Statistique Bayésienne

Notes de Cours

Master M2 — 2025–2026

Yaë Ulrich Gaba

« La probabilité est le guide de la vie. »
— Marcus Tullius Cicero

25 mars 2026

Table des matières

Préface

Objectifs de ce cours

La statistique bayésienne constitue l'un des deux grands paradigmes de l'inférence statistique, aux côtés de l'approche fréquentiste. Fondée sur le théorème de Bayes, elle offre un cadre cohérent pour quantifier l'incertitude, incorporer des connaissances a priori et mettre à jour les croyances à la lumière des données.

Ce cours s'adresse aux étudiants de niveau Master en mathématiques appliquées, en science des données et en intelligence artificielle. Il suppose une maîtrise préalable de la théorie des probabilités, de la statistique mathématique élémentaire et de l'algèbre linéaire.

Organisation du cours

Le cours est structuré en onze chapitres, regroupés en quatre parties thématiques :

1. **Fondements** (Chapitres 1–5) : paradigme bayésien, familles conjuguées, estimation, tests et prédiction.
2. **Modèles avancés** (Chapitre 6) : modèles hiérarchiques.
3. **Méthodes de calcul** (Chapitres 7–9) : MCMC (Metropolis–Hastings, échantillonneur de Gibbs) et inférence variationnelle.
4. **Extensions et applications** (Chapitres 10–11) : statistique bayésienne non paramétrique et applications concrètes.

Philosophie pédagogique

Chaque chapitre alterne entre :

- des **développements théoriques** rigoureux (définitions, théorèmes, preuves),
- des **exemples détaillés** avec calculs explicites,
- des **implémentations numériques** en Python (PyMC, NumPy, SciPy),
- des **exercices** de difficulté croissante.

Approche bayésienne *versus* fréquentiste

Avant d'entamer l'étude détaillée, situons les deux paradigmes :

Aspect	**Fréquentiste**	**Bayésien**
Paramètre θ	Fixe, inconnu	Variable aléatoire
Probabilité	Fréquence limite	Degré de croyance
Incertitude	Intervalles de confiance	Intervalles de crédibilité
Information a priori	Non utilisée formellement	Intégrée via le prior
Résultat principal	Estimateur $\hat{\theta}(x)$	Loi a posteriori $\pi(\theta \mid x)$
Cohérence	Peut violer le principe de vraisemblance	Cohérence assurée
Petit échantillon	Difficultés possibles	Naturellement adaptée
Calcul	Souvent analytique	Souvent numérique (MCMC)

Le théorème fondateur

Tout le paradigme bayésien repose sur la formule :

Théorème de Bayes

Soit $\theta \in \Theta$ un paramètre de loi a priori $\pi(\theta)$ et $x = (x_1, \ldots, x_n)$ un échantillon de vraisemblance $L(\theta \mid x)$. La **loi a posteriori** est :

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

Notation et conventions

Tout au long de ce cours, nous adoptons les conventions suivantes :

- X désigne une variable aléatoire, x une réalisation.
- θ est le paramètre d'intérêt, Θ l'espace des paramètres.
- $\pi(\theta)$ est la loi a priori (*prior*).
- $\pi(\theta \mid x)$ est la loi a posteriori (*posterior*).
- $L(\theta \mid x) = f(x \mid \theta)$ est la vraisemblance.
- $m(x) = \int L(\theta \mid x)\, \pi(\theta)\, d\theta$ est la vraisemblance marginale (*evidence*).
- $\mathbb{E}[\cdot]$, $\mathrm{Var}[\cdot]$, $\mathrm{Cov}[\cdot]$ désignent espérance, variance et covariance.
- $\mathbb{P}(\cdot)$ désigne la probabilité.

- $\mathrm{KL}(p\|q)$ désigne la divergence de Kullback–Leibler.
- $\|\cdot\|$ désigne la norme euclidienne.

Distributions utilisées fréquemment

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

Jalons historiques

- **1763** : Publication posthume du théorème de Thomas Bayes par Richard Price.
- **1812** : Pierre-Simon de Laplace formalise et généralise la méthode dans *Théorie analytique des probabilités*.
- **1937** : Bruno de Finetti développe la notion de probabilité subjective et le théorème d'échangeabilité.
- **1954** : Leonard Jimmie Savage publie *The Foundations of Statistics*, pilier de la théorie de la décision bayésienne.
- **1970s** : Dennis Lindley et d'autres établissent les fondements modernes.
- **1990s** : Révolution MCMC — Metropolis–Hastings devient le cheval de bataille du calcul bayésien.
- **2000s–2020s** : Inférence variationnelle, processus gaussiens, apprentissage profond bayésien.

Logiciels et bibliothèques

Les implémentations pratiques de ce cours utilisent :

- **Python 3.10+** comme langage principal.
- **PyMC** (v5+) pour la modélisation bayésienne et MCMC.

- **ArviZ** pour le diagnostic et la visualisation des chaînes.
- **NumPy / SciPy** pour le calcul numérique.
- **Matplotlib / Seaborn** pour les graphiques.

Installation des dépendances

```
# Installation recommandée
# 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__}")
```

Comment utiliser ce cours

Prérequis importants

Ce cours suppose que l'étudiant maîtrise :

1. La théorie des probabilités (variables aléatoires, lois classiques, convergences).
2. La statistique mathématique (estimation, tests, vraisemblance).
3. L'algèbre linéaire (matrices, décompositions, formes quadratiques).
4. La programmation en Python (syntaxe de base, NumPy).

Fil conducteur

L'idée directrice de ce cours est simple : *toute inférence est un problème de mise à jour d'une distribution de probabilité.* On part d'une croyance a priori $\pi(\theta)$, on observe des données x, et on obtient une croyance a posteriori $\pi(\theta \mid x)$. Cette vision unifiée sous-tend l'ensemble des méthodes présentées.

Références principales

- A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, D.B. Rubin, *Bayesian Data Analysis* (3e éd.), 2013.
- C.P. Robert, *The Bayesian Choice* (2e éd.), 2007.
- J.K. Kruschke, *Doing Bayesian Data Analysis* (2e éd.), 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.

L'auteur
Mars 2026

Chapitre 1

Paradigme Bayésien

En 1763, deux ans après sa mort, le révérend Thomas Bayes voit son essai publié à titre posthume par la Royal Society de Londres. L'article propose une méthode pour inverser les probabilités : connaissant les données observées, que peut-on dire de la cause qui les a produites ? Cette question, d'apparence innocente, va engendrer l'un des débats les plus durables de toute la statistique. D'un côté, les *fréquentistes*, pour qui les paramètres sont des constantes fixes et inconnues. De l'autre, les *bayésiens*, pour qui les paramètres sont des variables aléatoires dotées d'une distribution reflétant notre état de connaissance. Ce chapitre présente le paradigme bayésien : sa logique, ses forces, et les raisons pour lesquelles il connaît aujourd'hui un renouveau spectaculaire.

Idée centrale

Le paradigme bayésien traite le paramètre θ comme une variable aléatoire et encode toute l'incertitude dans des distributions de probabilité. L'inférence consiste à passer de la loi a priori $\pi(\theta)$ à la loi a posteriori $\pi(\theta \mid x)$ via le théorème de Bayes.

1.1 Introduction et motivation

L'approche fréquentiste considère le paramètre θ comme une constante fixe inconnue et évalue les procédures d'inférence par leurs propriétés répétées sur des échantillons hypothétiques. L'approche bayésienne, en revanche, modélise θ comme une variable aléatoire dont la distribution reflète notre état de connaissance.

Cette différence de philosophie entraîne des conséquences profondes :

- L'incertitude est directement quantifiée par la loi a posteriori.
- L'information a priori est formellement intégrée.
- Toute décision optimale peut être déduite du posterior.

1.2 Fondements axiomatiques

1.2.1 Probabilité subjective

Définition 1.1 (Probabilité subjective)**.** Une **probabilité subjective** est une fonction $\mathbb{P} : \mathcal{F} \to [0, 1]$ satisfaisant les axiomes de Kolmogorov, interprétée comme le degré de

croyance d'un agent rationnel dans la réalisation d'un événement.

Les axiomes de cohérence de de Finetti montrent qu'un agent dont les paris sont cohérents (pas de *Dutch book*) se comporte nécessairement comme s'il utilisait des probabilités satisfaisant les axiomes de Kolmogorov.

Théorème 1.2 (Théorème de représentation de de Finetti)**.** *Si $(X_1, X_2, \ldots)$ est une suite échangeable de variables aléatoires à valeurs dans $\{0, 1\}$, alors il existe une mesure de probabilité μ sur $[0, 1]$ telle que :*

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

où $s_n = \sum_{i=1}^n x_i$.

Remarque 1.3. Ce théorème justifie le modèle bayésien : l'échangeabilité (hypothèse plus faible que l'indépendance) implique l'existence d'un paramètre latent θ avec une distribution a priori μ.

1.2.2 Axiomes de Savage

Théorème 1.4 (Théorème de Savage, 1954)**.** *Sous les sept axiomes de Savage (ordonnancement, sure-thing principle, etc.), il existe :*

1. *une mesure de probabilité subjective $\mathbb{P}$ sur l'espace des états,*

2. *une fonction d'utilité u unique à transformation affine près,*

telles que l'agent préfère l'action a à l'action b si et seulement si $\mathbb{E}_{\mathbb{P}}[u(a)] > \mathbb{E}_{\mathbb{P}}[u(b)]$.

1.3 Le théorème de Bayes

Théorème 1.5 (Théorème de Bayes)**.** *Soit $\theta \in \Theta$ un paramètre de loi a priori $\pi(\theta)$ et $x = (x_1, \ldots, x_n)$ un échantillon de densité $f(x \mid \theta)$. La loi a posteriori est :*

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

Démonstration. Par la règle de Bayes sur les densités conditionnelles :

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

□

Composantes du théorème de Bayes

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

1.4 Choix de la loi a priori

1.4.1 Types de priors

Définition 1.6 (Prior informatif)**.** Un **prior informatif** est une loi $\pi(\theta)$ qui concentre sa masse sur une région restreinte de Θ, reflétant une connaissance préalable substantielle.

Définition 1.7 (Prior vague (non informatif))**.** Un **prior vague** ou **non informatif** est une loi $\pi(\theta)$ qui cherche à laisser les données dominer l'inférence. Exemples : prior uniforme, prior de Jeffreys.

Définition 1.8 (Prior de Jeffreys)**.** Le **prior de Jeffreys** est défini par :

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

où $I(\theta)$ est la matrice d'information de Fisher.

Proposition 1.9 (Invariance du prior de Jeffreys)**.** Le prior de Jeffreys est invariant par reparamétrisation : si $\phi = g(\theta)$ est une bijection différentiable, alors $\pi_J(\phi) \propto \sqrt{\det I_\phi(\phi)}$.

Démonstration. Soit $\phi = g(\theta)$. Par la règle de changement de variable et la transformation de l'information de Fisher :

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

d'où $\sqrt{I_\phi(\phi)} = \left|\frac{d\theta}{d\phi}\right| \sqrt{I(\theta)}$, ce qui correspond exactement au jacobien du changement de variable. □

Exemple 1.10. Pour $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ avec σ^2 connu, l'information de Fisher pour μ est $I(\mu) = n/\sigma^2$, constante. Le prior de Jeffreys est donc $\pi_J(\mu) \propto 1$, c'est-à-dire le prior uniforme (impropre) sur $\mathbb{R}$.

1.4.2 Prior de référence

Définition 1.11 (Prior de référence de Bernardo)**.** Le **prior de référence** maximise l'information attendue de Kullback–Leibler entre la loi a priori et la loi a posteriori :

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

1.5 Exemples fondamentaux

Exemple 1.12. Modèle Bernoulli–Bêta. Soit $X_1, \dots, X_n \overset{\text{iid}}{\sim} \text{Ber}(\theta)$ et $\theta \sim \text{Be}(a, b)$. Posons $s = \sum x_i$. Alors :

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

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

Exemple 1.13. Modèle Normal–Normal (variance connue). Soit $X_1, \dots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ avec σ^2 connu, et $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$. La loi a posteriori est :

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

où

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

Moyenne a posteriori comme moyenne pondérée

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

Quand $n \to \infty$, $w \to 0$ et $\mu_n \to \bar{x}$: le posterior converge vers l'estimateur du maximum de vraisemblance.

1.6 Propriétés asymptotiques

Théorème 1.14 (Théorème de Bernstein–von Mises)**.** *Sous des conditions de régularité, la loi a posteriori converge en loi vers une gaussienne centrée sur le maximum de vraisemblance :*

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

où θ_0 est la vraie valeur du paramètre.

Remarque 1.15. Ce résultat montre que, pour de grands échantillons, le choix du prior a un impact négligeable. Autrement dit, les approches bayésienne et fréquentiste convergent asymptotiquement.

Théorème 1.16 (Consistance a posteriori)**.** *Soit θ_0 la vraie valeur. Sous des conditions de régularité (identifiabilité, support du prior contenant θ_0), pour tout voisinage U de θ_0 :*

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

1.7 Principe de vraisemblance

Définition 1.17 (Principe de vraisemblance)**.** Deux expériences produisant la même fonction de vraisemblance $L(\theta \mid x)$ (á une constante multiplicative près) doivent conduire à la même inférence sur θ.

Proposition 1.18. L'inférence bayésienne satisfait automatiquement le principe de vraisemblance, car la loi a posteriori ne dépend de x qu'à travers $L(\theta \mid x)$.

Violation fréquentiste

Certaines procédures fréquentistes (comme les p-valeurs) violent le principe de vraisemblance car elles dépendent de l'espace échantillonnal et du plan d'expérience, pas seulement des données observées.

