*, then:*

$$\tilde{X} \mid x \sim \mathcal{N}(\mu_n, \sigma^2 + \tau_n^2).$$

Proof. Conditionally on μ, $\tilde{X} \sim \mathcal{N}(\mu, \sigma^2)$. Integrating over $\mu \mid x \sim \mathcal{N}(\mu_n, \tau_n^2)$, the sum of two independent Gaussians gives:

$$\tilde{X} \mid x \sim \mathcal{N}(\mu_n, \sigma^2 + \tau_n^2).$$

□

Theorem 5.6 (Bernoulli–Beta posterior predictive)**.** *If* $X_i \overset{iid}{\sim} \mathit{Ber}(\theta)$ *with* $\theta \mid x \sim \mathit{Be}(a_n, b_n)$*, then:*

$$\mathbb{P}(\tilde{X} = 1 \mid x) = \mathbb{E}[\theta \mid x] = \frac{a_n}{a_n + b_n}.$$

This is ***Laplace's rule of succession*** *when* $a = b = 1$*:* $\mathbb{P}(\tilde{X} = 1 \mid x) = (s+1)/(n+2)$.

Theorem 5.7 (Poisson–Gamma posterior predictive)**.** *If* $X_i \overset{iid}{\sim} \mathit{Poi}(\lambda)$ *with* $\lambda \mid x \sim \mathit{Ga}(\alpha_n, \beta_n)$*, the predictive is a* ***negative binomial****:*

$$\tilde{X} \mid x \sim \mathit{NegBin}\left(\alpha_n, \frac{\beta_n}{\beta_n + 1}\right).$$

Proof.

$$\begin{aligned}
\mathbb{P}(\tilde{X} = k \mid x) &= \int_0^\infty \frac{e^{-\lambda}\lambda^k}{k!} \cdot \frac{\beta_n^{\alpha_n}}{\Gamma(\alpha_n)} \lambda^{\alpha_n - 1} e^{-\beta_n \lambda}\, d\lambda \\
&= \frac{\beta_n^{\alpha_n}}{k!\,\Gamma(\alpha_n)} \cdot \frac{\Gamma(k + \alpha_n)}{(\beta_n + 1)^{k + \alpha_n}} \\
&= \binom{k + \alpha_n - 1}{k} \left(\frac{\beta_n}{\beta_n + 1}\right)^{\alpha_n} \left(\frac{1}{\beta_n + 1}\right)^k.
\end{aligned}$$

□

Theorem 5.8 (Normal–NIG posterior predictive)**.** *If* $(\mu, \sigma^2) \mid x \sim \mathit{NIG}(\mu_n, \kappa_n, \alpha_n, \beta_n)$*, the posterior predictive is a Student-t:*

$$\tilde{X} \mid x \sim t_{2\alpha_n}\left(\mu_n,\ \frac{\beta_n(\kappa_n + 1)}{\alpha_n \kappa_n}\right).$$

5.3 Posterior predictive checking

Definition 5.9 (Posterior predictive check)**.** A **posterior predictive check** (PPC) simulates replicate datasets x^{rep} from the predictive distribution and compares them to the observed data to assess model adequacy.

Definition 5.10 (Bayesian predictive p-value)**.** The **predictive p-value** for a test statistic $T(x)$ is:

$$p_B = \mathbb{P}(T(x^{\text{rep}}) \geq T(x_{\text{obs}}) \mid x_{\text{obs}}).$$

Values near 0 or 1 suggest poor model fit.

Posterior predictive check algorithm

1. For $t = 1, \ldots, T$:
   (a) Draw $\theta^{(t)} \sim \pi(\theta \mid x_{\text{obs}})$ (from MCMC).
   (b) Draw $x^{\text{rep},(t)} \sim f(x \mid \theta^{(t)})$.
   (c) Compute $T^{(t)} = T(x^{\text{rep},(t)})$.
2. Estimate $p_B = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}(T^{(t)} \geq T(x_{\text{obs}}))$.
3. Plot the distribution of $T^{(t)}$ and compare to $T(x_{\text{obs}})$.

5.4 Prediction intervals

Definition 5.11 (Prediction interval)**.** A $100(1-\alpha)\%$ **prediction interval** for $\tilde{X}$ is an interval $[a, b]$ such that:

$$\mathbb{P}(\tilde{X} \in [a, b] \mid x) = 1 - \alpha.$$

Remark 5.12. The prediction interval is always wider than the credible interval for θ, because it integrates both parameter uncertainty and the randomness of the new observation.

5.5 Bayesian vs. plug-in prediction

Plug-in danger

The plug-in approach replaces θ by a point estimate $\hat{\theta}$ and predicts via $f(\tilde{x} \mid \hat{\theta})$. It underestimates predictive uncertainty because it ignores parameter uncertainty. Bayesian prediction is better calibrated.

5.6 Python implementation

Posterior predictive distribution

```
import numpy as np
import pymc as pm
import arviz as az
import matplotlib.pyplot as plt

# Gaussian data
np.random.seed(42)
n = 30
mu_true, sigma_true = 5.0, 2.0
data = np.random.normal(mu_true, sigma_true, n)

# Bayesian model
with pm.Model() as model_pred:
    mu = pm.Normal("mu", mu=0, sigma=10)
    sigma = pm.HalfNormal("sigma", sigma=10)
    y = pm.Normal("y", mu=mu, sigma=sigma, observed=data)
    # Predictive
    y_pred = pm.Normal("y_pred", mu=mu, sigma=sigma)
    trace = pm.sample(3000, random_seed=42)
    ppc = pm.sample_posterior_predictive(trace,
                                         random_seed=42)

# Visualization
fig, axes = plt.subplots(1, 2, figsize=(12, 4))

az.plot_ppc(ppc, observed_adata=trace, ax=axes[0])
axes[0].set_title("Posterior predictive check")

y_pred_samples = ppc.posterior_predictive["y_pred"].values.flatten()
ci_pred = np.percentile(y_pred_samples, [2.5, 97.5])
axes[1].hist(y_pred_samples, bins=50, density=True,
             alpha=0.5, label="Predictive")
axes[1].axvline(ci_pred[0], color="red", linestyle="--",
                label=f"95% PI: [{ci_pred[0]:.1f}, {ci_pred[1]:.1f}]")
axes[1].axvline(ci_pred[1], color="red", linestyle="--")
axes[1].legend()
axes[1].set_title("Posterior predictive distribution")

plt.tight_layout()
plt.savefig("predictive.pdf")
```

Bayesian predictive p-value

```
# Test statistic: standard deviation
T_obs = data.std()
```

```python
# Simulate replicates
T_rep = []
mu_samples = trace.posterior["mu"].values.flatten()
sigma_samples = trace.posterior["sigma"].values.flatten()
for i in range(len(mu_samples)):
    x_rep = np.random.normal(mu_samples[i],
                             sigma_samples[i], n)
    T_rep.append(x_rep.std())
T_rep = np.array(T_rep)

# Predictive p-value
p_B = np.mean(T_rep >= T_obs)
print(f"T(x_obs) = {T_obs:.3f}")
print(f"Predictive p-value = {p_B:.3f}")

plt.figure(figsize=(8, 4))
plt.hist(T_rep, bins=50, density=True, alpha=0.5,
         label="T(x_rep)")
plt.axvline(T_obs, color="red",
            label=f"T(x_obs) = {T_obs:.2f}")
plt.xlabel("Standard deviation")
plt.ylabel("Density")
plt.legend()
plt.title(f"Predictive p-value = {p_B:.3f}")
plt.tight_layout()
plt.savefig("ppc_pvalue.pdf")
```

Poisson–Gamma predictive (negative binomial)

```python
from scipy import stats

# Poisson data
np.random.seed(123)
data_pois = np.random.poisson(3.0, size=20)

# Gamma(2, 1) prior
alpha_n = 2 + data_pois.sum()
beta_n = 1 + len(data_pois)
p_nb = beta_n / (beta_n + 1)

# Predictive: NegBin(alpha_n, p_nb)
x_grid = np.arange(0, 15)
pred_pmf = stats.nbinom.pmf(x_grid, alpha_n, p_nb)

plt.figure(figsize=(8, 4))
plt.bar(x_grid, pred_pmf, alpha=0.7,
        label="NegBin predictive")
plt.xlabel(r"$\tilde{x}$")
plt.ylabel("Probability")
```

```
plt.title("Poisson-Gamma predictive")
plt.legend()
plt.tight_layout()
plt.savefig("pred_negbin.pdf")
```

5.7 Exercises

Exercise 5.1. Derive the posterior predictive distribution for the Bernoulli–Beta model for m new observations (not just one). Show that it is a beta-binomial distribution.

Exercise 5.2. Show that the predictive variance is always greater than or equal to the conditional variance $\mathrm{Var}[\tilde{X} \mid \theta]$ for any value of θ.

Exercise 5.3. Derive the predictive distribution for the Normal–NIG model and show that it is a Student-t.

Exercise 5.4. (Numerical) Simulate Poisson data and compare:

1. the Bayesian predictive (negative binomial),
2. the plug-in predictive (Poisson with $\hat{\lambda} = \bar{x}$).

Compute the actual coverage of 95% prediction intervals for both approaches over 1000 repetitions.

Exercise 5.5. Perform a posterior predictive check for a Poisson model applied to overdispersed data. Use the variance as the test statistic.

Chapter 6

Hierarchical Models

Suppose you are studying school performance across 50 schools. Each school has its own characteristics, but they also share a common environment. Should you model each school independently (many parameters, little data per school) or all together (ignoring differences)? Hierarchical models offer an elegant compromise: the parameters of each school are drawn from a common distribution, whose hyperparameters are themselves estimated. This is Bayesian *shrinkage*: individual estimates are "pulled" toward the group mean, the more so when they are uncertain. This mechanism, formalised by Efron and Morris (1973), is one of the most compelling arguments in favour of the Bayesian approach.

