Maximum Likelihood Estimation (MLE)

A notebook covering the theory, mathematics, and Python implementation of MLE.

1. Brief Idea

Maximum Likelihood Estimation finds parameter values that maximize the probability (likelihood) of observing the given data under a chosen statistical model. It's a general framework applicable to many distributions and models. For simple cases (e.g., Gaussian), closed-form solutions exist; for complex models, one resorts to iterative optimization. MLE underpins numerous inferential and learning algorithms.

2. Context and Backstory

Fisher introduced MLE in the 1920s, revolutionizing statistical estimation by providing consistent and asymptotically efficient estimators. It's used in fields from econometrics to machine learning (e.g., estimating mixture model parameters). Many advanced techniques (EM algorithm, generalized linear models) build directly on MLE principles.

3. Mathematical Derivations and Formulae

Given i.i.d. data $\{x^{(i)}\}$ and model $p(x;\theta)$, define the likelihood

$$L(\theta) = \prod_{i=1}^{N} p(x^{(i)};\theta), \quad \ell(\theta) = \log L(\theta) = \sum_{i=1}^{N} \log p(x^{(i)};\theta).$$

Solve

$$\frac{\partial \ell(\theta)}{\partial \theta} = 0$$

for θ. **Example (Gaussian)**: $x^{(i)} \sim \mathcal{N}(\mu, \sigma^2)$. Then

$$\hat{\mu} = \frac{1}{N}\sum_i x^{(i)}, \quad \hat{\sigma}^2 = \frac{1}{N}\sum_i \left(x^{(i)} - \hat{\mu}\right)^2.$$

4. Diagrams and Visual Aids

Figure: Hypothetical likelihood surface over two parameters.

(Insert a plot or image here if desired.)

In [ ]:
```
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt

# Sample data
data = np.random.normal(loc=5, scale=2, size=500)

# Log-Likelihood as function of mu
```

```
mus = np.linspace(4,6,200)
ll = [np.sum(norm.logpdf(data, loc=mu, scale=2)) for mu in mus]

plt.plot(mus, ll)
plt.xlabel("μ"); plt.ylabel("Log-Likelihood"); plt.title("MLE of μ")
plt.show()

# MLE estimate
mu_hat = data.mean()
print("Estimated μ:", mu_hat)
```

6. Frequently Asked Questions (FAQ)

1. **Why maximize log-likelihood instead of likelihood?** Log transforms products into sums for numerical stability.
2. **Are MLEs biased?** They can be biased in small samples—bias often vanishes asymptotically.
3. **When is MLE intractable?** For models with hidden variables or complex integrals—use EM or Monte Carlo methods.
4. **Relation to MAP?** MLE ignores priors; MAP incorporates them via Bayes' rule.

7. Higher-Order Thinking Questions (HOTS)

- Derive MLE for parameters of an exponential distribution.
- Discuss the Cramér–Rao lower bound and MLE efficiency.
- How would you implement MLE for a mixture of Gaussians (outline EM)?

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

# Generate data
np.random.seed(0)
data = np.random.normal(loc=5, scale=2, size=500)

# MLE for Gaussian: mean and variance
mu_hat = np.mean(data)
sigma2_hat = np.mean((data - mu_hat) ** 2)
print('MLE mean:', mu_hat)
print('MLE variance:', sigma2_hat)

# Plot histogram and fitted Gaussian
plt.hist(data, bins=30, density=True, alpha=0.5, label='Data')
x = np.linspace(data.min(), data.max(), 200)
pdf = 1/np.sqrt(2 * np.pi * sigma2_hat) * np.exp(-0.5 * (x - mu_hat) ** 2 / sigma2_hat)
plt.plot(x, pdf, color='red', label='Fitted Gaussian (MLE)')
plt.xlabel('x')
plt.ylabel('Density')
plt.title('MLE for Gaussian Parameters (from scratch)')
plt.legend()
plt.show()
```