1.8 Implémentation Python

Modèle Bernoulli–Bêta avec PyMC

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

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

# Modèle bayésien
with pm.Model() as model_beta_binom:
    # Prior Beta(2, 5)
    theta = pm.Beta("theta", alpha=2, beta=5)
    # Vraisemblance
    y_obs = pm.Bernoulli("y_obs", p=theta, observed=data)
    # Échantillonnage MCMC
    trace = pm.sample(2000, tune=1000, random_seed=42)

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

# Vérification analytique
a_post = 2 + data.sum()
b_post = 5 + n - data.sum()
print(f"Posterior analytique: Be({a_post}, {b_post})")
print(f"Moyenne analytique: {a_post/(a_post+b_post):.4f}")
```

Visualisation prior–posterior

```python
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="--")
# Posterior analytique
ax.plot(theta_grid, stats.beta.pdf(theta_grid, a_post, b_post),
        label=f"Posterior Be({a_post}, {b_post})")
# Vraisemblance (normalisée pour affichage)
s = data.sum()
like = theta_grid**s * (1-theta_grid)**(n-s)
like /= like.max() * 3
ax.plot(theta_grid, like, label="Vraisemblance (norm.)",
        linestyle=":")
ax.axvline(theta_true, color="red", alpha=0.5,
           label=f"Vraie valeur ({theta_true})")
ax.set_xlabel(r"$\theta$")
ax.set_ylabel("Densité")
ax.legend()
ax.set_title("Mise à jour bayésienne: Bernoulli-Bêta")
plt.tight_layout()
plt.savefig("bayes_update.pdf")
```

1.9 Exercices

Exercice 1.1. Soit $X_1, \dots, X_n \overset{\text{iid}}{\sim} \text{Poi}(\lambda)$ avec prior $\lambda \sim \text{Ga}(\alpha, \beta)$.

1. Déterminer la loi a posteriori de λ.
2. Calculer la moyenne et la variance a posteriori.
3. Montrer que la moyenne a posteriori est une moyenne pondérée de la moyenne a priori et de la moyenne empirique.

Exercice 1.2. Calculer le prior de Jeffreys pour le modèle $\text{Ber}(\theta)$ et montrer qu'il s'agit de $\text{Be}(1/2, 1/2)$.

Exercice 1.3. Démontrer que, dans le modèle Normal–Normal, la précision a posteriori est la somme de la précision a priori et de la précision des données. Interpréter ce résultat.

Exercice 1.4. Considérer deux expériences :

1. On lance une pièce $n = 12$ fois et on observe $s = 3$ succès.
2. On lance une pièce jusqu'au troisième succès et on observe $n = 12$ lancers.

Montrer que les vraisemblances sont proportionnelles. En déduire que l'inférence bayésienne est identique, alors que les p-valeurs fréquentistes diffèrent.

Exercice 1.5. (Numérique) Reprendre le modèle Normal–Normal avec $\sigma^2 = 1$, $\mu_0 = 0$, $\tau_0^2 = 10$ et un échantillon de taille $n = 5$ simulé depuis $\mathcal{N}(2, 1)$.

1. Calculer analytiquement le posterior.
2. Vérifier avec PyMC.
3. Tracer le prior, le posterior et la vraisemblance normalisée sur un même graphique.

Chapitre 2

Lois A Priori et A Posteriori — Conjugaison

Le théorème de Bayes nous dit *comment* passer du prior au posterior, mais il ne nous dit pas *quel* prior choisir. Ce choix est le cœur du paradigme bayésien, et la famille des *lois conjuguées* offre une réponse élégante : si le prior et la vraisemblance « s'accordent » bien, le posterior appartient à la même famille paramétrique que le prior, et la mise à jour se réduit à une simple modification des hyperparamètres. C'est une idée ancienne — elle remonte à Raiffa et Schlaifer (1961) — mais toujours aussi utile aujourd'hui, y compris comme brique de base des méthodes variationnelles.

Idée centrale

Une famille conjuguée est un couple (vraisemblance, prior) tel que le posterior appartient à la même famille paramétrique que le prior. Cela garantit des calculs analytiques et une interprétation transparente de la mise à jour bayésienne.

2.1 Famille exponentielle

Définition 2.1 (Famille exponentielle naturelle)**.** Une famille de lois $\{f(x \mid \theta) : \theta \in \Theta\}$ appartient à la **famille exponentielle** si elle s'écrit :

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

où $T(x)$ est la statistique suffisante, $\eta(\theta)$ le paramètre naturel et $A(\theta)$ le log-normalisateur.

Proposition 2.2 (Propriétés du log-normalisateur)**.**

1. $\mathbb{E}[T(X)] = \nabla_\eta A(\eta)$.
2. $\mathrm{Var}[T(X)] = \nabla^2_\eta A(\eta)$ (matrice hessienne).
3. $A(\eta)$ est convexe.

Théorème 2.3 (Existence de la conjuguée)**.** *Toute famille exponentielle admet un prior conjugué de la forme :*

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

où χ_0 et ν_0 sont les hyperParamètres. Après observation de $x_1, \dots, x_n$, le posterior est de la même forme avec :

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

Démonstration. La vraisemblance de l'échantillon s'écrit :

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

En multipliant par le prior conjugué :

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

qui est de la même forme avec (χ_n, ν_n). □

2.2 Table des familles conjuguées

Principales familles conjuguées

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

2.3 Exemples détaillés

2.3.1 Bernoulli–Bêta

Exemple 2.4. Soit $X_1, \dots, X_n \overset{\text{iid}}{\sim} \mathrm{Ber}(\theta)$ avec $\theta \sim \mathrm{Be}(a,b)$. La statistique suffisante est $s = \sum x_i$. Le posterior est $\mathrm{Be}(a + s, b + n - s)$.

Interprétation : le prior $\mathrm{Be}(a,b)$ équivaut à $a-1$ « succès »virtuels et $b-1$ « échecs »virtuels. L'observation ajoute s succès et $n - s$ échecs.

Moments a posteriori :

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

2.3.2 Poisson–Gamma

Exemple 2.5. Soit $X_1, \dots, X_n \overset{\text{iid}}{\sim} \text{Poi}(\lambda)$ avec $\lambda \sim \text{Ga}(\alpha, \beta)$. Alors :

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

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

La moyenne a posteriori est :

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

C'est une moyenne pondérée de la moyenne a priori α/β et de la moyenne empirique $\bar{x}$.

2.3.3 Normal–Normal (variance connue)

Théorème 2.6 (Conjugaison Normal–Normal)**.** *Soit $X_1, \dots, X_n \overset{iid}{\sim} \mathcal{N}(\mu, \sigma^2)$ avec σ^2 connu et $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$. Alors $\mu \mid x \sim \mathcal{N}(\mu_n, \tau_n^2)$ avec :*

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

Démonstration. La log-densité a posteriori (en ignorant les constantes) est :

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

ce qui correspond à une gaussienne de précision $1/\tau_n^2 = 1/\tau_0^2 + n/\sigma^2$ et de moyenne μ_n. □

2.3.4 Normal–Inverse-Gamma (moyenne connue)

Théorème 2.7 (Conjugaison pour la variance)**.** *Soit $X_1, \dots, X_n \overset{iid}{\sim} \mathcal{N}(\mu_0, \sigma^2)$ avec μ_0 connu et $\sigma^2 \sim IG(\alpha, \beta)$. Alors :*

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

2.3.5 Normale–Normale-Inverse-Gamma (cas général)

Théorème 2.8 (Conjugaison NIG)**.** *Soit $X_1, \dots, X_n \overset{iid}{\sim} \mathcal{N}(\mu, \sigma^2)$ avec prior conjoint $(\mu, \sigma^2) \sim NIG(\mu_0, \kappa_0, \alpha_0, \beta_0)$:*

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

Le posterior est $NIG(\mu_n, \kappa_n, \alpha_n, \beta_n)$ avec :

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

2.3.6 Multinomiale–Dirichlet

Exemple 2.9. Soit $\mathbf{x} \sim \mathrm{Mult}(n, \mathbf{p})$ avec $\mathbf{p} \sim \mathrm{Dir}(\boldsymbol{\alpha})$. Le posterior est :

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

La moyenne a posteriori de p_k est :

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

2.4 Priors non conjugués

Remarque 2.10. Quand le prior n'est pas conjugué, le posterior n'a pas de forme fermée. On doit alors recourir à :

- l'approximation de Laplace,
- les méthodes MCMC (chapitres 7–8),
- l'inférence variationnelle (chapitre 9).

Définition 2.11 (Approximation de Laplace)**.** L'approximation de Laplace approche le posterior par une gaussienne centrée sur le MAP (Maximum A Posteriori) :

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

2.5 Hyperpriors et priors hiérarchiques

Définition 2.12 (Hyperprior)**.** Un **hyperprior** est une loi placée sur les hyperparamètres du prior. Par exemple, si $\theta \sim \mathrm{Be}(a, b)$, on peut poser $a \sim \mathrm{Ga}(1, 1)$ et $b \sim \mathrm{Ga}(1, 1)$.

Remarque 2.13. Les hyperpriors mènent aux modèles hiérarchiques, traités en détail au chapitre 6.

2.6 Sensibilité au prior

Définition 2.14 (Analyse de sensibilité)**.** L'**analyse de sensibilité** consiste à évaluer la robustesse des conclusions bayésiennes en faisant varier le prior dans une classe $\Gamma = \{\pi_\gamma : \gamma \in \mathcal{G}\}$ et en étudiant l'étendue des quantités a posteriori.

Proposition 2.15 (Convergence au MLE)**.** Pour tout prior $\pi(\theta)$ absolument continu avec $\pi(\theta_0) > 0$, la moyenne a posteriori converge vers le MLE quand $n \to \infty$. Ainsi, l'influence du prior disparaît asymptotiquement.

2.7 Implémentation Python

Conjugaison Poisson–Gamma

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

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

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

# Posterior analytique
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"Moyenne post: {alpha_post/beta_post:.3f}")
print(f"MLE: {data.mean():.3f}")

# Visualisation
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"Vraie valeur ({lam_true})")
ax.set_xlabel(r"$\lambda$")
ax.set_ylabel("Densité")
ax.legend()
ax.set_title("Conjugaison Poisson-Gamma")
plt.tight_layout()
plt.savefig("poisson_gamma.pdf")
```

Conjugaison Multinomiale–Dirichlet

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

# Données catégorielles (K=4 catégories)
counts = np.array([45, 30, 15, 10])
```

```python
K = len(counts)
alpha_prior = np.ones(K)  # Dirichlet(1,...,1) = uniforme

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

# Vérification PyMC
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 Exercices

Exercice 2.1. Montrer que la famille exponentielle $X \sim \text{Exp}(\lambda)$ admet le prior conjugué $\text{Ga}(\alpha, \beta)$ et déterminer le posterior.

Exercice 2.2. Soit $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ avec (μ, σ^2) tous deux inconnus. Utiliser le prior NIG et calculer le posterior. Montrer que la loi marginale a posteriori de μ est une Student.

Exercice 2.3. Pour le modèle géométrique $X \sim \text{Geom}(\theta)$ avec $\mathbb{P}(X = k) = \theta(1-\theta)^k$ pour $k = 0, 1, 2, \ldots$:

1. Vérifier que ce modèle appartient à la famille exponentielle.
2. Déterminer le prior conjugué.
3. Calculer la moyenne a posteriori.

Exercice 2.4. Effectuer une analyse de sensibilité pour le modèle Bernoulli–Bêta avec $n = 10$ et $s = 3$. Comparer les posteriors obtenus avec les priors $\text{Be}(1, 1)$, $\text{Be}(0.5, 0.5)$, $\text{Be}(2, 8)$ et $\text{Be}(5, 5)$. Tracer les quatre posteriors sur un même graphique.

Exercice 2.5. (Théorique) Démontrer que le prior conjugué pour la famille exponentielle générale est de la forme donnée dans le théorème de cette section. Vérifier sur les cas Bernoulli, Poisson et Normal.

Chapitre 3

Estimation Bayésienne et Fonctions de Perte

En statistique fréquentiste, un estimateur est « bon »s'il est sans biais et de variance minimale. En statistique bayésienne, la question est formulée différemment : quel résumé de la loi a posteriori choisir, et à quel coût ? La réponse dépend de ce que l'on considère comme une erreur grave. Se tromper de beaucoup est-il proportionnellement pire que se tromper de peu (perte quadratique), ou toute erreur au-delà d'un seuil est-elle également inacceptable (perte 0-1) ? Abraham Wald, dans ses travaux fondateurs des années 1940 sur la théorie de la décision, a montré que cette question de la *fonction de perte* est inévitable : tout estimateur est, consciemment ou non, la solution d'un problème de décision.

Ce chapitre explore ce lien entre estimation et décision, et montre comment chaque fonction de perte conduit à un estimateur optimal différent : la moyenne a posteriori, la médiane a posteriori, le mode a posteriori.

Idée centrale

L'estimation bayésienne ne se réduit pas à un estimateur ponctuel unique : elle fournit toute la loi a posteriori. Néanmoins, lorsqu'un résumé ponctuel est nécessaire, le choix de l'estimateur optimal dépend de la **fonction de perte** retenue.

3.1 Théorie de la décision bayésienne

Définition 3.1 (Problème de décision)**.** Un **problème de décision** est un triplet $(\Theta, \mathcal{A}, L)$ où :

- Θ est l'espace des paramètres,
- $\mathcal{A}$ est l'espace des actions (décisions),
- $L : \Theta \times \mathcal{A} \to \mathbb{R}^+$ est la fonction de perte.