Central idea

Hierarchical (or multilevel) models structure parameters across multiple levels, enabling information sharing between groups. The resulting shrinkage typically improves estimates compared to either separate or fully pooled approaches.

6.1 Motivation and structure

Definition 6.1 (Hierarchical model)**.** A two-level **hierarchical model** has the structure:

$$\begin{aligned} \text{Level 1 (data)}: &\quad X_{ij} \mid \theta_j \sim f(x \mid \theta_j), \\ \text{Level 2 (parameters)}: &\quad \theta_j \mid \phi \sim g(\theta \mid \phi), \\ \text{Level 3 (hyperparameters)}: &\quad \phi \sim h(\phi), \end{aligned}$$

where $j = 1, \ldots, J$ indexes groups and $i = 1, \ldots, n_j$ the observations within group j.

Remark 6.2. Hierarchical models lie between two extremes:

- **Complete pooling**: $\theta_1 = \cdots = \theta_J = \theta$, a single parameter for all.
- **No pooling**: each θ_j is estimated independently.

The hierarchical model achieves optimal *partial pooling*.

6.2 Normal hierarchical model

Theorem 6.3 (Normal hierarchical model)**.** *Consider:*

$$\begin{aligned} X_{ij} \mid \mu_j, \sigma^2 &\sim \mathcal{N}(\mu_j, \sigma^2), \quad i = 1, \dots, n_j, \\ \mu_j \mid \mu, \tau^2 &\sim \mathcal{N}(\mu, \tau^2), \\ (\mu, \tau, \sigma) &\sim \pi(\mu, \tau, \sigma). \end{aligned}$$

The posterior mean of μ_j is:

$$\mathbb{E}[\mu_j \mid x] \approx (1 - B_j)\, \bar{x}_j + B_j\, \hat{\mu},$$

where $B_j = \sigma^2/(n_j\tau^2 + \sigma^2)$ is the shrinkage factor and $\hat{\mu}$ is the estimated overall mean.

Hierarchical shrinkage

$$\hat{\mu}_j^{\text{Bayes}} = \frac{n_j/\sigma^2}{n_j/\sigma^2 + 1/\tau^2}\, \bar{x}_j + \frac{1/\tau^2}{n_j/\sigma^2 + 1/\tau^2}\, \mu.$$

- If $\tau^2 \gg \sigma^2/n_j$: little shrinkage ($\hat{\mu}_j \approx \bar{x}_j$).
- If $\tau^2 \ll \sigma^2/n_j$: strong shrinkage ($\hat{\mu}_j \approx \mu$).

6.3 Foundational example: eight schools

Example 6.4. The classic "eight schools" example (Rubin, 1981) analyzes the effect of a coaching program on SAT scores across $J = 8$ schools. The data are the estimated effects y_j and their standard errors σ_j:

$$y_j \mid \theta_j \sim \mathcal{N}(\theta_j, \sigma_j^2), \qquad \theta_j \mid \mu, \tau \sim \mathcal{N}(\mu, \tau^2).$$

The σ_j are known (estimated with high precision).

6.4 Inference in hierarchical models

6.4.1 Full conditionals

Proposition 6.5 (Full conditionals)**.** In the normal hierarchical model, the full conditionals are:

$$\begin{aligned} \mu_j \mid \text{rest} &\sim \mathcal{N}\left(\frac{n_j \bar{x}_j/\sigma^2 + \mu/\tau^2}{n_j/\sigma^2 + 1/\tau^2},\ \frac{1}{n_j/\sigma^2 + 1/\tau^2} \right), \\ \mu \mid \text{rest} &\sim \mathcal{N}\left(\frac{\sum_j \mu_j/\tau^2}{J/\tau^2 + 1/\sigma_\mu^2},\ \frac{1}{J/\tau^2 + 1/\sigma_\mu^2} \right). \end{aligned}$$

These expressions enable Gibbs sampling (Chapter 8).

6.4.2 Estimating between-group variance

Difficulty of estimating τ

Estimating the between-group variance τ^2 is delicate, especially when J is small. The choice of prior on τ has a significant impact. Recommendations include:

- $\tau \sim \text{Half-Cauchy}(0, A)$ (Gelman, 2006),
- $\tau \sim \text{Half-Normal}(0, A)$,
- $\tau \sim \text{Exp}(\lambda)$.

Avoid the conjugate prior $\tau^2 \sim \text{IG}(\varepsilon, \varepsilon)$ with small ε, which is informative near zero.

6.5 Hierarchical model for binomial data

Example 6.6. Meta-analysis of success rates. Let $X_j \sim \text{Bin}(n_j, \theta_j)$ with $\theta_j \sim \text{Be}(\alpha, \beta)$ and hyperpriors on (α, β):

$$\alpha \sim \text{Exp}(1), \qquad \beta \sim \text{Exp}(1).$$

This model shares information across J studies while allowing heterogeneity.

6.6 Parameterization and MCMC convergence

Definition 6.7 (Centered vs. non-centered parameterization)**.**

- **Centered**: $\theta_j \sim \mathcal{N}(\mu, \tau^2)$.
- **Non-centered**: $\theta_j = \mu + \tau\,\eta_j$, $\eta_j \sim \mathcal{N}(0, 1)$.

The non-centered parameterization often improves MCMC convergence, especially when τ is small or the data are weakly informative.

Remark 6.8. The non-centered parameterization removes the posterior dependence between τ and θ_j, reducing the "funnel" geometry (Neal's funnel) that causes mixing difficulties for MCMC samplers.

6.7 General multilevel models

Definition 6.9 (Bayesian linear mixed model)**.**

$$y_i = X_i\beta + Z_i b_j + \varepsilon_i,$$

where β are fixed effects, $b_j \sim \mathcal{N}(0, \Sigma_b)$ are random effects, and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. The priors are:

$$\beta \sim \mathcal{N}(0, \sigma_\beta^2 I), \quad \Sigma_b \sim \text{InvWishart}(\nu, \Psi), \quad \sigma^2 \sim \text{IG}(a, b).$$

6.8 Python implementation

Eight schools model with PyMC

```
import pymc as pm
import arviz as az
import numpy as np

# Eight schools data
y = np.array([28, 8, -3, 7, -1, 1, 18, 12])
sigma = np.array([15, 10, 16, 11, 9, 11, 10, 18])
J = len(y)

# Non-centered parameterization
with pm.Model() as eight_schools_ncp:
    mu = pm.Normal("mu", 0, sigma=10)
    tau = pm.HalfCauchy("tau", beta=5)
    eta = pm.Normal("eta", 0, 1, shape=J)
    theta = pm.Deterministic("theta", mu + tau * eta)
    obs = pm.Normal("obs", mu=theta, sigma=sigma,
                    observed=y)
    trace = pm.sample(4000, tune=2000,
                      target_accept=0.95,
                      random_seed=42)

# Diagnostics
print(az.summary(trace, var_names=["mu", "tau", "theta"]))

# R-hat and ESS
rhat = az.rhat(trace)
print("Max R-hat:", max(rhat["theta"].values))
```

Visualizing hierarchical shrinkage

```
import matplotlib.pyplot as plt

theta_post = trace.posterior["theta"].mean(
    dim=["chain", "draw"]).values
mu_post = trace.posterior["mu"].mean().item()

fig, ax = plt.subplots(figsize=(8, 5))
schools = [f"School {j+1}" for j in range(J)]
x_pos = np.arange(J)

ax.scatter(x_pos, y, marker="o", s=100,
           label="Observed effect $y_j$", zorder=3)
ax.scatter(x_pos, theta_post, marker="s", s=100,
           label=r"Posterior $\hat{\theta}_j$", zorder=3)
ax.axhline(mu_post, color="gray", linestyle="--",
           label=f"$\\hat{{\\mu}}$ = {mu_post:.1f}")
```

```
# Shrinkage arrows
for j in range(J):
    ax.annotate("", xy=(j, theta_post[j]),
                xytext=(j, y[j]),
                arrowprops=dict(arrowstyle="->",
                                color="red", alpha=0.5))
ax.set_xticks(x_pos)
ax.set_xticklabels(schools, rotation=45, ha="right")
ax.set_ylabel("Coaching effect")
ax.legend()
ax.set_title("Hierarchical shrinkage: Eight Schools")
plt.tight_layout()
plt.savefig("eight_schools_shrinkage.pdf")
```

Centered vs. non-centered comparison

```
# Centered parameterization (for comparison)
with pm.Model() as eight_schools_cp:
    mu = pm.Normal("mu", 0, sigma=10)
    tau = pm.HalfCauchy("tau", beta=5)
    theta = pm.Normal("theta", mu=mu, sigma=tau, shape=J)
    obs = pm.Normal("obs", mu=theta, sigma=sigma,
                    observed=y)
    trace_cp = pm.sample(4000, tune=2000,
                         target_accept=0.95,
                         random_seed=42)

# Compare ESS
ess_ncp = az.ess(trace, var_names=["theta"])
ess_cp = az.ess(trace_cp, var_names=["theta"])
print("ESS (non-centered):", ess_ncp["theta"].values.mean())
print("ESS (centered):", ess_cp["theta"].values.mean())

# Divergences
div_ncp = trace.sample_stats["diverging"].sum().item()
div_cp = trace_cp.sample_stats["diverging"].sum().item()
print(f"Divergences NCP: {div_ncp}, CP: {div_cp}")
```

6.9 Exercises

Exercise 6.1. Derive the full conditionals for the normal hierarchical model and write a Gibbs sampler.

Exercise 6.2. Show that hierarchical shrinkage reduces the total mean squared error $\sum_j \mathbb{E}[(\hat{\mu}_j - \mu_j)^2]$ compared to separate estimation $\hat{\mu}_j = \bar{x}_j$.

Exercise 6.3. Implement a hierarchical beta-binomial model for click-through rate data from $J = 10$ advertisements.

