STA 3180 Statistical Modelling: Bayesian Inference

Bayesian Inference

Bayesian inference is a statistical technique used to make inferences about unknown parameters based on observed data. It is based on the Bayes theorem, which states that the probability of an event given some prior knowledge is equal to the probability of the prior knowledge given the event. This theorem can be used to estimate the probability of an event given some observed data.

Key Concepts

* Prior Probability: The probability of an event before any data is observed.

* Posterior Probability: The probability of an event after data is observed.

* Likelihood: The probability of observing the data given the event.

* Posterior Distribution: The distribution of the posterior probabilities for all possible values of the unknown parameter.

Definitions

* Bayes Theorem: The probability of an event given some prior knowledge is equal to the probability of the prior knowledge given the event.

* Maximum A Posteriori (MAP) Estimation: A method of estimating the unknown parameter by finding the value that maximizes the posterior distribution.

* Markov Chain Monte Carlo (MCMC): A method of sampling from a posterior distribution to estimate the unknown parameter.

Practice Multiple Choice Questions

Q1. What is the maximum a posteriori (MAP) estimation?

A. A method of estimating the unknown parameter by finding the value that maximizes the posterior distribution.

Q2. What is the Markov chain Monte Carlo (MCMC) method?

A. A method of sampling from a posterior distribution to estimate the unknown parameter.

Coding Examples

Example 1: Maximum A Posteriori (MAP) Estimation

```
Start of Code
```

```
import numpy as np

def MAP_estimate(prior, likelihood):
        # Compute the posterior distribution
        posterior = prior * likelihood

        # Find the value that maximizes the posterior distribution
        max_posterior = np.argmax(posterior)

        return max_posterior
```

End of Code

Example 2: Markov Chain Monte Carlo (MCMC)

Start of Code

```
import numpy as np

def MCMC(prior, likelihood, num_samples):
        # Initialize the chain
        chain = np.zeros(num_samples)
        chain[0] = np.random.choice(prior.shape[0])

        # Sample from the posterior distribution
        for i in range(1, num_samples):
                # Compute the posterior distribution
                posterior = prior * likelihood

                # Sample from the posterior distribution
                chain[i] = np.random.choice(posterior.shape[0], p=posterior)

        return chain
```

End of Code