Définition 3.2 (Risque a posteriori)**.** Le **risque a posteriori** (ou perte espérée a posteriori) d'une action $a \in \mathcal{A}$ est :

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

Définition 3.3 (Estimateur de Bayes)**.** L'**estimateur de Bayes** sous la perte L est l'action qui minimise le risque a posteriori :

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

Définition 3.4 (Risque bayésien (intégré))**.** Le **risque bayésien** d'un estimateur $\delta(x)$ est :

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

Théorème 3.5 (Optimalité de l'estimateur de Bayes)**.** *L'estimateur de Bayes $\hat{\theta}_B$ minimise le risque bayésien $r(\pi, \delta)$ parmi tous les estimateurs.*

Démonstration. Le risque bayésien peut s'écrire :

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

où $m(x)$ est la vraisemblance marginale. Comme $m(x) \geq 0$, minimiser $r(\pi, \delta)$ revient à minimiser $\rho(\pi, \delta(x) \mid x)$ pour chaque x, ce qui donne $\hat{\theta}_B(x)$. □

3.2 Fonctions de perte classiques

Estimateurs de Bayes selon la perte

Fonction de perte	**Expression**	**Estimateur de Bayes**
Quadratique	$L(\theta, a) = (\theta - a)^2$	$\mathbb{E}[\theta \mid x]$ (moyenne post.)
Valeur absolue	$L(\theta, a) = \lvert\theta - a\rvert$	$\mathrm{Med}[\theta \mid x]$ (médiane post.)
0–1 (tout ou rien)	$L(\theta, a) = \mathbf{1}(\lvert\theta - a\rvert > \varepsilon)$	$\mathrm{Mode}[\theta \mid x]$ (MAP)
LINEX	$L(\theta, a) = e^{c(\theta - a)} - c(\theta - a) - 1$	$\frac{1}{c} \log \mathbb{E}[e^{c\theta} \mid x]$
Quadratique pondérée	$L(\theta, a) = w(\theta)(\theta - a)^2$	$\frac{\mathbb{E}[w(\theta)\theta \mid x]}{\mathbb{E}[w(\theta) \mid x]}$

3.2.1 Perte quadratique

Théorème 3.6 (Estimateur de Bayes sous perte quadratique)**.** *Sous la perte quadratique $L(\theta, a) = (\theta - a)^2$, l'estimateur de Bayes est la moyenne a posteriori :*

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

Démonstration.

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

Ce qui s'annule pour $a = \mathbb{E}[\theta \mid x]$. La dérivée seconde est $2 > 0$, confirmant un minimum. □

Corollaire 3.7. *Le risque a posteriori sous perte quadratique est la variance a posteriori :*

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

3.2.2 Perte valeur absolue

Théorème 3.8 (Estimateur de Bayes sous perte absolue)**.** *Sous la perte $L(\theta, a) = |\theta - a|$, l'estimateur de Bayes est la médiane a posteriori.*

Démonstration. La dérivée généralisée du risque a posteriori est :

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

qui s'annule quand $\mathbb{P}(\theta < a \mid x) = 1/2$, i.e., a est la médiane de $\pi(\theta \mid x)$. □

3.2.3 Maximum a posteriori (MAP)

Définition 3.9 (Estimateur MAP)**.** L'**estimateur du maximum a posteriori** (MAP) est :

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

Remarque 3.10. Le MAP coïncide avec le MLE quand le prior est uniforme. Avec un prior gaussien $\mathcal{N}(0, \tau^2)$, le MAP correspond à la régularisation de Ridge :

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

Proposition 3.11 (MAP et régularisation Lasso)**.** Si le prior est un Laplace $\pi(\theta) \propto e^{-\lambda|\theta|}$, le MAP correspond à la régularisation Lasso (ℓ_1).

3.3 Intervalles de crédibilité

Définition 3.12 (Intervalle de crédibilité)**.** Un **intervalle de crédibilité** à $100(1-\alpha)\%$ est un intervalle $[a, b]$ tel que :

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

Définition 3.13 (HDI — Highest Density Interval)**.** L'**intervalle de plus haute densité** (HDI) est l'intervalle de crédibilité de longueur minimale :

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

où c_α est choisi tel que $\mathbb{P}(\pi(\theta \mid x) \geq c_\alpha \mid x) = 1 - \alpha$.

Crédibilité vs. confiance

Un intervalle de crédibilité à 95% signifie que θ appartient à l'intervalle avec probabilité 0.95 *conditionnellement aux données observées*. Un intervalle de confiance à 95% signifie que la procédure contient θ dans 95% des répétitions *hypothétiques*.

Exemple 3.14. Pour le modèle Bernoulli–Bêta avec $n = 100$, $s = 30$, $\theta \sim \text{Be}(1, 1)$, le posterior est $\text{Be}(31, 71)$.

L'intervalle de crédibilité équi-queue à 95% est :

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

3.4 Comparaison bayésien–fréquentiste

Définition 3.15 (Risque fréquentiste)**.** Le risque fréquentiste d'un estimateur $\delta(x)$ est :

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

Définition 3.16 (Admissibilité)**.** Un estimateur δ est **admissible** s'il n'existe pas d'estimateur δ' tel que $R(\theta, \delta') \leq R(\theta, \delta)$ pour tout θ, avec inégalité stricte pour au moins un θ.

Théorème 3.17 (Admissibilité des estimateurs de Bayes)**.** *Sous des conditions de régularité, tout estimateur de Bayes dont le risque bayésien est fini est admissible.*

Théorème 3.18 (Classe complète)**.** *Sous des conditions générales, la classe des estimateurs de Bayes (propres et limites) forme une* ***classe complète*** *: tout estimateur admissible est un estimateur de Bayes (éventuellement généralisé).*

Exemple 3.19. Estimateur de James–Stein. Pour $X \sim \mathcal{N}_p(\theta, I_p)$ avec $p \geq 3$, l'estimateur $\hat{\theta}_{\text{JS}} = (1 - \frac{p-2}{\|X\|^2})X$ domine le MLE $\hat{\theta} = X$ sous perte quadratique. C'est un estimateur de Bayes généralisé.

3.5 Estimation ponctuelle : exemples

Exemple 3.20. Modèle Poisson–Gamma. Avec $X_i \sim \text{Poi}(\lambda)$ et $\lambda \sim \text{Ga}(\alpha, \beta)$, le posterior est $\text{Ga}(\alpha + \sum x_i, \beta + n)$. Les estimateurs ponctuels sont :

$$\begin{aligned}
\hat{\lambda}_{\text{moyenne}} &= \frac{\alpha + \sum x_i}{\beta + n}, \\
\hat{\lambda}_{\text{MAP}} &= \frac{\alpha + \sum x_i - 1}{\beta + n} \quad (\text{si } \alpha + \sum x_i > 1), \\
\hat{\lambda}_{\text{médiane}} &\approx \hat{\lambda}_{\text{moyenne}} - \frac{1}{3(\beta + n)} \quad (\text{approx. pour grande } \alpha').
\end{aligned}$$

Exemple 3.21. Modèle Normal–Normal. Avec $X_i \sim \mathcal{N}(\mu, \sigma^2)$, σ^2 connu, et $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$: le posterior est gaussien donc symétrique, et les trois estimateurs coïncident :

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

3.6 Perte LINEX et estimation asymétrique

Définition 3.22 (Perte LINEX)**.** La perte LINEX (*LINear-EXponential*) est :

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

Pour $c > 0$, la surestimation est pénalisée exponentiellement. Pour $c < 0$, c'est la sous-estimation.

Théorème 3.23 (Estimateur de Bayes sous perte LINEX)**.** *L'estimateur de Bayes sous perte LINEX est :*

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

Démonstration. Le risque a posteriori est :

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

En dérivant par rapport à a et en annulant :

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

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

3.7 Rétrécissement bayésien (*shrinkage*)

Proposition 3.24 (Rétrécissement vers la moyenne a priori)**.** Dans le modèle Normal–Normal, la moyenne a posteriori s'écrit :

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

où $B \in (0, 1)$ est le **facteur de rétrécissement**. L'estimateur bayésien est rétréci vers la moyenne a priori μ_0.

3.8 Implémentation Python

Estimation bayésienne et intervalles de crédibilité

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

# Données Bernoulli
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)

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

print(f"Moyenne a posteriori: {mean_post:.4f}")
print(f"Médiane a posteriori: {median_post:.4f}")
print(f"MAP: {mode_post:.4f}")
print(f"MLE: {s/n:.4f}")

# Intervalles de crédibilité
ci_equi = posterior.ppf([0.025, 0.975])
print(f"IC équi-queue 95%: [{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"HDI 95%: [{hdi[0]:.4f}, {hdi[1]:.4f}]")
```

Comparaison des fonctions de perte

```python
# Visualisation des fonctions de perte
delta = np.linspace(-3, 3, 500)

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

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

# Perte valeur absolue
axes[1].plot(delta, np.abs(delta))
axes[1].set_title("Perte valeur absolue")
axes[1].set_xlabel(r"$\theta - a$")

# Perte LINEX (c=1 et 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("Perte LINEX")
axes[2].set_xlabel(r"$\theta - a$")
axes[2].legend()