Exercise 6.4. Empirically compare centered and non-centered parameterizations for different values of τ/σ (0.1, 1, 10). Measure the number of divergences and the ESS.

Exercise 6.5. (Theoretical) Show that the normal hierarchical model admits the marginal likelihood:

$$f(y_j \mid \mu, \tau, \sigma_j) = \mathcal{N}(y_j \mid \mu, \sigma_j^2 + \tau^2).$$

Derive an analytic expression for the marginal posterior of (μ, τ) in the eight schools model.

Chapter 7

MCMC — Metropolis-Hastings

In 1953, at Los Alamos National Laboratory, Nicholas Metropolis, Arianna and Marshall Rosenbluth, and Augusta and Edward Teller published a paper on simulating molecules using a particular Monte Carlo method: instead of sampling the Boltzmann distribution directly (impossible in high dimensions), they constructed a Markov chain whose stationary distribution *is* the target distribution. Twenty years later, W. Keith Hastings generalized the algorithm by allowing asymmetric proposal kernels. It would take until the 1990s for Bayesian statisticians — notably Alan Gelfand and Adrian Smith — to realize that this method solved *their* fundamental problem: sampling from complex posterior distributions. The Metropolis–Hastings algorithm thus became the keystone of computational Bayesian inference, making tractable models that had seemed beyond reach.

Central idea

When the posterior has no closed form, we construct a Markov chain whose stationary distribution is the posterior. The Metropolis–Hastings algorithm is the foundational MCMC method: it proposes transitions and accepts or rejects them according to a ratio that guarantees convergence to the target.

7.1 Markov chain background

Definition 7.1 (Markov chain)**.** A sequence $(\theta^{(t)})_{t\geq 0}$ is a **Markov chain** if $\mathbb{P}(\theta^{(t+1)} \in A \mid \theta^{(0)}, \ldots, \theta^{(t)}) = \mathbb{P}(\theta^{(t+1)} \in A \mid \theta^{(t)})$ for all measurable A.

Definition 7.2 (Reversibility and detailed balance)**.** A transition kernel $K(\theta, \theta')$ satisfies **detailed balance** with respect to π if:

$$\pi(\theta)\, K(\theta, \theta') = \pi(\theta')\, K(\theta', \theta) \quad \forall \theta, \theta'.$$

This implies that π is the stationary distribution of the chain.

Theorem 7.3 (Ergodic theorem)**.** *If the chain is irreducible and aperiodic with stationary distribution π, then for every integrable function g:*

$$\frac{1}{T}\sum_{t=1}^{T} g(\theta^{(t)}) \xrightarrow{a.s.} \int g(\theta)\,\pi(\theta)\,d\theta \quad \text{as } T \to \infty.$$

7.2 Metropolis–Hastings algorithm

Metropolis–Hastings algorithm

1. Initialize $\theta^{(0)}$.
2. For $t = 0, 1, 2, \ldots, T-1$:
   (a) Propose $\theta^* \sim q(\theta^* \mid \theta^{(t)})$.
   (b) Compute the acceptance ratio:
   $$\alpha(\theta^{(t)}, \theta^*) = \min\left(1,\ \frac{\pi(\theta^* \mid x)\, q(\theta^{(t)} \mid \theta^*)}{\pi(\theta^{(t)} \mid x)\, q(\theta^* \mid \theta^{(t)})}\right).$$
   (c) Draw $u \sim \mathrm{Unif}(0,1)$.
   (d) If $u \le \alpha$: set $\theta^{(t+1)} = \theta^*$ (accept). Else: set $\theta^{(t+1)} = \theta^{(t)}$ (reject).

Theorem 7.4 (Validity of Metropolis–Hastings)**.** *The Metropolis–Hastings kernel satisfies detailed balance with respect to $\pi(\theta \mid x)$. If the chain is irreducible and aperiodic, it converges to $\pi(\theta \mid x)$.*

Proof. The transition kernel is:

$$K(\theta, \theta') = q(\theta' \mid \theta)\, \alpha(\theta, \theta') \quad (\theta' \neq \theta).$$

To verify detailed balance, assume without loss of generality that $\pi(\theta \mid x)\, q(\theta' \mid \theta) \le \pi(\theta' \mid x)\, q(\theta \mid \theta')$. Then $\alpha(\theta, \theta') = 1$ and:

$$\begin{aligned}
\pi(\theta \mid x)\, K(\theta, \theta') &= \pi(\theta \mid x)\, q(\theta' \mid \theta) \\
&= \pi(\theta' \mid x)\, q(\theta \mid \theta') \cdot \frac{\pi(\theta \mid x)\, q(\theta' \mid \theta)}{\pi(\theta' \mid x)\, q(\theta \mid \theta')} \\
&= \pi(\theta' \mid x)\, q(\theta \mid \theta')\, \alpha(\theta', \theta) \\
&= \pi(\theta' \mid x)\, K(\theta', \theta).
\end{aligned}$$

□

Remark 7.5. Only the ratio $\pi(\theta^*)/\pi(\theta^{(t)})$ is needed, so the normalizing constant $m(x)$ cancels. It suffices to know the posterior up to a constant.

7.3 Special cases

7.3.1 Metropolis algorithm (symmetric)

Definition 7.6 (Metropolis)**.** When the proposal is symmetric, $q(\theta^* \mid \theta) = q(\theta \mid \theta^*)$, the ratio simplifies to:

$$\alpha = \min\left(1,\ \frac{\pi(\theta^* \mid x)}{\pi(\theta^{(t)} \mid x)}\right).$$

Example 7.7. Gaussian random walk. $q(\theta^* \mid \theta^{(t)}) = \mathcal{N}(\theta^{(t)}, \sigma_q^2 I)$. Symmetry is immediate. The step size σ_q must be calibrated: an acceptance rate of $\approx 23\%$ is optimal in high dimensions.

7.3.2 Independence sampler

Definition 7.8 (Independent Metropolis–Hastings)**.** The proposal does not depend on the current state: $q(\theta^* \mid \theta^{(t)}) = q(\theta^*)$. The ratio becomes:

$$\alpha = \min\left(1,\ \frac{\pi(\theta^* \mid x)\, q(\theta^{(t)})}{\pi(\theta^{(t)} \mid x)\, q(\theta^*)}\right).$$

Domination condition

The independence sampler works well only if q dominates $\pi(\cdot \mid x)$ in the sense that the tails of q are at least as heavy as those of π.

7.4 Calibration and adaptation

Optimal acceptance rate

Dimension	Optimal rate
$d = 1$	$\approx 44\%$
$d \geq 2$ (Gaussian target)	$\approx 23.4\%$

Rule of thumb: adjust σ_q to achieve an acceptance rate between 20% and 50%.

Definition 7.9 (Adaptive Metropolis (AM))**.** The AM algorithm adjusts the proposal covariance during burn-in:

$$\Sigma_q^{(t)} = \frac{2.38^2}{d}\,\hat{\Sigma}_t + \varepsilon I_d,$$

where $\hat{\Sigma}_t$ is the empirical covariance of $\theta^{(1)}, \ldots, \theta^{(t)}$.

7.5 Convergence diagnostics

Definition 7.10 (Trace plot)**.** A **trace plot** displays $\theta^{(t)}$ versus t. One looks for "white noise" behavior (good mixing) with no visible trend or correlation.

Definition 7.11 ($\hat{R}$ (Gelman–Rubin))**.** The $\hat{R}$ diagnostic compares within-chain variance W and between-chain variance B for M chains:

$$\hat{R} = \sqrt{\frac{\hat{V}}{W}}, \qquad \hat{V} = \frac{n-1}{n}\,W + \frac{1}{n}\,B.$$

Convergence if $\hat{R} < 1.01$ (strict criterion).

Definition 7.12 (Effective sample size (ESS))**.** The **ESS** measures the number of equivalent independent samples:

$$\mathrm{ESS} = \frac{T}{1 + 2\sum_{k=1}^{\infty} \rho_k},$$

where ρ_k is the lag-k autocorrelation. The recommendation is ESS > 400 per parameter.

MCMC diagnostics checklist

1. Check trace plots (no trends).
2. $\hat{R} < 1.01$ for all parameters.
3. ESS > 400 (or > 100 per chain).
4. No divergences (for HMC/NUTS).
5. Consistency across multiple chains.

7.6 Python implementation

Metropolis–Hastings from scratch

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

# Target: Beta-Binomial posterior
n_data, s = 50, 15  # 15 successes out of 50
a_prior, b_prior = 1, 1

def log_posterior(theta):
    if theta <= 0 or theta >= 1:
        return -np.inf
    return (a_prior + s - 1) * np.log(theta) \
         + (b_prior + n_data - s - 1) * np.log(1 - theta)

# MH random walk
T = 50000
sigma_q = 0.1
samples = np.zeros(T)
samples[0] = 0.5
accepted = 0

for t in range(1, T):
    # Proposal
    theta_star = samples[t-1] + sigma_q * np.random.randn()
    # Ratio (Metropolis since symmetric)
    log_alpha = log_posterior(theta_star) \
              - log_posterior(samples[t-1])
    if np.log(np.random.rand()) < log_alpha:
        samples[t] = theta_star
        accepted += 1
    else:
        samples[t] = samples[t-1]

burn_in = 5000
```

```
samples_post = samples[burn_in:]
print(f"Acceptance rate: {accepted/T:.2%}")
print(f"Posterior mean: {samples_post.mean():.4f}")
print(f"Analytic: {(a_prior+s)/(a_prior+b_prior+n_data):.4f}")
```

MCMC diagnostics

```
fig, axes = plt.subplots(2, 2, figsize=(12, 8))

# Trace plot
axes[0, 0].plot(samples[:2000], alpha=0.7)
axes[0, 0].set_title("Trace plot (first 2000 iterations)")
axes[0, 0].set_xlabel("Iteration")
axes[0, 0].set_ylabel(r"$\theta$")

