Maximum A Posteriori Estimation (MAP)

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

1. Brief Idea

Maximum A Posteriori (MAP) estimation finds parameter values that maximize the posterior probability given observed data and a prior belief. Unlike MLE, which only considers the likelihood, MAP incorporates prior information, making it especially useful when data is scarce or prior knowledge is strong. MAP is a Bayesian generalization of MLE.

2. Context and Backstory

MAP estimation is rooted in Bayesian statistics, where parameters are treated as random variables with prior distributions. It is widely used in machine learning, signal processing, and natural language processing, especially when prior knowledge or regularization is important. MAP bridges the gap between pure likelihood-based inference (MLE) and full Bayesian inference.

3. Mathematical Derivations and Formulae

Given data $\{x^{(i)}\}$, model $p(x;\theta)$, and prior $p(\theta)$, the posterior is:

$$p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$$

MAP estimate maximizes the posterior:

$$\hat{\theta}_{MAP} = \arg\max_{\theta}\ p(\theta|x) = \arg\max_{\theta}\ \log p(x|\theta) + \log p(\theta)$$

Example (Gaussian with Gaussian prior): If $x^{(i)} \sim \mathcal{N}(\mu, \sigma^2)$ and $\mu \sim \mathcal{N}(\mu_0, \tau^2)$, then

$$\hat{\mu}_{MAP} = \frac{\tau^2}{N\sigma^2 + \tau^2}\mu_0 + \frac{N\sigma^2}{N\sigma^2 + \tau^2}\bar{x}$$

where $\bar{x}$ is the sample mean.

4. Diagrams and Visual Aids

Figure: Prior, likelihood, and posterior distributions for a parameter.

(Insert a plot or image here if desired.)

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

# Prior: mu ~ N(mu0, tau^2)
mu0 = 0
sigma = 2
N = 100
tau = 1
```

```python
# Generate data
np.random.seed(0)
data = np.random.normal(loc=2, scale=sigma, size=N)

# Posterior parameters
sample_mean = data.mean()
mu_map = (tau**2/(N*sigma**2+tau**2))*mu0 + (N*sigma**2/(N*sigma**2+tau**2))*sample_mean

# Plot prior, likelihood, posterior
mus = np.linspace(-1, 4, 200)
prior = norm.pdf(mus, mu0, tau)
likelihood = norm.pdf(mus, sample_mean, sigma/np.sqrt(N))
posterior = norm.pdf(mus, mu_map, np.sqrt(1/(1/tau**2 + N/sigma**2)))

plt.plot(mus, prior, label='Prior')
plt.plot(mus, likelihood, label='Likelihood')
plt.plot(mus, posterior, label='Posterior')
plt.axvline(mu_map, color='k', linestyle='--', label='MAP Estimate')
plt.legend()
plt.title('MAP Estimation for Gaussian Mean')
plt.xlabel('μ')
plt.show()

print('MAP estimate of μ:', mu_map)
```

6. Frequently Asked Questions (FAQ)

1. **How does MAP differ from MLE?** MAP incorporates prior beliefs; MLE does not.
2. **When is MAP preferred?** When prior knowledge is available or data is limited.
3. **Does MAP always yield a unique solution?** Not always—depends on the prior and likelihood.
4. **Is MAP Bayesian?** It is a Bayesian point estimate, but not full Bayesian inference (which uses the entire posterior).

7. Higher-Order Thinking Questions (HOTS)

- Derive the MAP estimate for the variance of a Gaussian with an inverse-gamma prior.
- Discuss the effect of different priors on the MAP estimate.
- How would you use MAP estimation in a Bayesian neural network?

In [ ]:

```python
import numpy as np
import matplotlib.pyplot as plt

# Data and prior
np.random.seed(0)
data = np.random.normal(loc=2, scale=2, size=100)
mu0 = 0
sigma2 = 4  # variance of likelihood
N = len(data)
tau2 = 1    # variance of prior

# MAP estimate for mean with Gaussian prior
sample_mean = np.mean(data)
mu_map = (tau2 / (N * sigma2 + tau2)) * mu0 + (N * sigma2 / (N * sigma2 + tau2)) * sample_mea
print('MAP estimate of μ:', mu_map)

# Plot prior, likelihood, posterior
mus = np.linspace(-1, 4, 200)
prior = 1/np.sqrt(2 * np.pi * tau2) * np.exp(-0.5 * (mus - mu0) ** 2 / tau2)
likelihood = 1/np.sqrt(2 * np.pi * (sigma2/N)) * np.exp(-0.5 * (mus - sample_mean) ** 2 / (si
```

```
posterior_var = 1 / (1/tau2 + N/sigma2)
posterior_mean = mu_map
posterior = 1/np.sqrt(2 * np.pi * posterior_var) * np.exp(-0.5 * (mus - posterior_mean) ** 2 

plt.plot(mus, prior, label='Prior')
plt.plot(mus, likelihood, label='Likelihood')
plt.plot(mus, posterior, label='Posterior')
plt.axvline(mu_map, color='k', linestyle='--', label='MAP Estimate')
plt.legend()
plt.title('MAP Estimation for Gaussian Mean (from scratch)')
plt.xlabel('μ')
plt.show()
```