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

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

Effet de rétrécissement bayésien

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

# Simulation: J groupes, estimation simultanée
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])

# Estimateurs MLE vs Bayes (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="Vrai $\\mu_j$", zorder=3)
ax.scatter(range(J), x_bar, marker="o",
           label="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("Groupe $j$")
ax.set_ylabel(r"$\mu_j$")
ax.legend()
ax.set_title("Rétrécissement bayésien")
plt.tight_layout()
plt.savefig("shrinkage.pdf")
```

3.9 Exercices

Exercice 3.1. Démontrer que l'estimateur de Bayes sous la perte pondérée $L(\theta, a) = w(\theta)(\theta - a)^2$ est :

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

Exercice 3.2. Pour le modèle Poisson–Gamma, calculer l'estimateur de Bayes sous la perte LINEX avec $c = 1$ et comparer avec la moyenne a posteriori.

Exercice 3.3. Montrer que dans le cas gaussien unidimensionnel, le HDI coïncide avec l'intervalle équi-queue.

Exercice 3.4. Soit $X \sim \mathcal{N}_p(\theta, I_p)$ avec $\theta \sim \mathcal{N}_p(0, \tau^2 I_p)$.

1. Calculer l'estimateur de Bayes sous perte quadratique.
2. Montrer qu'il s'agit d'un estimateur de rétrécissement de la forme $c \cdot X$ avec $c \in (0, 1)$.
3. Comparer avec l'estimateur de James–Stein.

Exercice 3.5. (Numérique) Simuler $n = 50$ observations de $\mathcal{N}(3, 4)$. Comparer les estimateurs bayésiens de μ (moyenne, médiane, MAP) avec le MLE pour différents priors : $\mathcal{N}(0, 100)$ (vague), $\mathcal{N}(0, 1)$ (informatif centré), $\mathcal{N}(10, 1)$ (informatif décentré).

Chapitre 4

Tests d'Hypothèses Bayésiens

Le test d'hypothèse fréquentiste repose sur la p-valeur : la probabilité d'observer des données au moins aussi extrêmes que celles obtenues, sous l'hypothèse nulle. Mais cette quantité répond à une question que personne ne pose vraiment : ce qu'on veut savoir, c'est la probabilité que l'hypothèse soit vraie *étant donné les données*, pas l'inverse. Le test bayésien répond directement à cette question grâce au *facteur de Bayes* : le rapport des vraisemblances marginales sous chaque hypothèse. Harold Jeffreys, dans son traité de 1961, a systématisé cette approche et proposé l'échelle qui porte son nom pour interpréter la force de l'évidence.

Idée centrale

Le test bayésien compare les hypothèses via leurs probabilités a posteriori ou le **facteur de Bayes**, qui mesure la force relative des évidences en faveur de chaque hypothèse, sans dépendre de la règle d'arrêt ou de l'espace échantillonnal.

4.1 Formulation bayésienne du test

Définition 4.1 (Test bayésien)**.** Soient deux hypothèses $H_0 : \theta \in \Theta_0$ et $H_1 : \theta \in \Theta_1$ avec $\Theta_0 \cup \Theta_1 = \Theta$ et $\Theta_0 \cap \Theta_1 = \emptyset$. On attribue des probabilités a priori $\mathbb{P}(H_0) = \pi_0$ et $\mathbb{P}(H_1) = \pi_1 = 1 - \pi_0$. Les probabilités a posteriori sont :

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

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

Définition 4.2 (Facteur de Bayes)**.** Le **facteur de Bayes** en faveur de H_0 contre H_1 est :

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

Rapport des cotes a posteriori

$$\underbrace{\frac{\mathbb{P}(H_0 \mid x)}{\mathbb{P}(H_1 \mid x)}}_{\text{cote post.}} = \underbrace{B_{01}}_{\text{facteur de Bayes}} \times \underbrace{\frac{\pi_0}{\pi_1}}_{\text{cote a priori}}.$$

4.2 Interprétation du facteur de Bayes

Échelle de Jeffreys pour B_{01}

$\log_{10} B_{01}$	**Évidence en faveur de H_0**
0 à 1/2	Faible
1/2 à 1	Substantielle
1 à 2	Forte
> 2	Décisive

Remarque 4.3. Le facteur de Bayes est symétrique : $B_{10} = 1/B_{01}$. Si $B_{01} > 1$, les données soutiennent H_0 ; si $B_{01} < 1$, elles soutiennent H_1.

4.3 Test d'une hypothèse ponctuelle

Définition 4.4 (Hypothèse ponctuelle)**.** On teste $H_0 : \theta = \theta_0$ contre $H_1 : \theta \neq \theta_0$. On attribue une masse $\pi_0 = \mathbb{P}(\theta = \theta_0) > 0$ et une densité continue $\pi_1(\theta)$ sur Θ_1.

Théorème 4.5 (Facteur de Bayes pour H_0 ponctuelle)**.**

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

Paradoxe de Jeffreys–Lindley

Avec un prior vague ($\tau_0^2 \to \infty$) sous H_1, le facteur de Bayes $B_{01} \to \infty$, favorisant toujours H_0, même quand le p-value fréquentiste est très petit. Ce paradoxe illustre l'importance du choix du prior sous H_1.

Exemple 4.6. Test $\mu = 0$ dans le modèle gaussien. Soit $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ avec σ^2 connu. Sous $H_0 : \mu = 0$, sous $H_1 : \mu \sim \mathcal{N}(0, \tau^2)$.

Le facteur de Bayes est :

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

4.4 Règle de décision bayésienne

Définition 4.7 (Règle de décision optimale)**.** Sous la perte 0–1 :

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

la décision optimale est : accepter H_0 si $\mathbb{P}(H_0 \mid x) > \mathbb{P}(H_1 \mid x)$, soit $B_{01} > \pi_1/\pi_0$.

Avec $\pi_0 = \pi_1 = 1/2$, cela revient à accepter H_0 si $B_{01} > 1$.

Théorème 4.8 (Règle de décision sous perte asymétrique)**.** *Sous la perte $L(H_0, d = 1) = c_0$ (erreur de type I) et $L(H_1, d = 0) = c_1$ (erreur de type II), la règle optimale est : accepter H_0 si*

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

4.5 Calcul du facteur de Bayes

4.5.1 Cas conjugué

Proposition 4.9 (Facteur de Bayes en conjugaison)**.** Pour un modèle conjugué, la vraisemblance marginale est souvent calculable analytiquement. Par exemple, pour le modèle Bernoulli–Bêta avec $\theta \sim \mathrm{Be}(a, b)$:

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

où $B(\cdot, \cdot)$ est la fonction bêta.

4.5.2 Méthode de Savage–Dickey

Théorème 4.10 (Rapport de densité de Savage–Dickey)**.** *Pour tester $H_0 : \theta = \theta_0$ contre $H_1 : \theta \sim \pi_1(\theta)$, si π_1 est le prior sous H_1 et $\pi_1(\theta \mid x)$ le posterior correspondant :*

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

Démonstration. Par définition :

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

Or $\pi_1(\theta_0 \mid x) = f(x \mid \theta_0)\pi_1(\theta_0)/m_1(x)$, d'où $f(x \mid \theta_0)/m_1(x) = \pi_1(\theta_0 \mid x)/\pi_1(\theta_0)$. □

4.5.3 Approximations numériques

Définition 4.11 (Approximation harmonique)**.** La vraisemblance marginale peut être estimée par la moyenne harmonique des vraisemblances évaluées sur les échantillons MCMC :

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

Instabilité de la moyenne harmonique

Cette estimation a une variance potentiellement infinie et ne doit être utilisée qu'avec prudence. On préfère des méthodes plus stables comme le *bridge sampling* ou WAIC.

4.6 Critères d'information bayésiens

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

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

où k est le nombre de paramètres. Le BIC est une approximation du log-facteur de Bayes : $-2 \log B_{01} \approx \mathrm{BIC}_0 - \mathrm{BIC}_1$.

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

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

où $p_{\mathrm{WAIC}} = \sum_{i=1}^{n} \mathrm{Var}_{\theta \mid x}[\log f(x_i \mid \theta)]$ est le nombre effectif de paramètres.

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

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

où $f(x_i \mid x_{-i}) = \int f(x_i \mid \theta)\, \pi(\theta \mid x_{-i})\, d\theta$ est la densité prédictive *leave-one-out*.

4.7 ROPE et équivalence pratique

Définition 4.15 (ROPE — Region of Practical Equivalence)**.** La **ROPE** est un intervalle $[\theta_0 - \varepsilon, \theta_0 + \varepsilon]$ définissant l'équivalence pratique à θ_0. La décision est :

- Accepter H_0 si HDI $\subset$ ROPE.
- Rejeter H_0 si HDI $\cap$ ROPE $= \emptyset$.
- Indécis sinon.

4.8 Comparaison avec les p-valeurs

Proposition 4.16 (Calibration Bayes–p-value)**.** Sellke, Bayarri et Berger (2001) montrent que :

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

où p est le p-value. Ainsi, un p-value de 0.05 correspond à $B_{01} \geq 2.5$ au plus, soit une évidence faible à modérée pour H_0.

4.9 Implémentation Python

Facteur de Bayes pour un test gaussien

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

# Données
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()

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

# Facteur de Bayes analytique
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("Données en faveur de H0")
else:
    print(f"Données en faveur de H1 (BF10 = {1/BF01:.4f})")

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

ROPE et HDI avec PyMC

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

# Modèle bayésien
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 et 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 du posterior dans la 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}")
```

Comparaison de modèles avec WAIC

```
# Deux modèles en compétition
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)

# Comparaison WAIC
waic_lin = az.waic(trace_lin, model_linear)
waic_null = az.waic(trace_null, model_null)
comp = az.compare({"linear": trace_lin,
                   "null": trace_null})
print(comp)
```

4.10 Exercices

Exercice 4.1. Calculer le facteur de Bayes pour le test $H_0 : \theta = 1/2$ contre $H_1 : \theta \sim \text{Be}(1, 1)$ dans le modèle Bernoulli avec $n = 20$ et $s = 14$.

Exercice 4.2. Démontrer le rapport de Savage–Dickey. Vérifier numériquement sur le

modèle Normal–Normal.

Exercice 4.3. Illustrer le paradoxe de Jeffreys–Lindley : pour $n = 100$, $\bar{x} = 0.2$, $\sigma = 1$, calculer le p-value et le facteur de Bayes B_{01} pour différentes valeurs de $\tau \in \{1, 10, 100\}$. Commenter.

Exercice 4.4. (Numérique) Comparer WAIC et LOO-CV pour la sélection entre un modèle linéaire et un modèle quadratique sur des données simulées.

Exercice 4.5. Dériver l'approximation BIC du facteur de Bayes à partir de l'approximation de Laplace de la vraisemblance marginale.

Chapitre 5

Prédiction Bayésienne

En statistique fréquentiste, la prédiction utilise un estimateur ponctuel : on estime θ, puis on prédit $\tilde{y}$ comme si $\hat{\theta}$ était la vraie valeur. Cette approche ignore l'incertitude sur le paramètre et produit des intervalles de prédiction trop étroits. L'approche bayésienne est fondamentalement différente : au lieu de fixer θ, on *intègre* sur toutes les valeurs possibles de θ, pondérées par la loi a posteriori. La distribution prédictive qui en résulte incorpore naturellement deux sources d'incertitude — l'aléa intrinsèque des données et l'incertitude sur le modèle — et produit des prévisions mieux calibrées.

Idée centrale

La prédiction bayésienne intègre l'incertitude sur le paramètre en moyennant la distribution prédictive sur le posterior. Cela produit des intervalles prédictifs calibrés et évite le sur-ajustement inhérent à l'utilisation d'estimations ponctuelles.

5.1 Distribution prédictive a priori

Définition 5.1 (Distribution prédictive a priori)**.** La **distribution prédictive a priori** (ou marginale) d'une nouvelle observation $\tilde{x}$ est :

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

Elle ne dépend pas des données et représente la prédiction avant toute observation.

Exemple 5.2. Modèle Bernoulli–Bêta. Avec $\theta \sim \mathrm{Be}(a, b)$ et $\tilde{X} \mid \theta \sim \mathrm{Ber}(\theta)$:

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

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

Exemple 5.3. Modèle Normal–Normal. Avec $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$ et $\tilde{X} \mid \mu \sim \mathcal{N}(\mu, \sigma^2)$:

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

La variance prédictive $\sigma^2 + \tau_0^2$ additionne l'incertitude aléatoire et l'incertitude épistémique.

5.2 Distribution prédictive a posteriori

Définition 5.4 (Distribution prédictive a posteriori)**.** La **distribution prédictive a posteriori** de $\tilde{x}$ étant donné $x = (x_1, \ldots, x_n)$ est :

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

Décomposition de la variance prédictive

$$\mathrm{Var}[\tilde{X} \mid x] = \underbrace{\mathbb{E}_{\theta|x}[\mathrm{Var}[\tilde{X} \mid \theta]]}_{\text{incertitude aléatoire}} + \underbrace{\mathrm{Var}_{\theta|x}[\mathbb{E}[\tilde{X} \mid \theta]]}_{\text{incertitude épistémique}}.$$

Cette décomposition est la **loi de la variance totale**.

Théorème 5.5 (Prédictive a posteriori Normal–Normal)**.** *Si* $X_i \overset{iid}{\sim} \mathcal{N}(\mu, \sigma^2)$ *avec* $\mu \mid x \sim \mathcal{N}(\mu_n, \tau_n^2)$*, alors :*

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

Démonstration. Conditionnellement à μ, $\tilde{X} \sim \mathcal{N}(\mu, \sigma^2)$. En intégrant sur $\mu \mid x \sim \mathcal{N}(\mu_n, \tau_n^2)$, la somme de deux gaussiennes indépendantes donne :

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

□

Théorème 5.6 (Prédictive a posteriori Bernoulli–Bêta)**.** *Si* $X_i \overset{iid}{\sim} Ber(\theta)$ *avec* $\theta \mid x \sim Be(a_n, b_n)$*, alors :*

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

C'est la ***règle de succession de Laplace*** *quand* $a = b = 1$ *:* $\mathbb{P}(\tilde{X} = 1 \mid x) = (s+1)/(n+2)$.

Théorème 5.7 (Prédictive Poisson–Gamma)**.** *Si* $X_i \overset{iid}{\sim} Poi(\lambda)$ *avec* $\lambda \mid x \sim Ga(\alpha_n, \beta_n)$*, la prédictive est une* ***binomiale négative*** *:*

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

Démonstration.

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

□

Théorème 5.8 (Prédictive Normal–NIG)**.** *Si* $(\mu, \sigma^2) \mid x \sim NIG(\mu_n, \kappa_n, \alpha_n, \beta_n)$*, la prédictive a posteriori est une Student :*

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

5.3 Vérification prédictive a posteriori

Définition 5.9 (Posterior predictive check)**.** La **vérification prédictive a posteriori** consiste à simuler des répliques x^{rep} depuis la prédictive et à les comparer aux données observées pour évaluer l'adéquation du modèle.

Définition 5.10 (p-valeur prédictive bayésienne)**.** La **p-valeur prédictive** pour une statistique de test $T(x)$ est :

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

Des valeurs proches de 0 ou 1 suggèrent un mauvais ajustement.

Vérification prédictive a posteriori

1. Pour $t = 1, \ldots, T$:
   (a) Tirer $\theta^{(t)} \sim \pi(\theta \mid x_{\text{obs}})$ (depuis MCMC).
   (b) Tirer $x^{\text{rep},(t)} \sim f(x \mid \theta^{(t)})$.
   (c) Calculer $T^{(t)} = T(x^{\text{rep},(t)})$.
2. Estimer $p_B = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}(T^{(t)} \geq T(x_{\text{obs}}))$.
3. Tracer la distribution de $T^{(t)}$ et comparer à $T(x_{\text{obs}})$.

5.4 Intervalles prédictifs

Définition 5.11 (Intervalle prédictif)**.** Un **intervalle prédictif** à $100(1-\alpha)\%$ pour $\tilde{X}$ est un intervalle $[a, b]$ tel que :

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

Remarque 5.12. L'intervalle prédictif est toujours plus large que l'intervalle de crédibilité pour θ, car il intègre à la fois l'incertitude paramétrique et l'aléa de la nouvelle observation.

5.5 Prédiction bayésienne vs. plug-in

Danger du plug-in

L'approche *plug-in* remplace θ par une estimation ponctuelle $\hat{\theta}$ et prédit via $f(\tilde{x} \mid \hat{\theta})$. Elle sous-estime l'incertitude prédictive car elle ignore l'incertitude sur θ. La prédiction bayésienne est mieux calibrée.

5.6 Implémentation Python

Distribution prédictive a posteriori

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

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

# Modèle bayésien
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)
    # Prédictive
    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)

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

# Prédictive vs données
az.plot_ppc(ppc, observed_adata=trace, ax=axes[0])
axes[0].set_title("Vérification prédictive")

# Intervalle prédictif
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="Prédictive")
axes[1].axvline(ci_pred[0], color="red", linestyle="--",
                label=f"IC 95%: [{ci_pred[0]:.1f}, {ci_pred[1]:.1f}]")
axes[1].axvline(ci_pred[1], color="red", linestyle="--")
axes[1].legend()
axes[1].set_title("Distribution prédictive a posteriori")

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

p-valeur prédictive bayésienne

```python
# Statistique de test: écart-type des données
T_obs = data.std()

# Simuler des répliques
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)

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

# Visualisation
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("Écart-type")
plt.ylabel("Densité")
plt.legend()
plt.title(f"p-valeur prédictive = {p_B:.3f}")
plt.tight_layout()
plt.savefig("ppc_pvalue.pdf")
```

Prédictive Poisson–Gamma (binomiale négative)

```python
from scipy import stats

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

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

# Prédictive: 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="Prédictive NegBin")
plt.xlabel(r"$\tilde{x}$")
plt.ylabel("Probabilité")
plt.title("Prédictive Poisson-Gamma")
plt.legend()
plt.tight_layout()
plt.savefig("pred_negbin.pdf")
```

5.7 Exercices

Exercice 5.1. Dériver la distribution prédictive a posteriori pour le modèle Bernoulli–Bêta pour m nouvelles observations (pas une seule). Montrer qu'il s'agit d'une loi bêta-binomiale.