# Histogram vs analytic
theta_grid = np.linspace(0, 1, 200)
a_post, b_post = a_prior + s, b_prior + n_data - s
axes[0, 1].hist(samples_post, bins=50, density=True,
                alpha=0.5, label="MCMC")
axes[0, 1].plot(theta_grid,
                stats.beta.pdf(theta_grid, a_post, b_post),
                label="Analytic", color="red")
axes[0, 1].legend()
axes[0, 1].set_title("Posterior")

# Autocorrelation
from statsmodels.graphics.tsaplots import plot_acf
plot_acf(samples_post, lags=50, ax=axes[1, 0])
axes[1, 0].set_title("Autocorrelation")

# Running mean
running_mean = np.cumsum(samples_post) / \
               np.arange(1, len(samples_post) + 1)
axes[1, 1].plot(running_mean)
axes[1, 1].axhline((a_prior+s)/(a_prior+b_prior+n_data),
                   color="red", linestyle="--")
axes[1, 1].set_title("Running mean")
axes[1, 1].set_xlabel("Iteration")

plt.tight_layout()
plt.savefig("mcmc_diagnostics.pdf")
```

Multidimensional MH with PyMC

```
import pymc as pm
import arviz as az
```

```python
# Bayesian regression model
np.random.seed(42)
n = 100
x = np.random.randn(n)
y = 2.5 + 1.3 * x + np.random.randn(n) * 0.8

with pm.Model() as model_reg:
    beta0 = pm.Normal("beta0", 0, 10)
    beta1 = pm.Normal("beta1", 0, 10)
    sigma = pm.HalfNormal("sigma", 5)
    mu = beta0 + beta1 * x
    y_obs = pm.Normal("y_obs", mu=mu, sigma=sigma,
                      observed=y)
    # Use Metropolis explicitly
    step = pm.Metropolis()
    trace_mh = pm.sample(20000, step=step,
                         tune=5000, random_seed=42)

# Diagnostics
az.plot_trace(trace_mh, var_names=["beta0", "beta1", "sigma"])
plt.savefig("trace_regression.pdf")

print(az.summary(trace_mh,
                 var_names=["beta0", "beta1", "sigma"]))
```

7.7 Exercises

Exercise 7.1. Implement the Metropolis random walk algorithm to sample from a $t_3(0, 1)$ distribution. Study the effect of the step size σ_q on the acceptance rate and ESS.

Exercise 7.2. Prove that the Metropolis–Hastings algorithm satisfies detailed balance (complete proof).

Exercise 7.3. Implement the Adaptive Metropolis (AM) algorithm and compare with standard Metropolis on a two-dimensional Gaussian posterior with strong correlation ($\rho = 0.95$).

Exercise 7.4. For a two-component Gaussian mixture model, implement MH and observe the multimodality problem. Propose solutions (parallel tempering, large-jump proposals).

Exercise 7.5. (Numerical) Compare the efficiency (ESS per second) of Metropolis–Hastings with NUTS (PyMC default) on a logistic regression model with $p = 10$ predictors.

Chapter 8

Gibbs Sampler

In 1984, Stuart and Donald Geman published a paper on image restoration that introduced a particularly elegant MCMC algorithm: the *Gibbs sampler*. Instead of proposing moves in the full parameter space (as Metropolis–Hastings does), the algorithm updates each component separately, drawing directly from its *full conditional distribution* — the distribution of that component given all the others. The beauty of the procedure is that the acceptance rate is always 100%, eliminating the need to calibrate a proposal distribution. Alan Gelfand and Adrian Smith (1990) then showed that the Gibbs sampler applies to a vast class of hierarchical Bayesian models, triggering the MCMC revolution in statistics.

Central idea

The Gibbs sampler is a special case of Metropolis–Hastings where each component of the parameter is updated by drawing directly from its **full conditional distribution**. The acceptance rate is always 1, eliminating the need to tune the proposal.

8.1 Full conditional distributions

Definition 8.1 (Full conditional)**.** Let $\theta = (\theta_1, \ldots, \theta_d)$. The **full conditional** of θ_k is:

$$\pi(\theta_k \mid \theta_{-k}, x) \propto \pi(\theta_1, \ldots, \theta_d \mid x) \quad \text{as a function of } \theta_k,$$

where $\theta_{-k} = (\theta_1, \ldots, \theta_{k-1}, \theta_{k+1}, \ldots, \theta_d)$.

Remark 8.2. To identify the full conditional, fix all components except θ_k in the joint posterior and recognize the distribution in θ_k.

8.2 Gibbs sampling algorithm

Gibbs sampler

1. Initialize $\theta^{(0)} = (\theta_1^{(0)}, \ldots, \theta_d^{(0)})$.
2. For $t = 0, 1, \ldots, T-1$:

(a) Draw $\theta_1^{(t+1)} \sim \pi(\theta_1 \mid \theta_2^{(t)}, \dots, \theta_d^{(t)}, x)$.
(b) Draw $\theta_2^{(t+1)} \sim \pi(\theta_2 \mid \theta_1^{(t+1)}, \theta_3^{(t)}, \dots, \theta_d^{(t)}, x)$.
$\vdots$
(c) Draw $\theta_d^{(t+1)} \sim \pi(\theta_d \mid \theta_1^{(t+1)}, \dots, \theta_{d-1}^{(t+1)}, x)$.

Theorem 8.3 (Gibbs as a special case of MH)**.** *The Gibbs sampler is a Metropolis–Hastings algorithm where the proposal for θ_k is $q_k(\theta_k^* \mid \theta_{-k}) = \pi(\theta_k^* \mid \theta_{-k}, x)$. The acceptance ratio is always* 1.

Proof. The acceptance ratio for the update of θ_k is:

$$\begin{aligned}\alpha &= \frac{\pi(\theta_k^*, \theta_{-k} \mid x)\, \pi(\theta_k^{(t)} \mid \theta_{-k}, x)}{\pi(\theta_k^{(t)}, \theta_{-k} \mid x)\, \pi(\theta_k^* \mid \theta_{-k}, x)} \\ &= \frac{\pi(\theta_k^* \mid \theta_{-k}, x)\, \pi(\theta_{-k} \mid x) \cdot \pi(\theta_k^{(t)} \mid \theta_{-k}, x)}{\pi(\theta_k^{(t)} \mid \theta_{-k}, x)\, \pi(\theta_{-k} \mid x) \cdot \pi(\theta_k^* \mid \theta_{-k}, x)} = 1.\end{aligned}$$

□

Theorem 8.4 (Convergence of the Gibbs sampler)**.** *Under regularity conditions (positivity of the posterior on the full space), the Gibbs sampler converges to the target distribution $\pi(\theta \mid x)$.*

8.3 Fundamental examples

8.3.1 Bivariate normal model

Example 8.5. Let $(\theta_1, \theta_2) \sim \mathcal{N}_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$. The full conditionals are:

$$\begin{aligned}\theta_1 \mid \theta_2 &\sim \mathcal{N}(\mu_1 + \rho(\theta_2 - \mu_2),\ 1 - \rho^2), \\ \theta_2 \mid \theta_1 &\sim \mathcal{N}(\mu_2 + \rho(\theta_1 - \mu_1),\ 1 - \rho^2).\end{aligned}$$

High correlation

When ρ is close to ± 1, the Gibbs sampler mixes slowly because component-wise updates cannot efficiently explore the elongated distribution. HMC/NUTS or reparameterization is preferred.

8.3.2 Normal model with unknown mean and variance

Example 8.6. Let $X_1, \dots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ with priors $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$ and $\sigma^2 \sim \text{IG}(\alpha_0, \beta_0)$. The conditionals are:

$$\begin{aligned}\mu \mid \sigma^2, x &\sim \mathcal{N}\left(\frac{\mu_0/\tau_0^2 + n\bar{x}/\sigma^2}{1/\tau_0^2 + n/\sigma^2},\ \frac{1}{1/\tau_0^2 + n/\sigma^2}\right), \\ \sigma^2 \mid \mu, x &\sim \text{IG}\left(\alpha_0 + \frac{n}{2},\ \beta_0 + \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2\right).\end{aligned}$$

8.3.3 Mixture model

Example 8.7. Mixture of K Gaussians. Given data $x_1, \dots, x_n$ with latent variables $z_i \in \{1, \dots, K\}$ and parameters $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. The conditionals are:

Allocation: $\mathbb{P}(z_i = k \mid \cdot) \propto \pi_k \mathcal{N}(x_i \mid \mu_k, \sigma_k^2)$.

Proportions: $\boldsymbol{\pi} \mid \cdot \sim \mathrm{Dir}(\alpha_1 + n_1, \dots, \alpha_K + n_K)$, where $n_k = \#\{i : z_i = k\}$.

Means: $\mu_k \mid \cdot \sim \mathcal{N}\Big(\frac{\mu_0/\tau^2 + \sum_{z_i=k} x_i/\sigma_k^2}{1/\tau^2 + n_k/\sigma_k^2},\ \frac{1}{1/\tau^2 + n_k/\sigma_k^2}\Big)$.

Variances: $\sigma_k^2 \mid \cdot \sim \mathrm{IG}\big(a + n_k/2,\ b + \frac{1}{2}\sum_{z_i=k}(x_i - \mu_k)^2\big)$.

8.4 Gibbs variants

Definition 8.8 (Block Gibbs)**.** Instead of updating each component separately, **block Gibbs** updates groups of components simultaneously. This improves mixing when the components within a block are strongly correlated.

Definition 8.9 (Collapsed Gibbs / Rao–Blackwellization)**. Collapsed Gibbs** analytically marginalizes some parameters before sampling, reducing dimensionality and improving efficiency.

Theorem 8.10 (Rao–Blackwellization)**.** *If $g(\theta)$ is a function of interest and $\mathbb{E}[g(\theta_1) \mid \theta_2, x]$ can be computed analytically, then the Rao–Blackwellized estimator:*

$$\hat{g}_{RB} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[g(\theta_1) \mid \theta_2^{(t)}, x]$$

has variance less than or equal to $\hat{g} = \frac{1}{T}\sum_t g(\theta_1^{(t)})$.

8.5 Data augmentation

Definition 8.11 (Data augmentation)**. Data augmentation** introduces latent variables z to facilitate sampling. One alternates:

1. Draw $z \mid \theta, x$ (I-step).
2. Draw $\theta \mid z, x$ (P-step).

Example 8.12. Probit model. For $Y_i \in \{0, 1\}$ with $\mathbb{P}(Y_i = 1) = \Phi(X_i^\top \beta)$, introduce $Z_i \sim \mathcal{N}(X_i^\top \beta, 1)$ and $Y_i = \mathbf{1}(Z_i > 0)$. Conditional on the Z_i, β has a Gaussian posterior. Sampling Z_i is done by truncation.

8.6 Python implementation

Gibbs sampler for the normal model

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
```

```python
# Data
np.random.seed(42)
n = 50
mu_true, sigma_true = 5.0, 2.0
data = np.random.normal(mu_true, sigma_true, n)
xbar = data.mean()
S2 = ((data - data.mean())**2).sum()

# Hyperparameters
mu_0, tau_0 = 0, 10
alpha_0, beta_0 = 1, 1

# Gibbs sampler
T = 10000
mu_samples = np.zeros(T)
sigma2_samples = np.zeros(T)
mu_samples[0] = 0
sigma2_samples[0] = 1

for t in range(1, T):
    # Draw mu | sigma^2, x
    prec_post = 1/tau_0**2 + n/sigma2_samples[t-1]
    mu_post = (mu_0/tau_0**2 + n*xbar/sigma2_samples[t-1]) \
              / prec_post
    mu_samples[t] = np.random.normal(mu_post,
                                     1/np.sqrt(prec_post))

    # Draw sigma^2 | mu, x
    alpha_post = alpha_0 + n/2
    beta_post = beta_0 + 0.5 * np.sum(
        (data - mu_samples[t])**2)
    sigma2_samples[t] = 1 / np.random.gamma(alpha_post,
                                            1/beta_post)

burn_in = 1000
print(f"mu: {mu_samples[burn_in:].mean():.3f} "
      f"(true: {mu_true})")
print(f"sigma: {np.sqrt(sigma2_samples[burn_in:]).mean():.3f} "
      f"(true: {sigma_true})")
```

Gibbs for Gaussian mixture

```python
# Mixture of 2 Gaussians
np.random.seed(42)
n = 200
z_true = np.random.choice([0, 1], size=n, p=[0.4, 0.6])
data_mix = np.where(z_true == 0,
                    np.random.normal(0, 1, n),
                    np.random.normal(5, 1.5, n))
```

```
K = 2
T = 5000
pi_samples = np.zeros((T, K))
mu_mix = np.zeros((T, K))
sigma2_mix = np.zeros((T, K))
z_samples = np.zeros((T, n), dtype=int)

# Initialization
mu_mix[0] = [-1, 6]
sigma2_mix[0] = [1, 1]
pi_samples[0] = [0.5, 0.5]

for t in range(1, T):
    # E-step: draw z_i
    for i in range(n):
        probs = np.array([
            pi_samples[t-1, k] *
            stats.norm.pdf(data_mix[i], mu_mix[t-1, k],
                           np.sqrt(sigma2_mix[t-1, k]))
            for k in range(K)])
        probs /= probs.sum()
        z_samples[t, i] = np.random.choice(K, p=probs)

    # M-step: draw parameters
    for k in range(K):
        idx = z_samples[t] == k
        nk = idx.sum()
        if nk > 0:
            xk = data_mix[idx]
            prec = 1/100 + nk/sigma2_mix[t-1, k]
            m = (nk * xk.mean() / sigma2_mix[t-1, k]) / prec
            mu_mix[t, k] = np.random.normal(m, 1/np.sqrt(prec))
            a = 1 + nk/2
            b = 1 + 0.5 * np.sum((xk - mu_mix[t, k])**2)
            sigma2_mix[t, k] = 1/np.random.gamma(a, 1/b)
        else:
            mu_mix[t, k] = mu_mix[t-1, k]
            sigma2_mix[t, k] = sigma2_mix[t-1, k]

    counts = np.array([(z_samples[t]==k).sum()
                       for k in range(K)])
    pi_samples[t] = np.random.dirichlet(1 + counts)

burn = 1000
print("mu estimates:", mu_mix[burn:].mean(axis=0))
print("pi estimates:", pi_samples[burn:].mean(axis=0))
```

Diagnostics and visualization

```
fig, axes = plt.subplots(2, 2, figsize=(12, 8))

# Trace plots
axes[0, 0].plot(mu_samples[:3000])
axes[0, 0].set_title(r"Trace of $\mu$")
axes[0, 1].plot(sigma2_samples[:3000])
axes[0, 1].set_title(r"Trace of $\sigma^2$")

# Joint posterior
axes[1, 0].scatter(mu_samples[burn_in::10],
                   sigma2_samples[burn_in::10],
                   alpha=0.1, s=1)
axes[1, 0].set_xlabel(r"$\mu$")
axes[1, 0].set_ylabel(r"$\sigma^2$")
axes[1, 0].set_title("Joint posterior")

# Marginal of sigma
axes[1, 1].hist(np.sqrt(sigma2_samples[burn_in:]),
                bins=50, density=True, alpha=0.5)
axes[1, 1].axvline(sigma_true, color="red",
                   linestyle="--")
axes[1, 1].set_title(r"Marginal of $\sigma$")

plt.tight_layout()
plt.savefig("gibbs_diagnostics.pdf")
```

8.7 Exercises

Exercise 8.1. Derive the full conditionals for the Bayesian linear regression model $y = X\beta + \varepsilon$ with $\beta \sim \mathcal{N}(0, \sigma_\beta^2 I)$ and $\sigma^2 \sim \text{IG}(\alpha, \beta)$.

Exercise 8.2. Implement the Albert and Chib (1993) data augmentation for the probit model and compare with PyMC.

Exercise 8.3. For the bivariate Gaussian Gibbs sampler, show analytically that the lag-k autocorrelation is ρ^{2k} and compute the ESS as a function of ρ.

Exercise 8.4. Compare standard Gibbs and block Gibbs on the multivariate normal model with different correlation structures.

Exercise 8.5. (Advanced) Implement collapsed Gibbs for the mixture model by marginalizing the proportions π_k. Compare the ESS with standard Gibbs.

Chapter 9

Variational Inference

MCMC methods are powerful but slow: each sample depends on the previous one, convergence is difficult to diagnose, and scaling remains problematic. In the 2000s, an alternative emerged: *variational inference*, which transforms the sampling problem into an *optimization* problem. The idea, formalized by Michael Jordan, Tommi Jaakkola, and collaborators, is to search within a family of simple distributions $\mathcal{Q}$ for the one that best approximates the posterior, by minimizing the Kullback–Leibler divergence. David Blei and his students democratized the approach in 2017 with ADVI (Automatic Differentiation Variational Inference), making variational inference as accessible as a function call. Today, it powers variational autoencoders (VAEs) and deep generative models.

Central idea

Variational inference transforms the Bayesian integration problem (computing the posterior) into an *optimization* problem. Instead of sampling as in MCMC, we search for the distribution $q^*(\theta)$ in a parametric family that best approximates the true posterior $p(\theta \mid x)$.

9.1 Motivation and problem setup

In many Bayesian models, the posterior

$$p(\theta \mid x) = \frac{p(x \mid \theta)\, \pi(\theta)}{p(x)}$$

is intractable because the normalizing constant $p(x) = \int p(x \mid \theta)\, \pi(\theta)\, d\theta$ has no closed form. MCMC methods provide asymptotically exact solutions but can be prohibitively expensive in high dimensions or with large datasets.

Remark 9.1. Variational inference is often preferred over MCMC when:

- the dataset is very large ($n > 10^5$),
- a fast approximate answer is acceptable,
- the model has many latent variables (e.g., LDA, VAE).

9.2 KL divergence and the ELBO

Definition 9.2 (KL divergence)**.** Let q and p be two densities on Θ. The **Kullback–Leibler divergence** from q to p is:

$$\mathrm{KL}(q\|p) = \int q(\theta) \ln \frac{q(\theta)}{p(\theta)}\, d\theta.$$

It satisfies $\mathrm{KL}(q\|p) \geq 0$ with equality if and only if $q = p$ almost everywhere.

The variational objective is to minimize $\mathrm{KL}(q\|p(\cdot \mid x))$ over a family $\mathcal{Q}$. Decomposing:

$$\ln p(x) = \underbrace{\mathbb{E}_q\big[\ln p(x,\theta) - \ln q(\theta)\big]}_{\mathrm{ELBO}(q)} + \mathrm{KL}\big(q\|p(\cdot \mid x)\big).$$

Definition 9.3 (ELBO — Evidence Lower Bound)**.** The **ELBO** (Evidence Lower Bound) is defined as:

$$\mathcal{L}(q) = \mathbb{E}_q\big[\ln p(x,\theta)\big] - \mathbb{E}_q\big[\ln q(\theta)\big].$$

Since $\mathrm{KL} \geq 0$, we have $\mathcal{L}(q) \leq \ln p(x)$ for all q.

Theorem 9.4 (Equivalence of objectives)**.** *Maximizing the ELBO $\mathcal{L}(q)$ with respect to $q \in \mathcal{Q}$ is equivalent to minimizing* $\mathrm{KL}(q\|p(\cdot \mid x))$.