Exercice 5.2. Montrer que la variance prédictive est toujours supérieure ou égale à la variance conditionnelle $\mathrm{Var}[\tilde{X} \mid \theta]$ pour toute valeur de θ.

Exercice 5.3. Dériver la distribution prédictive pour le modèle Normal–NIG et montrer qu'il s'agit d'une Student.

Exercice 5.4. (Numérique) Simuler des données Poisson et comparer :

1. la prédictive bayésienne (binomiale négative),
2. la prédictive plug-in (Poisson avec $\hat{\lambda} = \bar{x}$).

Calculer la couverture réelle des intervalles prédictifs à 95% des deux approches sur 1000 répétitions.

Exercice 5.5. Effectuer une vérification prédictive a posteriori pour un modèle Poisson appliqué à des données sur-dispersées. Utiliser la variance comme statistique de test.

Chapitre 6

Modèles Hiérarchiques

Supposez que vous étudiez les performances scolaires dans 50 écoles. Chaque école a ses propres caractéristiques, mais elles partagent aussi un environnement commun. Faut-il modéliser chaque école indépendamment (beaucoup de paramètres, peu de données par école) ou toutes ensemble (en ignorant les différences) ? Les modèles hiérarchiques offrent un compromis élégant : les paramètres de chaque école sont tirés d'une distribution commune, dont les hyperparamètres sont eux-mêmes estimés. C'est le *shrinkage* bayésien : les estimations individuelles sont « tirées »vers la moyenne du groupe, d'autant plus qu'elles sont incertaines. Ce mécanisme, formalisé par Efron et Morris (1973), est l'un des arguments les plus convaincants en faveur de l'approche bayésienne.

Idée centrale

Les modèles hiérarchiques (ou multiniveaux) structurent les paramètres en plusieurs niveaux, permettant le partage d'information entre groupes. Le rétrécissement (*shrinkage*) qui en résulte améliore typiquement les estimations par rapport aux approches individuelles ou complètement groupées.

6.1 Motivation et structure

Définition 6.1 (Modèle hiérarchique)**.** Un **modèle hiérarchique** à deux niveaux a la structure :

$$\begin{aligned} \text{Niveau 1 (données) :} \quad & X_{ij} \mid \theta_j \sim f(x \mid \theta_j), \\ \text{Niveau 2 (paramètres) :} \quad & \theta_j \mid \phi \sim g(\theta \mid \phi), \\ \text{Niveau 3 (hyperparamètres) :} \quad & \phi \sim h(\phi), \end{aligned}$$

où $j = 1, \ldots, J$ indexe les groupes et $i = 1, \ldots, n_j$ les observations dans le groupe j.

Remarque 6.2. Les modèles hiérarchiques se situent entre deux extrêmes :

- **Modèle complet** (*complete pooling*) : $\theta_1 = \cdots = \theta_J = \theta$, un seul paramètre pour tous.
- **Modèle séparé** (*no pooling*) : chaque θ_j est estimé indépendamment.

Le modèle hiérarchique réalise un *partial pooling* optimal.

6.2 Modèle normal hiérarchique

Théorème 6.3 (Modèle normal hiérarchique)**.** *Considérons :*

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

La moyenne a posteriori de μ_j est :

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

où $B_j = \sigma^2/(n_j\tau^2+\sigma^2)$ est le facteur de rétrécissement et $\hat{\mu}$ est la moyenne globale estimée.

Rétrécissement hiérarchique

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

- Si $\tau^2 \gg \sigma^2/n_j$: peu de rétrécissement ($\hat{\mu}_j \approx \bar{x}_j$).
- Si $\tau^2 \ll \sigma^2/n_j$: rétrécissement fort ($\hat{\mu}_j \approx \mu$).

6.3 Exemple fondamental : les huit écoles

Exemple 6.4. Le célèbre exemple des « huit écoles »(Rubin, 1981) analyse l'effet d'un programme de coaching sur les scores SAT dans $J = 8$ écoles. Les données sont les effets estimés y_j et leurs erreurs-types σ_j :

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

Les σ_j sont connus (estimés avec grande précision).

6.4 Inférence dans les modèles hiérarchiques

6.4.1 Posterior conditionnel

Proposition 6.5 (Conditionnels complets)**.** Dans le modèle normal hiérarchique, les conditionnels complets sont :

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

Ces expressions permettent l'échantillonnage de Gibbs (chapitre 8).

6.4.2 Estimation de la variance inter-groupes

Difficulté de l'estimation de τ

L'estimation de la variance inter-groupes τ^2 est délicate, surtout quand J est petit. Le choix du prior sur τ a un impact important. Les recommandations incluent :

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

Éviter le prior conjugué $\tau^2 \sim \text{IG}(\varepsilon, \varepsilon)$ avec ε petit, qui est informatif près de zéro.

6.5 Modèle hiérarchique pour données binomiales

Exemple 6.6. Méta-analyse de taux de succès. Soit $X_j \sim \text{Bin}(n_j, \theta_j)$ avec $\theta_j \sim \text{Be}(\alpha, \beta)$ et des hyperpriors sur (α, β) :

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

Ce modèle partage l'information entre les J études tout en permettant l'hétérogénéité.

6.6 Paramétrisation et convergence MCMC

Définition 6.7 (Paramétrisation centrée vs. non centrée)**.**

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

La paramétrisation non centrée améliore souvent la convergence MCMC, surtout quand τ est petit ou les données peu informatives.

Remarque 6.8. La paramétrisation non centrée élimine la dépendance a posteriori entre τ et θ_j, réduisant la géométrie en « entonnoir »(*Neal's funnel*) qui cause des difficultés de mélange pour les échantillonneurs MCMC.

6.7 Modèles multiniveaux généraux

Définition 6.9 (Modèle linéaire mixte bayésien)**.**

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

où β sont les effets fixes, $b_j \sim \mathcal{N}(0, \Sigma_b)$ les effets aléatoires, et $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. Les priors sont :

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

6.8 Implémentation Python

Modèle des huit écoles avec PyMC

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

# Données des huit écoles
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)

# Paramétisation non centrée
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)

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

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

Visualisation du rétrécissement

```
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"École {j+1}" for j in range(J)]
x_pos = np.arange(J)

ax.scatter(x_pos, y, marker="o", s=100,
           label="Effet observé $y_j$", 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}")
# Flèches de rétrécissement
```

```
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("Effet du coaching")
ax.legend()
ax.set_title("Rétrécissement hiérarchique: Huit écoles")
plt.tight_layout()
plt.savefig("eight_schools_shrinkage.pdf")
```

Comparaison centrée vs. non centrée

```
# Paramétrisation centrée (pour comparaison)
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)

# Comparaison des ESS
ess_ncp = az.ess(trace, var_names=["theta"])
ess_cp = az.ess(trace_cp, var_names=["theta"])
print("ESS (non centrée):", ess_ncp["theta"].values.mean())
print("ESS (centrée):", 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 Exercices

Exercice 6.1. Dériver les conditionnels complets pour le modèle normal hiérarchique et écrire un échantillonneur de Gibbs.

Exercice 6.2. Montrer que le rétrécissement hiérarchique réduit l'erreur quadratique moyenne totale $\sum_j \mathbb{E}[(\hat{\mu}_j - \mu_j)^2]$ par rapport à l'estimation séparée $\hat{\mu}_j = \bar{x}_j$.

Exercice 6.3. Implémenter un modèle hiérarchique bêta-binomial pour des données de taux de clics (*click-through rates*) de $J = 10$ publicités.

Exercice 6.4. Comparer empiriquement les paramétrisations centrée et non centrée pour différentes valeurs de τ/σ (0.1, 1, 10). Mesurer le nombre de divergences et l'ESS.

Exercice 6.5. (Théorique) Montrer que le modèle hiérarchique normal admet la vraisemblance marginale :

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

En déduire une expression analytique pour la loi a posteriori marginale de (μ, τ) dans le modèle des huit écoles.

Chapitre 7

Méthodes MCMC — Metropolis-Hastings

En 1953, au laboratoire de Los Alamos, Nicholas Metropolis, Arianna et Marshall Rosenbluth, et Augusta et Edward Teller publient un article sur la simulation de molécules par une méthode de Monte Carlo particulière : au lieu d'échantillonner directement la distribution de Boltzmann (impossible en haute dimension), ils construisent une chaîne de Markov dont la loi stationnaire *est* la distribution cible. Vingt ans plus tard, W. Keith Hastings généralise l'algorithme en autorisant des noyaux de proposition asymétriques. Il faudra encore attendre les années 1990 pour que les statisticiens bayésiens — notamment Alan Gelfand et Adrian Smith — réalisent que cette méthode résout *leur* problème fondamental : échantillonner des lois a posteriori de forme complexe. L'algorithme de Metropolis–Hastings est ainsi devenu la clé de voûte de l'inférence bayésienne computationnelle, rendant calculables des modèles qui semblaient hors de portée.

Idée centrale

Quand le posterior n'a pas de forme analytique, on construit une chaîne de Markov dont la loi stationnaire est la loi a posteriori. L'algorithme de Metropolis–Hastings est la méthode MCMC fondamentale : il propose des transitions et les accepte ou rejette selon un ratio garantissant la convergence vers la cible.

7.1 Chaînes de Markov : rappels

Définition 7.1 (Chaîne de Markov)**.** Une suite $(\theta^{(t)})_{t\geq 0}$ est une **chaîne de Markov** si $\mathbb{P}(\theta^{(t+1)} \in A \mid \theta^{(0)}, \ldots, \theta^{(t)}) = \mathbb{P}(\theta^{(t+1)} \in A \mid \theta^{(t)})$ pour tout A mesurable.

Définition 7.2 (Réversibilité et équilibre détaillé)**.** Un noyau de transition $K(\theta, \theta')$ satisfait l'**équilibre détaillé** par rapport à π si :

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

Cela implique que π est la loi stationnaire de la chaîne.

Théorème 7.3 (Théorème ergodique)**.** *Si la chaîne est irréductible et apériodique avec loi stationnaire π, alors pour toute fonction g intégrable :*

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

7.2 Algorithme de Metropolis–Hastings

Algorithme de Metropolis–Hastings

1. Initialiser $\theta^{(0)}$.
2. Pour $t = 0, 1, 2, \ldots, T-1$:
   (a) Proposer $\theta^* \sim q(\theta^* \mid \theta^{(t)})$.
   (b) Calculer le ratio d'acceptation :
   $$\alpha(\theta^{(t)}, \theta^*) = \min\left(1,\ \frac{\pi(\theta^* \mid x)\, q(\theta^{(t)} \mid \theta^*)}{\pi(\theta^{(t)} \mid x)\, q(\theta^* \mid \theta^{(t)})}\right).$$
   (c) Tirer $u \sim \mathrm{Unif}(0,1)$.
   (d) Si $u \le \alpha$: $\theta^{(t+1)} = \theta^*$ (accepter). Sinon : $\theta^{(t+1)} = \theta^{(t)}$ (rejeter).

Théorème 7.4 (Validité de Metropolis–Hastings)**.** *Le noyau de Metropolis–Hastings satisfait l'équilibre détaillé par rapport à $\pi(\theta \mid x)$. Si la chaîne est irréductible et apériodique, elle converge vers $\pi(\theta \mid x)$.*

Démonstration. Le noyau de transition est :

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

Vérifions l'équilibre détaillé. Sans perte de généralité, supposons $\pi(\theta \mid x)\, q(\theta' \mid \theta) \le \pi(\theta' \mid x)\, q(\theta \mid \theta')$. Alors $\alpha(\theta, \theta') = 1$ et :

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

□

Remarque 7.5. On n'a besoin que du rapport $\pi(\theta^*)/\pi(\theta^{(t)})$, donc la constante de normalisation $m(x)$ s'annule. Il suffit de connaître le posterior à une constante près.

7.3 Cas particuliers

7.3.1 Algorithme de Metropolis (symétrique)

Définition 7.6 (Metropolis)**.** Quand la proposition est symétrique, $q(\theta^* \mid \theta) = q(\theta \mid \theta^*)$, le ratio se simplifie :

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

Exemple 7.7. Marche aléatoire gaussienne. $q(\theta^* \mid \theta^{(t)}) = \mathcal{N}(\theta^{(t)}, \sigma_q^2 I)$. La symétrie est immédiate. Le paramètre σ_q (pas de la marche) doit être calibré : un taux d'acceptation de $\approx 23\%$ est optimal en grande dimension.

7.3.2 Algorithme d'indépendance

Définition 7.8 (Metropolis–Hastings indépendant)**.** La proposition ne dépend pas de l'état courant : $q(\theta^* \mid \theta^{(t)}) = q(\theta^*)$. Le ratio est :

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

Condition de domination

L'algorithme d'indépendance fonctionne bien seulement si q domine $\pi(\cdot \mid x)$ au sens où les queues de q sont au moins aussi lourdes que celles de π.

7.4 Calibration et adaptation

Taux d'acceptation optimal

Dimension	Taux optimal
$d = 1$	$\approx 44\%$
$d \geq 2$ (cible gaussienne)	$\approx 23.4\%$

Règle pratique : ajuster σ_q pour obtenir un taux d'acceptation entre 20% et 50%.

Définition 7.9 (Metropolis adaptatif (AM))**.** L'algorithme AM ajuste la matrice de covariance de la proposition pendant le *burn-in* :

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

où $\hat{\Sigma}_t$ est la covariance empirique des échantillons $\theta^{(1)}, \ldots, \theta^{(t)}$.

7.5 Diagnostic de convergence

Définition 7.10 (Trace plot)**.** Le **trace plot** représente $\theta^{(t)}$ en fonction de t. On recherche un comportement de « bruit blanc »(bon mélange) sans tendance ni corrélation visible.

Définition 7.11 ($\hat{R}$ de Gelman–Rubin)**.** Le diagnostic $\hat{R}$ compare la variance intra-chaîne W et la variance inter-chaînes B pour M chaînes :

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

Convergence si $\hat{R} < 1.01$ (critère strict).

Définition 7.12 (Taille d'échantillon effective (ESS))**.** L'**ESS** mesure le nombre d'échantillons indépendants équivalents :

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

où ρ_k est l'autocorrélation d'ordre k. On recommande ESS > 400 par paramètre.

Check-list des diagnostics MCMC

1. Vérifier les trace plots (pas de tendance).
2. $\hat{R} < 1.01$ pour tous les paramètres.
3. ESS > 400 (ou > 100 par chaîne).
4. Pas de divergences (pour HMC/NUTS).
5. Cohérence entre chaînes multiples.

7.6 Implémentation Python

Metropolis–Hastings à la main

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

# Cible: posterior Beta-Binomiale
n_data, s = 50, 15  # 15 succès sur 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 avec marche aléatoire
T = 50000
sigma_q = 0.1
samples = np.zeros(T)
samples[0] = 0.5
accepted = 0

for t in range(1, T):
    # Proposition
    theta_star = samples[t-1] + sigma_q * np.random.randn()
    # Ratio (Metropolis car symétrique)
    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"Taux d'acceptation: {accepted/T:.2%}")
print(f"Moyenne post.: {samples_post.mean():.4f}")
print(f"Analytique: {(a_prior+s)/(a_prior+b_prior+n_data):.4f}")
```