Attention

KL divergence is not symmetric: $\mathrm{KL}(q\|p) \neq \mathrm{KL}(p\|q)$. Variational inference uses $\mathrm{KL}(q\|p)$ (the "reverse" or "exclusive" direction), which makes q mode-seeking and tend to underestimate the spread of p. Expectation propagation uses $\mathrm{KL}(p\|q)$ (the "forward" or "inclusive" direction), which makes q cover the full support of p.

9.3 Mean-field approximation

Definition 9.5 (Mean-field family)**.** Partition $\theta = (\theta_1, \ldots, \theta_K)$ and define:

$$\mathcal{Q}_{\mathrm{MF}} = \Big\{ q : q(\theta) = \prod_{k=1}^{K} q_k(\theta_k) \Big\}.$$

Each factor q_k is free (nonparametric within the factorization).

Theorem 9.6 (Optimal mean-field update)**.** *Fixing all q_j $(j \neq k)$, the optimal factor q_k^* satisfies:*

$$\ln q_k^*(\theta_k) = \mathbb{E}_{q_{-k}}\big[\ln p(x,\theta)\big] + \text{const},$$

where $q_{-k} = \prod_{j \neq k} q_j(\theta_j)$.

Remark 9.7. For conjugate exponential family models, each q_k^* belongs to the same family as the full conditional prior, making the updates analytic.

9.4 CAVI — Coordinate Ascent Variational Inference

Definition 9.8 (CAVI algorithm)**.** The **CAVI** algorithm cyclically applies the update from Theorem 9.6 for $k = 1, \dots, K$ and monitors convergence via the ELBO.

Proposition 9.9 (Convergence of CAVI)**.** Each CAVI update increases (or maintains) the ELBO. The algorithm converges to a local maximum of $\mathcal{L}(q)$.

Example 9.10. Consider a univariate Gaussian model: $x_i \mid \mu, \tau \sim \mathcal{N}(\mu, \tau^{-1})$, with $\mu \sim \mathcal{N}(0, \sigma_0^2)$ and $\tau \sim \text{Gamma}(a_0, b_0)$. The mean-field approximation $q(\mu, \tau) = q_\mu(\mu)\, q_\tau(\tau)$ gives:

$$q_\mu^*(\mu) = \mathcal{N}\left(\frac{n\bar{x}\, \mathbb{E}[\tau]}{n\mathbb{E}[\tau] + \sigma_0^{-2}},\ \frac{1}{n\mathbb{E}[\tau] + \sigma_0^{-2}} \right), \tag{9.1}$$

$$q_\tau^*(\tau) = \text{Gamma}\left(a_0 + \tfrac{n}{2},\ b_0 + \tfrac{1}{2} \sum_{i=1}^{n} \mathbb{E}_\mu\big[(x_i - \mu)^2\big] \right). \tag{9.2}$$

The cross-expectations are iterated until convergence.

9.5 Stochastic Variational Inference (SVI)

CAVI requires a full pass over all data at each iteration. For large datasets, we use **SVI** (Stochastic Variational Inference).

Definition 9.11 (SVI)**.** Distinguish *global* variables β and *local* variables z_i. At each iteration:

1. Sample a mini-batch $S \subset \{1, \dots, n\}$.
2. Update local variational parameters $q(z_i)$ for $i \in S$.
3. Compute the natural gradient of the ELBO with respect to the global parameters and take a gradient step.

Proposition 9.12 (Natural gradient)**.** For an exponential family model, the natural gradient of the ELBO with respect to the natural parameters λ of $q(\beta)$ is:

$$\hat{\nabla}_\lambda \mathcal{L} = \lambda_{\text{prior}} + n \cdot \mathbb{E}_{q(z_S)}\big[t(\beta, x_S, z_S)\big] - \lambda,$$

where t is the sufficient statistic and S the mini-batch.

```
# SVI pseudo-code
for epoch in range(max_epochs):
    S = random_minibatch(data, batch_size)
    # Local update
    for i in S:
        q_z[i] = update_local(q_beta, x[i])
    # Natural gradient for global parameters
    grad = lambda_prior + (n / len(S)) * sufficient_stats(S) - lam
    lam = lam + rho * grad  # rho = Robbins-Monro step size
```

9.6 Reparameterization trick

Definition 9.13 (Reparameterization trick)**.** If $\theta \sim q_\phi(\theta)$ can be written as $\theta = g(\phi, \epsilon)$ with $\epsilon \sim p(\epsilon)$ independent of ϕ, then:

$$\nabla_\phi \mathbb{E}_{q_\phi}\big[f(\theta)\big] = \mathbb{E}_{p(\epsilon)}\big[\nabla_\phi f(g(\phi, \epsilon))\big],$$

which enables backpropagation through the sampling step.

Example 9.14. For $q_\phi(\theta) = \mathcal{N}(\mu, \sigma^2)$, set $\theta = \mu + \sigma\epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$. This is the foundation of **Variational Autoencoders** (VAEs).

9.7 Comparison with MCMC

Criterion	MCMC	VI
Nature	Sampling	Optimization
Convergence	Exact (asymptotic)	Approximate
Speed	Slow for large n	Fast, scalable
Diagnostics	Traces, $\hat{R}$	ELBO
Multimodality	Possible (HMC, …)	Difficult
Uncertainty	Faithful	Underestimated (forward KL)

Remark 9.15. In practice, it is recommended to use VI for model exploration and MCMC for final inference when accuracy is critical.

9.8 Extensions: normalizing flows

One can enrich the variational family $\mathcal{Q}$ by composing invertible transformations:

$$\theta_K = f_K \circ f_{K-1} \circ \cdots \circ f_1(\theta_0), \quad \theta_0 \sim q_0(\theta_0).$$

The resulting density is obtained by the change-of-variables formula:

$$\ln q_K(\theta_K) = \ln q_0(\theta_0) - \sum_{k=1}^{K} \ln \left| \det \frac{\partial f_k}{\partial \theta_{k-1}} \right|.$$

9.9 Key formulas

Key Formulas

$$\text{ELBO:} \quad \mathcal{L}(q) = \mathbb{E}_q[\ln p(x, \theta)] - \mathbb{E}_q[\ln q(\theta)] \tag{9.3}$$

$$\text{Decomposition:} \quad \ln p(x) = \mathcal{L}(q) + \mathrm{KL}(q \| p(\cdot \mid x)) \tag{9.4}$$

$$\text{Mean field:} \quad \ln q_k^*(\theta_k) = \mathbb{E}_{q_{-k}}[\ln p(x, \theta)] + C \tag{9.5}$$

$$\text{Reparam.:} \quad \nabla_\phi \mathbb{E}_{q_\phi}[f(\theta)] = \mathbb{E}_\epsilon[\nabla_\phi f(g(\phi, \epsilon))] \tag{9.6}$$

9.10 Exercises

Exercise 9.1. Show that $\mathrm{KL}(q\|p) \geq 0$ using Jensen's inequality. Deduce that the ELBO is indeed a lower bound on $\ln p(x)$.

Exercise 9.2. Consider the model: $x_i \mid \mu \sim \mathcal{N}(\mu, 1)$, $i = 1, \ldots, n$, with $\mu \sim \mathcal{N}(0, \sigma_0^2)$. Compute the ELBO for $q(\mu) = \mathcal{N}(m, s^2)$ and maximize it analytically in m and s^2. Verify that you recover the exact posterior.

Exercise 9.3. Implement the CAVI algorithm for the Gaussian model with unknown precision from the example above. Plot the ELBO evolution across iterations.

Exercise 9.4. Experimentally compare (in Python) the mean-field variational approximation and the Gibbs sampler for a two-component Gaussian mixture. Discuss the observed differences in uncertainty estimation.

Exercise 9.5 (ELBO computation for a Gaussian model)**.** Consider the model: $x_1, \ldots, x_n \mid \mu, \sigma^2 \sim \mathcal{N}(\mu, \sigma^2)$ with σ^2 known, and the prior $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$. Let $q(\mu) = \mathcal{N}(m, s^2)$ be the variational family.

1. Write the ELBO $\mathcal{L}(m, s^2) = \mathbb{E}_q[\ln p(x_{1:n}, \mu)] - \mathbb{E}_q[\ln q(\mu)]$ explicitly as a function of m, s^2, μ_0, τ_0^2, σ^2, $\bar{x}$, and n.
2. Maximize $\mathcal{L}$ with respect to m and s^2 and show that the exact posterior $\mathcal{N}(m^*, (s^*)^2)$ is recovered.
3. Evaluate the ELBO at the optimum and verify that $\mathcal{L}(m^*, (s^*)^2) = \ln p(x_{1:n})$ (the KL vanishes).

Exercise 9.6 (Mean-field update for a linear model)**.** Consider the Bayesian linear regression model: $y \mid X, \beta, \tau \sim \mathcal{N}(X\beta, \tau^{-1} I_n)$, $\beta \sim \mathcal{N}(0, \alpha^{-1} I_p)$, $\tau \sim \mathrm{Gamma}(a_0, b_0)$, with α fixed. Set $q(\beta, \tau) = q_\beta(\beta)\, q_\tau(\tau)$.

1. Derive the optimal update $q_\beta^*(\beta)$ by computing $\mathbb{E}_{q_\tau}[\ln p(y, \beta, \tau)]$ and show that $q_\beta^*(\beta) = \mathcal{N}(\mu_\beta, \Sigma_\beta)$ with $\Sigma_\beta = (\mathbb{E}[\tau]\, X^\top X + \alpha I_p)^{-1}$ and $\mu_\beta = \mathbb{E}[\tau]\, \Sigma_\beta\, X^\top y$.
2. Derive $q_\tau^*(\tau) = \mathrm{Gamma}(a_n, b_n)$ by computing $\mathbb{E}_{q_\beta}[\ln p(y, \beta, \tau)]$. Express a_n and b_n.
3. Implement the resulting CAVI algorithm and verify ELBO convergence on simulated data.

Exercise 9.7 (Comparison of VI quality metrics)**.** Let $p(\theta \mid x)$ be a bimodal posterior (mixture of two Gaussians in one dimension) and $q(\theta) = \mathcal{N}(m, s^2)$.

1. Numerically compute $\mathrm{KL}(q\|p)$ (reverse KL) and $\mathrm{KL}(p\|q)$ (forward KL) for various values of m and s. Graphically illustrate the respective minimizers.
2. Show that the minimizer of $\mathrm{KL}(q\|p)$ concentrates on a single mode ("mode-seeking"), while the minimizer of $\mathrm{KL}(p\|q)$ covers both modes ("mass-covering").
3. Also compute the chi-squared divergence $\chi^2(q\|p) = \int \frac{(q-p)^2}{p}\, d\theta$ and the Wasserstein distance $W_2(q, p)$. Compare the variational approximations obtained by minimizing these four metrics.

Chapter 10

Bayesian Nonparametrics

Central idea

In Bayesian nonparametrics, we place priors on infinite-dimensional spaces (distributions, functions, partitions). The model adapts its complexity to the data: the number of components or the flexibility of the fitted function grows with n.

10.1 Motivation

Parametric models fix the number of parameters in advance: a mixture of K Gaussians, a polynomial of degree p, etc. Yet the choice of K or p is often arbitrary.

Remark 10.1. The term "nonparametric" is misleading: there *are* parameters, but infinitely many (or a number that grows with the data).

10.2 The Dirichlet process

Definition 10.2 (Dirichlet process)**.** Let $\alpha > 0$ be a concentration parameter and G_0 a base probability measure on $(\Theta, \mathcal{B})$. We say $G \sim \mathrm{DP}(\alpha, G_0)$ if, for every measurable partition $(A_1, \dots, A_K)$ of Θ:

$$\big(G(A_1), \dots, G(A_K)\big) \sim \mathrm{Dir}\big(\alpha G_0(A_1), \dots, \alpha G_0(A_K)\big).$$

Proposition 10.3 (Properties of the DP)**.** Let $G \sim \mathrm{DP}(\alpha, G_0)$. Then:

1. $\mathbb{E}[G(A)] = G_0(A)$ for all A.
2. $\mathrm{Var}(G(A)) = \dfrac{G_0(A)(1 - G_0(A))}{\alpha + 1}$.
3. G is almost surely discrete (atomic measure).

Attention

The DP generates discrete distributions even when G_0 is continuous. It is this discreteness that induces clustering.

10.3 Stick-breaking construction

Theorem 10.4 (Sethuraman, 1994)**.** *If $G \sim \mathrm{DP}(\alpha, G_0)$, then G admits the representation:*

$$G = \sum_{k=1}^{\infty} \pi_k \, \delta_{\theta_k^*},$$

where $\theta_k^* \overset{iid}{\sim} G_0$ and the weights are constructed by stick-breaking:

$$V_k \overset{iid}{\sim} \mathrm{Beta}(1, \alpha), \qquad \pi_k = V_k \prod_{j=1}^{k-1} (1 - V_j).$$

Remark 10.5. The weights π_k decrease stochastically: the first atoms receive most of the mass. In practice, one truncates to K terms for inference.

```python
import numpy as np

def stick_breaking(alpha, K):
    """Generate K weights via stick-breaking."""
    V = np.random.beta(1, alpha, size=K)
    pi = np.zeros(K)
    remaining = 1.0
    for k in range(K):
        pi[k] = V[k] * remaining
        remaining *= (1 - V[k])
    return pi
```

10.4 The Chinese restaurant process

Definition 10.6 (Chinese restaurant process (CRP))**.** Let $(\theta_1, \theta_2, \ldots)$ be a sample from the DP marginal. The conditional predictive distribution is:

$$\theta_{n+1} \mid \theta_1, \ldots, \theta_n \sim \frac{1}{n + \alpha} \left(\sum_{k=1}^{K_n} n_k \, \delta_{\theta_k^*} + \alpha \, G_0 \right),$$

where $\theta_1^*, \ldots, \theta_{K_n}^*$ are the distinct values and n_k their counts.

Restaurant metaphor

Customers arrive one by one at a restaurant with infinitely many tables. Customer $n+1$ sits at table k (already occupied by n_k customers) with probability $n_k/(n+\alpha)$, or opens a new table with probability $\alpha/(n+\alpha)$.

Proposition 10.7 (Expected number of clusters)**.** For n observations drawn from a $\mathrm{DP}(\alpha, G_0)$, the expected number of distinct clusters is:

$$\mathbb{E}[K_n] = \alpha \sum_{i=1}^{n} \frac{1}{\alpha + i - 1} \approx \alpha \ln\!\left(1 + \frac{n}{\alpha}\right).$$

10.5 Dirichlet process Gaussian mixture model (DP-GMM)

Definition 10.8 (DP-GMM)**.** The **infinite Gaussian mixture model** (DP-GMM) is defined by:

$$G \sim \mathrm{DP}(\alpha, G_0), \tag{10.1}$$

$$(\mu_i, \Sigma_i) \sim G, \quad i = 1, \ldots, n, \tag{10.2}$$

$$x_i \mid \mu_i, \Sigma_i \sim \mathcal{N}(\mu_i, \Sigma_i). \tag{10.3}$$

The number of components is determined by the data.

Example 10.9. With $G_0 = \mathrm{NIW}(\mu_0, \kappa_0, \nu_0, \Psi_0)$ (Normal-Inverse-Wishart), the CRP Gibbs sampler iterates:

1. For each i, reassign z_i with $p(z_i = k \mid \ldots) \propto n_{-i,k}\, \mathcal{N}(x_i \mid \mu_k, \Sigma_k)$ or create a new cluster.
2. For each cluster k, update (μ_k, Σ_k) via the NIW posterior.

```python
from sklearn.mixture import BayesianGaussianMixture

# DP-GMM via scikit-learn (variational approximation)
dpgmm = BayesianGaussianMixture(
    n_components=20,             # upper bound
    covariance_type='full',
    weight_concentration_prior_type='dirichlet_process',
    weight_concentration_prior=0.1
)
dpgmm.fit(X)
labels = dpgmm.predict(X)
```

10.6 Gaussian processes as function-space priors

Definition 10.10 (Gaussian process)**.** A **Gaussian process** (GP) is a collection of random variables $\{f(x) : x \in \mathcal{X}\}$ such that every finite subcollection is jointly Gaussian:

$$f \sim \mathcal{GP}(m, k), \quad (f(x_1), \ldots, f(x_n)) \sim \mathcal{N}\big(\mathbf{m}, \mathbf{K}\big),$$

where $m_i = m(x_i)$ and $K_{ij} = k(x_i, x_j)$.

Theorem 10.11 (GP regression posterior)**.** *Let* $\mathbf{y} = f(\mathbf{X}) + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$. *The posterior of* $f_* = f(x_*)$ *is:*

$$f_* \mid \mathbf{X}, \mathbf{y}, x_* \sim \mathcal{N}(\bar{f}_*, \mathrm{Var}(f_*)), \tag{10.4}$$

$$\bar{f}_* = \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}, \tag{10.5}$$

$$\mathrm{Var}(f_*) = k(x_*, x_*) - \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_*. \tag{10.6}$$

Remark 10.12. The choice of kernel k encodes assumptions about the smoothness of f: RBF for smooth functions, Matérn for fine control over differentiability, periodic for cyclical data.

10.7 Other nonparametric processes

Definition 10.13 (Indian buffet process (IBP))**.** The **IBP** is a prior on binary matrices Z of size $n \times \infty$ used for feature selection. Each customer (observation) selects existing dishes (features) with probability m_k/n and tries Poisson(α/n) new dishes.

Definition 10.14 (Pitman–Yor process)**.** The **Pitman–Yor process** $\mathrm{PY}(d, \alpha, G_0)$ generalizes the DP with a discount parameter $d \in [0, 1)$. The predictive distribution is:

$$\theta_{n+1} \mid \theta_{1:n} \sim \frac{1}{n+\alpha}\left(\sum_{k=1}^{K_n}(n_k - d)\,\delta_{\theta_k^*} + (\alpha + dK_n)\,G_0\right).$$

For $d = 0$, one recovers the DP.

10.8 Key formulas

Key Formulas

$$\text{DP:} \quad (G(A_1), \ldots, G(A_K)) \sim \mathrm{Dir}(\alpha G_0(A_1), \ldots, \alpha G_0(A_K)) \tag{10.7}$$

$$\text{CRP:} \quad p(\theta_{n+1} = \theta_k^*) = \frac{n_k}{n+\alpha}, \quad p(\text{new}) = \frac{\alpha}{n+\alpha} \tag{10.8}$$

$$\text{Stick-breaking:} \quad \pi_k = V_k \prod_{j<k}(1 - V_j), \quad V_k \sim \mathrm{Beta}(1, \alpha) \tag{10.9}$$

$$\text{GP:} \quad \bar{f}_* = \mathbf{k}_*^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y} \tag{10.10}$$

10.9 Exercises

Exercise 10.1. Verify that the stick-breaking weights sum to 1 almost surely. *Hint:* compute $\mathbb{E}\left[\sum_{k=1}^{K} \pi_k\right]$ and take the limit.

Exercise 10.2. Simulate the Chinese restaurant process for $\alpha \in \{0.1, 1, 10, 100\}$ and $n = 500$. Plot the number of clusters K_n as a function of α. Compare with the theoretical formula $\mathbb{E}[K_n] \approx \alpha \ln(1 + n/\alpha)$.

Exercise 10.3. Implement a Gibbs sampler for the DP-GMM on simulated 2D data (three Gaussian clusters). Visualize the cluster assignments across iterations.

Exercise 10.4. Consider a GP with RBF kernel $k(x, x') = \sigma_f^2 \exp(-\|x - x'\|^2/(2\ell^2))$. Show that the GP predictive distribution converges to an exact interpolator as $\sigma^2 \to 0$. Illustrate graphically with simulated data.