Diagnostics MCMC

```
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 (2000 premières itérations)")
axes[0, 0].set_xlabel("Itération")
axes[0, 0].set_ylabel(r"$\theta$")

# Histogramme vs analytique
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="Analytique", color="red")
axes[0, 1].legend()
axes[0, 1].set_title("Posterior")

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

# 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("Moyenne cumulée")
axes[1, 1].set_xlabel("Itération")

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

MH multidimensionnel avec PyMC

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

```python
# Modèle de régression bayésienne
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)
    # Utiliser Metropolis explicitement
    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 Exercices

Exercice 7.1. Implémenter l'algorithme de Metropolis avec marche aléatoire pour échantillonner une distribution $t_3(0, 1)$. Étudier l'effet du pas σ_q sur le taux d'acceptation et l'ESS.

Exercice 7.2. Démontrer que l'algorithme de Metropolis–Hastings satisfait l'équilibre détaillé (preuve complète).

Exercice 7.3. Implémenter l'algorithme de Metropolis adaptatif (AM) et comparer avec Metropolis standard sur un posterior gaussien bidimensionnel avec forte corrélation ($\rho = 0.95$).

Exercice 7.4. Pour un modèle de mélange de deux gaussiennes, implémenter MH et observer le problème de multimodalité. Proposer des solutions (tempérage parallèle, propositions à grand saut).

Exercice 7.5. (Numérique) Comparer l'efficacité (ESS par seconde) de Metropolis–Hastings avec NUTS (PyMC par défaut) sur un modèle de régression logistique avec $p = 10$ prédicteurs.

Chapitre 8

Algorithme de Gibbs

En 1984, Stuart et Donald Geman publient un article sur la restauration d'images en introduisant un algorithme MCMC particulièrement élégant : l'*échantillonneur de Gibbs*. Au lieu de proposer des mouvements dans l'espace complet des paramètres (comme Metropolis–Hastings), l'algorithme met à jour chaque composante séparément, en tirant directement de sa *loi conditionnelle complète* — la distribution de cette composante conditionnellement à toutes les autres. Le génie du procédé est que le taux d'acceptation est toujours de 100 %, éliminant le besoin de calibrer une distribution de proposition. Alan Gelfand et Adrian Smith (1990) montrent ensuite que l'échantillonneur de Gibbs est applicable à une vaste classe de modèles bayésiens hiérarchiques, déclenchant la révolution MCMC en statistique.

Idée centrale

L'échantillonneur de Gibbs est un cas particulier de Metropolis–Hastings où chaque composante du paramètre est mise à jour en tirant directement depuis sa **loi conditionnelle complète**. Le taux d'acceptation est toujours 1, ce qui élimine le besoin de calibration de la proposition.

8.1 Lois conditionnelles complètes

Définition 8.1 (Loi conditionnelle complète)**.** Soit $\theta = (\theta_1, \ldots, \theta_d)$. La **loi conditionnelle complète** de θ_k est :

$$\pi(\theta_k \mid \theta_{-k}, x) \propto \pi(\theta_1, \ldots, \theta_d \mid x) \quad \text{en tant que fonction de } \theta_k,$$

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

Remarque 8.2. Pour identifier la conditionnelle complète, on fixe toutes les composantes sauf θ_k dans le posterior joint et on reconnaît la loi en θ_k.

8.2 Algorithme de Gibbs

Échantillonneur de Gibbs

1. Initialiser $\theta^{(0)} = (\theta_1^{(0)}, \dots, \theta_d^{(0)})$.
2. Pour $t = 0, 1, \dots, T-1$:
   (a) Tirer $\theta_1^{(t+1)} \sim \pi(\theta_1 \mid \theta_2^{(t)}, \dots, \theta_d^{(t)}, x)$.
   (b) Tirer $\theta_2^{(t+1)} \sim \pi(\theta_2 \mid \theta_1^{(t+1)}, \theta_3^{(t)}, \dots, \theta_d^{(t)}, x)$.
   $\vdots$
   (c) Tirer $\theta_d^{(t+1)} \sim \pi(\theta_d \mid \theta_1^{(t+1)}, \dots, \theta_{d-1}^{(t+1)}, x)$.

Théorème 8.3 (Gibbs comme cas particulier de MH)**.** *L'échantillonneur de Gibbs est un algorithme de Metropolis–Hastings où la proposition pour θ_k est $q_k(\theta_k^* \mid \theta_{-k}) = \pi(\theta_k^* \mid \theta_{-k}, x)$. Le ratio d'acceptation vaut toujours* 1.

Démonstration. Le ratio d'acceptation pour la mise à jour de θ_k est :

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

□

Théorème 8.4 (Convergence de l'échantillonneur de Gibbs)**.** *Sous des conditions de régularité (positivité du posterior sur tout l'espace), l'échantillonneur de Gibbs converge vers la loi cible $\pi(\theta \mid x)$.*

8.3 Exemples fondamentaux

8.3.1 Modèle normal bivarié

Exemple 8.5. Soit $(\theta_1, \theta_2) \sim \mathcal{N}_2(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ avec $\boldsymbol{\Sigma} = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$. Les conditionnelles complètes sont :

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

Forte corrélation

Quand ρ est proche de ± 1, le Gibbs sampler mélange lentement car les mises à jour par composante ne peuvent pas explorer efficacement la distribution allongée. On préfère alors HMC/NUTS ou une reparamétrisation.

8.3.2 Modèle normal avec moyenne et variance inconnues

Exemple 8.6. Soit $X_1, \dots, X_n \overset{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2)$ avec prior $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$ et $\sigma^2 \sim \text{IG}(\alpha_0, \beta_0)$. Les conditionnelles sont :

$$\mu \mid \sigma^2, x \sim \mathcal{N}\left(\frac{\mu_0/\tau_0^2 + n\bar{x}/\sigma^2}{1/\tau_0^2 + n/\sigma^2},\ \frac{1}{1/\tau_0^2 + n/\sigma^2}\right),$$

$$\sigma^2 \mid \mu, x \sim \text{IG}\left(\alpha_0 + \frac{n}{2},\ \beta_0 + \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2\right).$$

8.3.3 Modèle de mélange

Exemple 8.7. Mélange de K gaussiennes. Soient les données $x_1, \dots, x_n$ avec variables latentes $z_i \in \{1, \dots, K\}$ et paramètres $(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. Les conditionnelles sont :

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

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

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

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

8.4 Variantes de Gibbs

Définition 8.8 (Gibbs par blocs)**.** Au lieu de mettre à jour chaque composante séparément, le **Gibbs par blocs** met à jour des groupes de composantes simultanément. Cela améliore le mélange quand les composantes d'un bloc sont fortement corrélées.

Définition 8.9 (Collapsed Gibbs / Rao–Blackwellisation)**.** Le **collapsed Gibbs** marginalise analytiquement certains paramètres avant l'échantillonnage, réduisant la dimension et améliorant l'efficacité.

Théorème 8.10 (Rao–Blackwellisation)**.** *Si $g(\theta)$ est une fonction d'intérêt et qu'on peut calculer $\mathbb{E}[g(\theta_1) \mid \theta_2, x]$ analytiquement, alors l'estimateur Rao–Blackwellisé :*

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

a une variance inférieure ou égale à $\hat{g} = \frac{1}{T}\sum_t g(\theta_1^{(t)})$.

8.5 Augmentation de données

Définition 8.11 (Augmentation de données)**.** L'**augmentation de données** introduit des variables latentes z pour faciliter l'échantillonnage. On alterne :

1. Tirer $z \mid \theta, x$ (étape I).
2. Tirer $\theta \mid z, x$ (étape P).

Exemple 8.12. Modèle probit. Pour $Y_i \in \{0, 1\}$ avec $\mathbb{P}(Y_i = 1) = \Phi(X_i^\top \beta)$, on introduit $Z_i \sim \mathcal{N}(X_i^\top \beta, 1)$ et $Y_i = \mathbf{1}(Z_i > 0)$. Conditionnellement aux Z_i, β a un posterior gaussien. L'échantillonnage de Z_i se fait par troncature.

8.6 Implémentation Python

Gibbs pour le modèle normal

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

# Données
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()

# Hyperparamètres
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):
    # Tirer 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))

    # Tirer 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"(vrai: {mu_true})")
print(f"sigma: {np.sqrt(sigma2_samples[burn_in:]).mean():.3f} "
      f"(vrai: {sigma_true})")
```

Gibbs pour mélange de gaussiennes

```python
# Mélange de 2 gaussiennes
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

# Stockage
pi_samples = np.zeros((T, K))
mu_samples_mix = np.zeros((T, K))
sigma2_samples_mix = np.zeros((T, K))
z_samples = np.zeros((T, n), dtype=int)

# Initialisation
mu_samples_mix[0] = [-1, 6]
sigma2_samples_mix[0] = [1, 1]
pi_samples[0] = [0.5, 0.5]

for t in range(1, T):
    # Étape E: tirer z_i
    for i in range(n):
        probs = np.array([
            pi_samples[t-1, k] *
            stats.norm.pdf(data_mix[i],
                           mu_samples_mix[t-1, k],
                           np.sqrt(sigma2_samples_mix[t-1, k]))
            for k in range(K)])
        probs /= probs.sum()
        z_samples[t, i] = np.random.choice(K, p=probs)

    # Étape M: tirer paramètres
    for k in range(K):
        idx = z_samples[t] == k
        nk = idx.sum()
        # Proportions
        pi_samples[t, k] = nk  # sera normalisé
        if nk > 0:
            xk = data_mix[idx]
            # mu_k
            prec = 1/100 + nk/sigma2_samples_mix[t-1, k]
            m = (nk * xk.mean() / sigma2_samples_mix[t-1, k]) / prec
            mu_samples_mix[t, k] = np.random.normal(m, 1/np.sqrt(prec))
            # sigma^2_k
            a = 1 + nk/2
            b = 1 + 0.5 * np.sum((xk - mu_samples_mix[t, k])**2)
```

```
            sigma2_samples_mix[t, k] = 1/np.random.gamma(a, 1/b)
        else:
            mu_samples_mix[t, k] = mu_samples_mix[t-1, k]
            sigma2_samples_mix[t, k] = sigma2_samples_mix[t-1, k]

    # Dirichlet pour les proportions
    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_samples_mix[burn:].mean(axis=0))
print("pi estimates:", pi_samples[burn:].mean(axis=0))
```

Diagnostics et visualisation

```
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 de $\mu$")
axes[0, 1].plot(sigma2_samples[:3000])
axes[0, 1].set_title(r"Trace de $\sigma^2$")

# 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("Posterior joint")

# Marginal de 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 de $\sigma$")

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

8.7 Exercices

Exercice 8.1. Dériver les conditionnelles complètes pour le modèle de régression linéaire bayésien $y = X\beta + \varepsilon$ avec $\beta \sim \mathcal{N}(0, \sigma_\beta^2 I)$ et $\sigma^2 \sim \text{IG}(\alpha, \beta)$.

Exercice 8.2. Implémenter l'augmentation de données d'Albert et Chib (1993) pour le modèle probit et comparer avec PyMC.

Exercice 8.3. Pour le Gibbs bivarié gaussien, montrer analytiquement que l'autocorrélation d'ordre k du Gibbs sampler est ρ^{2k} et calculer l'ESS en fonction de ρ.

Exercice 8.4. Comparer Gibbs standard et Gibbs par blocs sur le modèle normal multivarié avec différentes structures de corrélation.

Exercice 8.5. (Avancé) Implémenter le collapsed Gibbs pour le modèle de mélange en marginalisant les proportions π_k. Comparer l'ESS avec le Gibbs standard.