Exercise 10.5. Show that the Pitman–Yor process generates $K_n = O(n^d)$ clusters, compared to $K_n = O(\ln n)$ for the DP. What are the implications for modeling textual data following Zipf's law?

Chapter 11

Applications

Central idea

Bayesian statistics is ubiquitous in modern applications: clinical trials, A/B testing, automatic hyperparameter optimization, Bayesian neural networks, and many scientific domains. This chapter illustrates the power of the Bayesian paradigm on concrete problems.

11.1 Bayesian clinical trials

Definition 11.1 (Adaptive Bayesian clinical trial)**.** An **adaptive Bayesian clinical trial** is a trial in which the design (sample size, patient allocation, stopping criteria) is modified during the study based on the posterior distribution of the efficacy parameters.

Example 11.2. Let θ_T be the response probability under treatment and θ_C under control. We model:

$$\theta_T \sim \mathrm{Beta}(1,1), \quad \theta_C \sim \mathrm{Beta}(1,1).$$

After observing s_T successes out of n_T treated patients and s_C out of n_C controls:

$$\theta_T \mid \text{data} \sim \mathrm{Beta}(1+s_T,\, 1+n_T-s_T).$$

We stop for efficacy if $\mathbb{P}(\theta_T > \theta_C \mid \text{data}) > 0.99$, or for futility if $\mathbb{P}(\theta_T > \theta_C + \delta \mid \text{data}) < 0.05$.

Remark 11.3. Adaptive Bayesian trials have been approved by the FDA since the 2010s. They reduce the required sample size by 20–40% on average compared to fixed designs.

```python
import numpy as np
from scipy.stats import beta

def prob_superiority(s_T, n_T, s_C, n_C, n_samples=100000):
    """P(theta_T > theta_C | data) via Monte Carlo."""
    theta_T = beta.rvs(1 + s_T, 1 + n_T - s_T, size=n_samples)
    theta_C = beta.rvs(1 + s_C, 1 + n_C - s_C, size=n_samples)
    return np.mean(theta_T > theta_C)
```

11.2 Bayesian A/B testing

Definition 11.4 (Bayesian A/B test)**.** A **Bayesian A/B test** compares two variants (A and B) of a product by maintaining a posterior distribution on each variant's conversion rate. A decision is made when the probability that one variant is better exceeds a fixed threshold.

Proposition 11.5 (Expected loss)**.** The **expected loss** from choosing B is:

$$\mathcal{L}_B = \mathbb{E}\big[\max(\theta_A - \theta_B, 0) \mid \text{data}\big].$$

We choose B if $\mathcal{L}_B < \epsilon$ (acceptable loss threshold). This criterion directly controls the cost of a wrong decision.

Attention

Unlike the frequentist test (p-value), the Bayesian A/B test allows "peeking" (examining data during collection) without inflating the error rate, because the posterior is coherent at any time.

11.3 Bayesian optimization

Definition 11.6 (Bayesian optimization)**. Bayesian optimization** seeks the global minimum of an expensive function $f : \mathcal{X} \to \mathbb{R}$ by building a probabilistic surrogate model (typically a GP) and selecting evaluation points via an **acquisition function**.

Theorem 11.7 (GP posterior as surrogate)**.** *Let $f \sim \mathcal{GP}(0, k)$ and $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$ with $y_i = f(x_i) + \epsilon_i$. The prediction at x is:*

$$\mu_n(x) = \mathbf{k}(x)^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}, \tag{11.1}$$

$$\sigma_n^2(x) = k(x, x) - \mathbf{k}(x)^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}(x). \tag{11.2}$$

Definition 11.8 (Acquisition functions)**.** Common acquisition functions include:

- **Expected Improvement (EI)**: $\text{EI}(x) = \mathbb{E}\big[\max(f_{\min} - f(x), 0)\big]$.
- **Upper Confidence Bound (UCB)**: $\text{UCB}(x) = \mu_n(x) - \kappa\, \sigma_n(x)$ (for minimization).
- **Probability of Improvement (PI)**: $\text{PI}(x) = \Phi\left(\frac{f_{\min} - \mu_n(x)}{\sigma_n(x)}\right)$.

Example 11.9. Hyperparameter optimization for a neural network:

```
from skopt import gp_minimize

def objective(params):
    lr, dropout = params
    model = build_model(lr=lr, dropout=dropout)
    return -model.fit_and_evaluate(X_train, y_train, X_val, y_val)

result = gp_minimize(
```

```
    objective,
    dimensions=[(1e-5, 1e-1, "log-uniform"),  # learning rate
                (0.0, 0.5)],                    # dropout
    n_calls=50,
    random_state=42
)
```

11.4 Bayesian neural networks

Definition 11.10 (Bayesian neural network (BNN))**.** A **BNN** places a prior $p(\mathbf{w})$ on the network weights $\mathbf{w}$ and computes the posterior $p(\mathbf{w} \mid \mathcal{D})$. The prediction is:

$$p(y_* \mid x_*, \mathcal{D}) = \int p(y_* \mid x_*, \mathbf{w})\, p(\mathbf{w} \mid \mathcal{D})\, d\mathbf{w}.$$

Remark 11.11. The above integral is intractable for deep networks. Three practical approaches:

1. **MC Dropout**: use dropout at inference time as a variational approximation (Gal and Ghahramani, 2016).
2. **Bayes by Backprop**: variational inference with reparameterization on each weight.
3. **Deep ensembles**: train M networks and average (frequentist approximation of uncertainty).

Proposition 11.12 (Epistemic vs. aleatoric uncertainty)**.** The predictive variance of a BNN decomposes as:

$$\underbrace{\mathrm{Var}_{p(\mathbf{w}|\mathcal{D})}\big[\mathbb{E}[y_* \mid x_*, \mathbf{w}]\big]}_{\text{epistemic}} + \underbrace{\mathbb{E}_{p(\mathbf{w}|\mathcal{D})}\big[\mathrm{Var}[y_* \mid x_*, \mathbf{w}]\big]}_{\text{aleatoric}}.$$

Epistemic uncertainty decreases with more data; aleatoric does not.

11.5 Applications in astronomy

Example 11.13. In cosmology, estimating the parameters of the ΛCDM model (Hubble constant H_0, matter density Ω_m, etc.) relies on Bayesian inference applied to cosmic microwave background (CMB) data. The software `CosmoMC` uses MCMC to explore the posterior in ~ 30 dimensions.

Example 11.14. Exoplanet detection via the radial velocity method models the signal as:

$$v(t) = \sum_{j=1}^{K} K_j \sin(2\pi t / P_j + \phi_j) + \epsilon(t),$$

where the number of planets K is unknown. Bayesian model comparison (Bayes factors) determines K.

11.6 Applications in ecology

Example 11.15. Bayesian **capture-recapture** models estimate population size N. With a prior $N \sim \text{Poisson}(\lambda)$ and recapture data, the posterior on N naturally quantifies uncertainty—essential for conservation management.

Definition 11.16 (Hierarchical model in ecology)**. Bayesian hierarchical models** are standard in ecology for modeling:

- between-site variability (spatial random effects),
- imperfect detection (occupancy models),
- temporal dynamics (state-space models).

The parameters for each site j share a common hyperprior: $\theta_j \sim p(\theta_j \mid \psi)$, $\psi \sim p(\psi)$.

11.7 Key formulas

Key Formulas

$$\text{EI:} \quad \text{EI}(x) = (f_{\min} - \mu_n(x))\,\Phi(z) + \sigma_n(x)\,\phi(z), \quad z = \frac{f_{\min} - \mu_n(x)}{\sigma_n(x)} \tag{11.3}$$

$$\text{BNN prediction:} \quad p(y_* \mid x_*) \approx \frac{1}{M}\sum_{m=1}^{M} p(y_* \mid x_*, \mathbf{w}^{(m)}), \quad \mathbf{w}^{(m)} \sim p(\mathbf{w} \mid \mathcal{D}) \tag{11.4}$$

$$\text{A/B loss:} \quad \mathcal{L}_B = \mathbb{E}[\max(\theta_A - \theta_B, 0) \mid \text{data}] \tag{11.5}$$

11.8 Exercises

Exercise 11.1. Implement a Bayesian A/B test with Beta(1,1) priors to compare click-through rates of two web pages. Simulate data and plot the evolution of $\mathbb{P}(\theta_A > \theta_B \mid \text{data})$ across observations.

Exercise 11.2. Perform Bayesian optimization to find the minimum of the Branin function $f(x_1, x_2)$ using a GP with Matérn kernel. Compare with random search in terms of the number of evaluations needed.

Exercise 11.3. Implement MC Dropout on a two-hidden-layer neural network for a regression problem. Plot the mean prediction and the 95% confidence interval. Observe how uncertainty behaves far from training data.

Exercise 11.4. Design an adaptive Bayesian clinical trial with the following rules: stop for efficacy if $\mathbb{P}(\theta_T - \theta_C > 0.05 \mid \text{data}) > 0.95$, stop for futility if $\mathbb{P}(\theta_T > \theta_C \mid \text{data}) < 0.2$. Simulate 1000 trials under H_0 and H_1 and estimate the operating characteristics (power, type I error rate, average sample size).

Exercise 11.5. In an exoplanet detection problem, compare models M_0 (pure noise) and M_1 (one planet). Compute the Bayes factor BF_{10} by Monte Carlo integration on simulated data. Discuss Jeffreys' scale for interpretation.

Bibliography

[1] Gelman, A. et al., *Bayesian Data Analysis*, 3rd ed., CRC Press, 2013.

[2] Robert, C.P., *The Bayesian Choice*, 2nd ed., Springer, 2007.

[3] Bernardo, J.M. and Smith, A.F.M., *Bayesian Theory*, Wiley, 2000.

[4] McElreath, R., *Statistical Rethinking*, 2nd ed., CRC Press, 2020.