Chapitre 9

Inférence Variationnelle

Les méthodes MCMC sont puissantes mais lentes : chaque échantillon dépend du précédent, la convergence est difficile à diagnostiquer, et le passage à l'échelle reste problématique. Dans les années 2000, une alternative émerge : l'*inférence variationnelle*, qui transforme le problème d'échantillonnage en un problème d'*optimisation*. L'idée, formalisée par Michael Jordan, Tommi Jaakkola et leurs collaborateurs, est de chercher dans une famille de distributions simples $\mathcal{Q}$ celle qui approxime le mieux la loi a posteriori, en minimisant la divergence de Kullback–Leibler. David Blei et ses étudiants démocratiseront l'approche en 2017 avec l'ADVI (Automatic Differentiation Variational Inference), rendant l'inférence variationnelle aussi accessible qu'un appel de fonction. Aujourd'hui, elle alimente les autoencodeurs variationnels (VAE) et les modèles génératifs profonds.

Idée centrale

L'inférence variationnelle transforme le problème d'intégration bayésienne (calcul de la loi a posteriori) en un problème d'*optimisation*. Au lieu d'échantillonner comme en MCMC, on cherche la distribution $q^*(\theta)$ dans une famille paramétrique qui approxime au mieux la vraie loi a posteriori $p(\theta \mid x)$.

9.1 Motivation et position du problème

Dans de nombreux modèles bayésiens, la loi a posteriori

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

est intraitable car la constante de normalisation $p(x) = \int p(x \mid \theta)\, \pi(\theta)\, d\theta$ ne se calcule pas en forme close. Les méthodes MCMC fournissent des solutions exactes asymptotiquement mais peuvent être coûteuses en grande dimension.

Remarque 9.1. L'inférence variationnelle est souvent préférée au MCMC lorsque :

- le jeu de données est très grand ($n > 10^5$),
- on a besoin d'une réponse rapide plutôt que d'une précision asymptotique,
- le modèle comporte de nombreuses variables latentes (LDA, VAE).

9.2 Divergence de Kullback–Leibler et ELBO

Définition 9.2 (Divergence KL)**.** Soient q et p deux densités sur Θ. La **divergence de Kullback–Leibler** de q vers p est :

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

Elle satisfait $\mathrm{KL}(q\|p) \geq 0$ avec égalité si et seulement si $q = p$ presque partout.

L'objectif variationnel est de minimiser $\mathrm{KL}(q\|p(\cdot \mid x))$ sur une famille $\mathcal{Q}$. En décomposant :

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

Définition 9.3 (ELBO — Evidence Lower Bound)**.** Le **ELBO** (*Evidence Lower Bound*) est défini par :

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

Puisque $\mathrm{KL} \geq 0$, on a $\mathcal{L}(q) \leq \ln p(x)$ pour tout q.

Théorème 9.4 (Équivalence des objectifs)**.** *Maximiser le ELBO $\mathcal{L}(q)$ par rapport à $q \in \mathcal{Q}$ est équivalent à minimiser* $\mathrm{KL}(q\|p(\cdot \mid x))$.

Attention

La divergence KL n'est pas symétrique : $\mathrm{KL}(q\|p) \neq \mathrm{KL}(p\|q)$. L'inférence variationnelle utilise $\mathrm{KL}(q\|p)$ (direction « exclusive », *reverse KL*, $\to q$ tend à être plus concentrée que p, cherchant un mode). L'EP utilise $\mathrm{KL}(p\|q)$ (direction « inclusive », *forward KL*, $\to q$ couvre tout le support de p).

9.3 Approximation en champ moyen

Définition 9.5 (Famille champ moyen)**.** On partitionne $\theta = (\theta_1, \ldots, \theta_K)$ et on pose :

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

Chaque facteur q_k est libre (non paramétrique au sein de la factorisation).

Théorème 9.6 (Mise à jour optimale en champ moyen)**.** *En fixant tous les q_j $(j \neq k)$, le facteur optimal q_k^* satisfait :*

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

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

Remarque 9.7. Pour les modèles à famille exponentielle conjuguée, chaque q_k^* appartient à la même famille que le prior conditionnel complet, ce qui rend les mises à jour analytiques.

9.4 CAVI — Coordinate Ascent Variational Inference

Définition 9.8 (Algorithme CAVI)**.** L'algorithme **CAVI** itère cycliquement la mise à jour du théorème 9.6 pour $k = 1, \dots, K$ et surveille la convergence via le ELBO.

Proposition 9.9 (Convergence de CAVI)**.** Chaque mise à jour de CAVI augmente (ou maintient) le ELBO. L'algorithme converge vers un maximum local de $\mathcal{L}(q)$.

Exemple 9.10. Considérons un modèle gaussien univarié : $x_i \mid \mu, \tau \sim \mathcal{N}(\mu, \tau^{-1})$, avec $\mu \sim \mathcal{N}(0, \sigma_0^2)$ et $\tau \sim \text{Gamma}(a_0, b_0)$. L'approximation champ moyen $q(\mu, \tau) = q_\mu(\mu)\, q_\tau(\tau)$ donne :

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

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

Les espérances croisées sont itérées jusqu'à convergence.

9.5 Inférence variationnelle stochastique (SVI)

CAVI nécessite de parcourir toutes les données à chaque itération. Pour les grands jeux de données, on utilise la **SVI** (*Stochastic Variational Inference*).

Définition 9.11 (SVI)**.** On distingue les variables *globales* β et *locales* z_i. À chaque itération :

1. Tirer un mini-batch $S \subset \{1, \dots, n\}$.
2. Mettre à jour les variables locales $q(z_i)$ pour $i \in S$.
3. Calculer le gradient naturel du ELBO par rapport aux paramètres globaux et effectuer un pas de descente.

Proposition 9.12 (Gradient naturel)**.** Pour un modèle à famille exponentielle, le gradient naturel du ELBO par rapport aux paramètres naturels λ de $q(\beta)$ est :

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

où t est la statistique suffisante et S le mini-batch.

```
# Pseudo-code SVI
for epoch in range(max_epochs):
    S = random_minibatch(data, batch_size)
    # Mise a jour locale
    for i in S:
        q_z[i] = update_local(q_beta, x[i])
    # Gradient naturel pour les parametres globaux
    grad = lambda_prior + (n / len(S)) * sufficient_stats(S) - lam
    lam = lam + rho * grad  # rho = pas de Robbins-Monro
```

9.6 Inférence variationnelle par réparametrisation

Définition 9.13 (Astuce de réparametrisation)**.** Si $\theta \sim q_\phi(\theta)$ peut s'écrire $\theta = g(\phi, \epsilon)$ avec $\epsilon \sim p(\epsilon)$ indépendant de ϕ, alors :

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

ce qui permet d'utiliser la rétropropagation.

Exemple 9.14. Pour $q_\phi(\theta) = \mathcal{N}(\mu, \sigma^2)$, on pose $\theta = \mu + \sigma\epsilon$ avec $\epsilon \sim \mathcal{N}(0, 1)$. C'est la base des **auto-encodeurs variationnels** (VAE).

9.7 Comparaison avec MCMC

Critère	**MCMC**	**VI**
Nature	Échantillonnage	Optimisation
Convergence	Exacte (asymptotique)	Approximative
Vitesse	Lente pour n grand	Rapide, scalable
Diagnostic	Traces, $\hat{R}$	ELBO
Multimodalité	Possible (HMC, …)	Difficult
Incertitude	Fidèle	Sous-estimée (KL forward)

Remarque 9.15. En pratique, il est recommandé d'utiliser le VI pour l'exploration de modèles et le MCMC pour l'inférence finale lorsque la précision est critique.

9.8 Extensions

9.8.1 Normalizing flows

On peut enrichir la famille variationnelle $\mathcal{Q}$ en composant des transformations inversibles :

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

La densité résultante s'obtient par la formule du changement de variable :

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

9.8.2 Inférence variationnelle amortie

Dans les modèles à variables latentes z_i par observation, on paramètre $q_\phi(z_i \mid x_i)$ par un réseau de neurones (*encoder*). C'est le principe du VAE.

9.9 Formules clés

Formules clés

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

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

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

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

9.10 Exercices

Exercice 9.1. Montrer que $\mathrm{KL}(q\|p) \geq 0$ en utilisant l'inégalité de Jensen. En déduire que le ELBO est bien une borne inférieure de $\ln p(x)$.

Exercice 9.2. Considérer le modèle : $x_i \mid \mu \sim \mathcal{N}(\mu, 1)$, $i = 1, \ldots, n$, avec $\mu \sim \mathcal{N}(0, \sigma_0^2)$. Calculer le ELBO pour $q(\mu) = \mathcal{N}(m, s^2)$ et le maximiser analytiquement en m et s^2. Vérifier que l'on retrouve la loi a posteriori exacte.

Exercice 9.3. Implémenter l'algorithme CAVI pour le modèle gaussien avec précision inconnue de la section précédente. Tracer l'évolution du ELBO au fil des itérations.

Exercice 9.4. Comparer expérimentalement (en Python) l'approximation variationnelle champ moyen et l'échantillonneur de Gibbs pour un mélange de deux gaussiennes. Discuter les différences observées sur l'estimation de l'incertitude.

Exercice 9.5 (Calcul du ELBO pour un modèle gaussien)**.** On considère le modèle : $x_1, \ldots, x_n \mid \mu, \sigma^2 \sim \mathcal{N}(\mu, \sigma^2)$ avec σ^2 connu, et le prior $\mu \sim \mathcal{N}(\mu_0, \tau_0^2)$. Soit $q(\mu) = \mathcal{N}(m, s^2)$ la famille variationnelle.

1. Écrire explicitement le ELBO $\mathcal{L}(m, s^2) = \mathbb{E}_q[\ln p(x_{1:n}, \mu)] - \mathbb{E}_q[\ln q(\mu)]$ en fonction de m, s^2, μ_0, τ_0^2, σ^2, $\bar{x}$ et n.

2. Maximiser $\mathcal{L}$ par rapport à m et s^2 et montrer que l'on retrouve la loi a posteriori exacte $\mathcal{N}(m^*, (s^*)^2)$.

3. Calculer le ELBO au point optimal et vérifier que $\mathcal{L}(m^*, (s^*)^2) = \ln p(x_{1:n})$ (la KL s'annule).

Exercice 9.6 (Mise à jour champ moyen pour un modèle linéaire)**.** Soit le modèle de régression bayésienne : $y \mid X, \beta, \tau \sim \mathcal{N}(X\beta, \tau^{-1} I_n)$, $\beta \sim \mathcal{N}(0, \alpha^{-1} I_p)$, $\tau \sim \mathrm{Gamma}(a_0, b_0)$, avec α fixé. On pose $q(\beta, \tau) = q_\beta(\beta)\, q_\tau(\tau)$.

1. Dériver la mise à jour optimale $q_\beta^*(\beta)$ en calculant $\mathbb{E}_{q_\tau}[\ln p(y, \beta, \tau)]$ et montrer que $q_\beta^*(\beta) = \mathcal{N}(\mu_\beta, \Sigma_\beta)$ avec $\Sigma_\beta = (\mathbb{E}[\tau]\, X^\top X + \alpha I_p)^{-1}$ et $\mu_\beta = \mathbb{E}[\tau]\, \Sigma_\beta\, X^\top y$.

2. Dériver $q_\tau^*(\tau) = \mathrm{Gamma}(a_n, b_n)$ en calculant $\mathbb{E}_{q_\beta}[\ln p(y, \beta, \tau)]$. Exprimer a_n et b_n.

3. Implémenter l'algorithme CAVI résultant et vérifier la convergence du ELBO sur des données simulées.

Exercice 9.7 (Comparaison de métriques de qualité pour le VI)**.** Soit $p(\theta \mid x)$ une loi a posteriori bimodale (mélange de deux gaussiennes en dimension 1) et $q(\theta) = \mathcal{N}(m, s^2)$.

1. Calculer numériquement $\mathrm{KL}(q\|p)$ (reverse KL) et $\mathrm{KL}(p\|q)$ (forward KL) pour différentes valeurs de m et s. Illustrer graphiquement les minimiseurs respectifs.

2. Montrer que le minimiseur de $\mathrm{KL}(q\|p)$ se concentre sur un seul mode (« mode-seeking »), tandis que le minimiseur de $\mathrm{KL}(p\|q)$ couvre les deux modes (« mass-covering »).

3. Calculer également le *chi-squared divergence* $\chi^2(q\|p) = \int \frac{(q-p)^2}{p}\, d\theta$ et la distance de Wasserstein $W_2(q, p)$. Comparer les approximations variationnelles obtenues par minimisation de ces quatre métriques.

Chapitre 10

Modèles Bayésiens Non-Paramétriques

En statistique paramétrique, on choisit à l'avance le nombre de paramètres : un modèle de mélange à K composantes, une régression polynomiale de degré p. Mais comment choisir K ou p ? La statistique bayésienne non paramétrique contourne le problème en plaçant des lois a priori sur des espaces de *dimension infinie* : le processus de Dirichlet (Ferguson, 1973) définit une loi sur l'espace des distributions de probabilité ; le processus gaussien définit une loi sur l'espace des fonctions. Le modèle adapte alors automatiquement sa complexité aux données : le nombre de clusters croît avec la taille de l'échantillon, la souplesse de la régression s'ajuste. Cette flexibilité a révolutionné le clustering, la modélisation de données de survie et l'analyse de textes.

Idée centrale

En statistique bayésienne non paramétrique, on place des lois a priori sur des espaces de dimension infinie (distributions, fonctions, partitions). Le modèle adapte sa complexité aux données : le nombre de composantes ou la souplesse de la fonction ajustée croît avec n.

10.1 Motivation

Les modèles paramétriques fixent le nombre de paramètres à l'avance : un mélange de K gaussiennes, une régression d'ordre p, etc. Or, le choix de K ou p est souvent arbitraire.

Remarque 10.1. Le terme « non paramétrique »est trompeur : il y a bien des paramètres, mais en nombre *infini* (ou croissant avec les données).

10.2 Le processus de Dirichlet

Définition 10.2 (Processus de Dirichlet)**.** Soit $\alpha > 0$ un paramètre de concentration et G_0 une mesure de probabilité de base sur $(\Theta, \mathcal{B})$. On dit que $G \sim \mathrm{DP}(\alpha, G_0)$ si, pour toute partition mesurable $(A_1, \ldots, A_K)$ de Θ :

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

Proposition 10.3 (Propriétés du DP)**.** Soit $G \sim \mathrm{DP}(\alpha, G_0)$. Alors :

1. $\mathbb{E}[G(A)] = G_0(A)$ pour tout A.
2. $\mathrm{Var}(G(A)) = \dfrac{G_0(A)(1 - G_0(A))}{\alpha + 1}$.
3. G est presque sûrement discrète (mesure atomique).

Attention

Le DP génère des distributions discrètes même si G_0 est continue. C'est cette discrétude qui induit le regroupement (*clustering*).

10.3 Construction par brisure de bâton

Théorème 10.4 (Sethuraman, 1994)**.** *Si $G \sim \mathrm{DP}(\alpha, G_0)$, alors G admet la représentation :*

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

où $\theta_k^* \overset{iid}{\sim} G_0$ et les poids sont construits par brisure de bâton :

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

Remarque 10.5. Les poids π_k décroissent stochastiquement : les premiers atomes reçoivent l'essentiel de la masse. En pratique, on tronque à K termes pour l'inférence.

```python
import numpy as np

def stick_breaking(alpha, K):
    """Genere K poids par brisure de baton."""
    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 Le processus du restaurant chinois

Définition 10.6 (Processus du restaurant chinois (CRP))**.** Soit $(\theta_1, \theta_2, \ldots)$ un échantillon du DP marginal. La loi conditionnelle prédictive est :

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

où $\theta_1^*, \ldots, \theta_{K_n}^*$ sont les valeurs distinctes et n_k leurs effectifs.

Métaphore du restaurant

Des clients arrivent un par un dans un restaurant avec des tables infinies. Le client $n+1$ s'assoit à la table k (déjà occupée par n_k clients) avec probabilité $n_k/(n+\alpha)$, ou ouvre une nouvelle table avec probabilité $\alpha/(n+\alpha)$.

Proposition 10.7 (Nombre attendu de clusters)**.** Pour n observations tirées d'un $\mathrm{DP}(\alpha, G_0)$, le nombre attendu de clusters distincts est :

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

10.5 Mélange par processus de Dirichlet (DP-GMM)

Définition 10.8 (Modèle DP-GMM)**.** Le **mélange gaussien infini** (DP-GMM) est défini par :

$$G \sim \mathrm{DP}(\alpha, G_0), \tag{10.1}$$
$$(\mu_i, \Sigma_i) \sim G, \quad i = 1, \ldots, n, \tag{10.2}$$
$$x_i \mid \mu_i, \Sigma_i \sim \mathcal{N}(\mu_i, \Sigma_i). \tag{10.3}$$

Le nombre de composantes est déterminé par les données.

Exemple 10.9. Avec $G_0 = \mathrm{NIW}(\mu_0, \kappa_0, \nu_0, \Psi_0)$ (Normal-Inverse-Wishart), l'échantillonneur de Gibbs du CRP itère :

1. Pour chaque i, réassigner z_i selon $p(z_i = k \mid \ldots) \propto n_{-i,k}\, \mathcal{N}(x_i \mid \mu_k, \Sigma_k)$ ou créer un nouveau cluster.
2. Pour chaque cluster k, mettre à jour (μ_k, Σ_k) via le posterior NIW.

```python
from sklearn.mixture import BayesianGaussianMixture

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

10.6 Processus gaussiens comme priors sur les fonctions

Définition 10.10 (Processus gaussien)**.** Un **processus gaussien** (GP) est une collection de variables aléatoires $\{f(x) : x \in \mathcal{X}\}$ telle que toute sous-collection finie suit une loi

gaussienne multivariée :

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

où $m_i = m(x_i)$ et $K_{ij} = k(x_i, x_j)$.

Théorème 10.11 (Posterior du GP pour la régression)**.** *Soit* $\mathbf{y} = f(\mathbf{X}) + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$. *La loi a posteriori de* $f_* = f(x_*)$ *est :*

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

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

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

Remarque 10.12. Le choix du noyau k encode les hypothèses sur la régularité de f : RBF pour des fonctions lisses, Matérn pour un contrôle fin de la différentiabilité, périodique pour des données cycliques.

10.7 Autres processus non paramétriques

Définition 10.13 (Processus du buffet indien (IBP))**.** Le **IBP** est un prior sur des matrices binaires Z de taille $n \times \infty$ utilisé pour la sélection de caractéristiques. Chaque client (observation) choisit des plats (caractéristiques) existants avec probabilité m_k/n et essaie Poisson(α/n) nouveaux plats.

Définition 10.14 (Processus de Pitman–Yor)**.** Le **processus de Pitman–Yor** $\mathrm{PY}(d, \alpha, G_0)$ généralise le DP avec un paramètre de discount $d \in [0, 1)$. La loi prédictive est :

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

Pour $d = 0$, on retrouve le DP.

10.8 Formules clés

Formules clés

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

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

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

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

10.9 Exercices

Exercice 10.1. Vérifier que les poids de la brisure de bâton somment à 1 presque sûrement. *Indication :* calculer $\mathbb{E}\big[\sum_{k=1}^{K} \pi_k\big]$ et passer à la limite.

Exercice 10.2. Simuler le processus du restaurant chinois pour $\alpha \in \{0.1, 1, 10, 100\}$ et $n = 500$. Tracer le nombre de clusters K_n en fonction de α. Comparer avec la formule théorique $\mathbb{E}[K_n] \approx \alpha \ln(1 + n/\alpha)$.

Exercice 10.3. Implémenter un échantillonneur de Gibbs pour le DP-GMM sur des données simulées en dimension 2 (trois clusters gaussiens). Visualiser les affectations au fil des itérations.

Exercice 10.4. Considérer un GP avec noyau RBF $k(x, x') = \sigma_f^2 \exp(-\|x - x'\|^2 / (2\ell^2))$. Montrer que la loi prédictive du GP converge vers un interpolateur exact lorsque $\sigma^2 \to 0$. Illustrer graphiquement avec des données simulées.

Exercice 10.5. Montrer que le processus de Pitman–Yor engendre un nombre de clusters $K_n = O(n^d)$, contre $K_n = O(\ln n)$ pour le DP. Quelle implication pour la modélisation de données textuelles suivant la loi de Zipf ?

Chapitre 11

Applications

Idée centrale

La statistique bayésienne est omniprésente dans les applications modernes : essais cliniques, tests A/B, optimisation automatique d'hyperparamètres, réseaux de neurones bayésiens, et de nombreux domaines scientifiques. Ce chapitre illustre la puissance du paradigme bayésien sur des problèmes concrets.

11.1 Essais cliniques bayésiens

Définition 11.1 (Essai clinique adaptatif bayésien)**.** Un **essai clinique adaptatif bayésien** est un essai où la conception (taille d'échantillon, allocation des patients, critères d'arrêt) est modifiée en cours d'étude en fonction de la loi a posteriori des paramètres d'efficacité.

Exemple 11.2. Soit θ_T la probabilité de réponse au traitement et θ_C celle du contrôle. On modélise :

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

Après observation de s_T succès sur n_T patients traités et s_C sur n_C contrôles :

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

On arrête l'essai pour efficacité si $\mathbb{P}(\theta_T > \theta_C \mid \text{data}) > 0{,}99$, ou pour futilité si $\mathbb{P}(\theta_T > \theta_C + \delta \mid \text{data}) < 0{,}05$.

Remarque 11.3. Les essais bayésiens adaptatifs sont approuvés par la FDA depuis les années 2010. Ils réduisent la taille d'échantillon nécessaire de 20 à 40 % en moyenne par rapport aux plans fixes.

```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) par 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 Tests A/B bayésiens

Définition 11.4 (Test A/B bayésien)**.** Un **test A/B bayésien** compare deux variantes (A et B) d'un produit en maintenant une distribution a posteriori sur le taux de conversion de chacune. La décision est prise lorsque la probabilité qu'une variante est meilleure dépasse un seuil fixé.

Proposition 11.5 (Perte attendue)**.** La **perte attendue** en choisissant B est :

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

On choisit B si $\mathcal{L}_B < \epsilon$ (seuil de perte acceptable). Ce critère contrôle directement le coût d'une mauvaise décision.

Attention

Contrairement au test fréquentiste (p-value), le test A/B bayésien permet de « peek »(regarder les données pendant la collecte) sans inflation du taux d'erreur, car la loi a posteriori est cohérente à tout instant.

11.3 Optimisation bayésienne

Définition 11.6 (Optimisation bayésienne)**.** L'**optimisation bayésienne** cherche le minimum global d'une fonction coûteuse $f : \mathcal{X} \to \mathbb{R}$ en construisant un modèle probabiliste (généralement un GP) de f et en sélectionnant les points d'évaluation via une **fonction d'acquisition**.

Théorème 11.7 (Posterior du GP comme surrogate)**.** *Soit* $f \sim \mathcal{GP}(0, k)$ *et* $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$ *avec* $y_i = f(x_i) + \epsilon_i$*. La prédiction en* x *est :*

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

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

Définition 11.8 (Fonctions d'acquisition)**.** Les fonctions d'acquisition classiques sont :

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

Exemple 11.9. Optimisation des hyperparamètres d'un réseau de neurones :

```
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 Réseaux de neurones bayésiens

Définition 11.10 (Réseau de neurones bayésien (BNN))**.** Un **BNN** place une loi a priori $p(\mathbf{w})$ sur les poids $\mathbf{w}$ du réseau et calcule la loi a posteriori $p(\mathbf{w} \mid \mathcal{D})$. La prédiction est :

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

Remarque 11.11. L'intégrale ci-dessus est intraitable pour les réseaux profonds. Trois approches pratiques :

1. **MC Dropout** : utiliser le dropout à l'inférence comme approximation variationnelle (Gal et Ghahramani, 2016).
2. **Bayes by Backprop** : inférence variationnelle avec réparametrisation sur chaque poids.
3. **Ensembles profonds** : entraîner M réseaux et moyenner (approximation fréquentiste de l'incertitude).

Proposition 11.12 (Incertitude épistémique vs. aléatoire)**.** La variance prédictive du BNN se décompose en :

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

L'incertitude épistémique diminue avec les données, l'aléatoire non.

11.5 Applications en astronomie

Exemple 11.13. En cosmologie, l'estimation des paramètres du modèle ΛCDM (constante de Hubble H_0, densité de matière Ω_m, etc.) repose sur l'inférence bayésienne appliquée aux données du fond diffus cosmologique (CMB). Le logiciel `CosmoMC` utilise MCMC pour explorer la loi a posteriori en dimension ~ 30.

Exemple 11.14. La détection d'exoplanètes par la méthode des vitesses radiales modélise le signal comme :

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

où le nombre de planètes K est inconnu. La comparaison de modèles bayésienne (facteur de Bayes) permet de déterminer K.

11.6 Applications en écologie

Exemple 11.15. Les modèles **capture-recapture** bayésiens estiment la taille N d'une population animale. Avec un prior $N \sim \text{Poisson}(\lambda)$ et des données de recapture, on obtient un posterior sur N qui quantifie naturellement l'incertitude, essentielle pour la gestion de la conservation.

Définition 11.16 (Modèle hiérarchique en écologie)**.** Les **modèles hiérarchiques bayésiens** sont standard en écologie pour modéliser :

- la variabilité entre sites (effets aléatoires spatiaux),
- l'imperfection de la détection (modèles d'occupation),
- les dynamiques temporelles (modèles espace-état).

Les paramètres de chaque site j partagent un hyperprior commun : $\theta_j \sim p(\theta_j \mid \psi)$, $\psi \sim p(\psi)$.

11.7 Formules clés

Formules clés

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

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

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

11.8 Exercices

Exercice 11.1. Implémenter un test A/B bayésien avec prior Beta(1,1) pour comparer les taux de clic de deux pages web. Simuler des données et tracer l'évolution de $\mathbb{P}(\theta_A > \theta_B \mid \text{data})$ au fil des observations.

Exercice 11.2. Réaliser une optimisation bayésienne pour trouver le minimum de la fonction de Branin $f(x_1, x_2)$ en utilisant un GP avec noyau Matérn. Comparer avec une recherche aléatoire en termes de nombre d'évaluations.

Exercice 11.3. Implémenter MC Dropout sur un réseau de neurones à deux couches cachées pour un problème de régression. Tracer la prédiction moyenne et l'intervalle de confiance à 95 %. Observer le comportement de l'incertitude loin des données d'entraînement.

Exercice 11.4. Concevoir un essai clinique adaptatif bayésien avec les règles suivantes : arrêt pour efficacité si $\mathbb{P}(\theta_T - \theta_C > 0.05 \mid \text{data}) > 0.95$, arrêt pour futilité si $\mathbb{P}(\theta_T > \theta_C \mid \text{data}) < 0.2$. Simuler 1000 essais sous H_0 et sous H_1 et estimer les caractéristiques opératoires (puissance, taux d'erreur de type I, taille d'échantillon moyenne).

Exercice 11.5. Dans un problème de détection d'exoplanètes, on compare les modèles M_0 (bruit pur) et M_1 (une planète). Calculer le facteur de Bayes BF_{10} par intégration de Monte Carlo sur des données simulées. Discuter l'échelle de Jeffreys pour l'interprétation.

Bibliographie

[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.