1 Probability

1.1 Bernoulli and Binomial Distribution

The probability distribution over a binary random variable $x \in \{0, 1\}$ can be described using the **Bernoulli distribution** as:

$$\text{Bern}(x|p) = p^x(1-p)^{1-x}$$

Consider n independent variables $X_1, X_2, \ldots, X_n$, and each follows $\text{Bern}(p)$. The sum of these variables is called a binomial random variable:

$$X = \sum_i^n X_i$$

It follows the **binomial distribution**:

$$\text{Bin}(m|n, p) = \binom{n}{m} p^m (1-p)^{n-m}$$

Where n is the number of trials, and m is the number of successes. The mean and the variance of the Binomial distribution are given by

$$\mathbb{E}[m] = np$$
$$\mathbb{V}[m] = np(1-p)$$

1.2 Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) is a statistical method used to estimate the parameters of a probability by maximizing the likelihood function.

Consider $X \sim \text{Bern}(n, \mu)$ and a data set $\mathcal{D} = \{x_i\}_1^n$ is observed. The likelihood function can be formulated under the assumption that observations are independently sampled from the distribution $p(x|\mu)$:

$$p(\mathcal{D}|\mu) = \prod_{i=1}^n p(x_i|\mu)$$

The log-likelihood can be written as

$$\ln p(\mathcal{D}|\mu) = \sum_{i=1}^n \ln p(x_i|\mu) = \sum_{i=1}^n x_i \ln \mu + (1-x_i)\ln(1-\mu)$$

Find the partial derivative of $\ln p(\mathcal{D}|\mu)$ w.r.t μ and set it to zero, we obtain

$$\mu_{ML} = \frac{1}{n}\sum_{i}^{n} x_i$$

Therefore, the MLE of the mean is equivalent to the sample mean.

1.3 Beta Distribution

The Beta distribution is a continuous probability distribution defined on the interval $[0, 1]$:

$$f(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}$$

Where $B(\alpha, \beta)$ is the **Beta function**, defined as:

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt$$

Bayesian inference is a method of statistical inference that updates our belief about unknown parameter using observed data, following Bayes' theorem:

$$P(\theta|X) = \frac{P(X|\theta)P(\theta)}{P(X)}$$

Where:

- $P(\theta|X)$ is the **posterior probability** of the parameter θ given the observed data X.
- $P(X|\theta)$ is the **likelihood**, the probability of observing the data given θ.
- $P(\theta)$ is the **prior**, our belief about θ before seeing the data.
- $P(X)$ is the **evidence**, a constant ensuring the total probability sums to 1.

Because the evidence $P(X)$ is a normalizing constant ensuring that the total probability sums to 1, we can write:

$$P(\theta|X) \propto P(X|\theta)P(\theta)$$

The beta distribution is the **conjugate prior probability distribution** for the Bernoulli, binomial, negative binomial, and geometric distributions.

Consider $X \sim \text{Bern}(n, \mu)$ and a data set $\mathcal{D} = \{x_i\}_1^n$ is observed. Assume that the prior is a Beta distribution:

$$\text{Beta}(\mu|a_0, b_0) = \frac{1}{B(a_0, b_0)}\mu^{a_0-1}(1-\mu)^{b_0-1}$$

We have:

$$\begin{aligned}
p(\mu|a_0, b_0, \mathcal{D}) &\propto p(\mathcal{D}|\mu)\text{Beta}(\mu|a_0, b_0) \\
&= \left(\prod_{i=1}^{n} \mu_i^x (1-\mu)^{1-x_i}\right) \cdot \frac{1}{B(a_0, b_0)} x^{a_0-1}(1-x)^{b_0-1} \\
&= \mu^m (1-\mu)^{n-m} \cdot \frac{1}{B(a_0, b_0)}\mu^{a_0-1}(1-\mu)^{b_0-1} \\
&\propto \mu^{m+a_0-1}(1-\mu)^{n-m+b_0-1} \\
&\propto \text{Beta}(\mu|m+a0, n-m+b_0)
\end{aligned}$$

1.4 Gaussian Distribution

The **Gaussian distribution** (also known as the **normal distribution**) for a scalar random variable x is defined as:

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\}$$

where μ is the mean and σ^2 is the variance.

For a D-dimensional random vector $\mathbf{x}$, the **multivariate Gaussian distribution** is:

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}$$

where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix.

Given independent and identically distributed observations $\{x_i\}_{i=1}^n$, the likelihood function is:

$$p(\mathbf{x} \mid \mu, \sigma^2) = \prod_{i=1}^{n} \mathcal{N}(x_i \mid \mu, \sigma^2)$$

Taking the natural logarithm of the likelihood function yields:

$$\ln p(\mathbf{x} \mid \mu, \sigma) = -\frac{1}{\sigma^2}\sum_{n=1}^{n}(x_n-\mu)^2 - \frac{n}{2}\ln\sigma^2 - \frac{n}{2}(2\pi)$$

To find the maximum likelihood estimate of μ, we take the derivative w.r.t. μ and set it equal to zero:

$$\frac{\partial}{\partial \mu} \ln p(\mathbf{x} \mid \mu, \sigma) = \frac{2}{\sigma^2} \sum_{i=1}^{n} (x_i - \mu)$$

Solving for μ:

$$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Thus, the maximum likelihood estimate for μ is the sample mean. Likewise, the maximum likelihood estimate for σ^2 is the sample variance. In conclusion:

$$\mu_{ML} = \frac{1}{N} \sum_{n=i}^{n} x_i \quad \sigma^2_{ML} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{ML})^2$$

2 Basic Concepts

2.1 Machine Learning Categories

Machine learning (ML) is a subset of **Artificial Intelligence (AI)** that enables computers to learn from data and improve their performance over time without being explicitly programmed. That primary goal is to develop models that can generalize well to unseen data, making accurate predictions or decisions. Types of machine learning consist of:

- **Supervised Learning**: The model learns from labeled data.
- **Unsupervised Learning**: The model finds patterns in unlabeled data.
- **Semi-supervised Learning**: Uses a small amount of labeled data with a large amount of unlabeled data.
- **Reinforcement Learning**: The model learns by interacting with an environment and receiving rewards or penalties.

Machine learning problems are categorized based on the type of output they generate and the nature of the task:

- **Classification**: Assign data points to predefined categories (labels). The output is discrete and categorical.
  - Spam detection
  - Image classification
  - Disease diagnosis

- **Regression**: Predict continuous values based on input data.
  - Predicting house prices.
  - Estimating stock prices.
  - Forecasting temperature.
- **Clustering**: Group similar data points together without predefined labels.
  - Customer segmentation (grouping customers by shopping behavior).
  - Document clustering (organizing articles by topics).
  - Image segmentation.
- **Density Estimation**: Estimate the probability density function of a dataset. The output is a continuous function describing the data distribution.
  - Anomaly detection (fraud detection, network security).
  - Data generation.
  - Feature engineering (transforming features based on density).

3 Polynomial Curve Fitting

Consider a **training set** of N paired observations $(x_i, y_i)_{i=1}^N$, where $x_i \in \mathbb{R}$ are the inputs and $y_i \in \mathbb{R}$ are the corresponding outputs. Let $f : \mathbb{R} \to \mathbb{R}$ be the function that maps inputs to outputs.

Our objective is to predict values $\hat{y}$ of the target variable for new input values x using the training set. We approach this through **curve fitting**, specifically by fitting a polynomial function of the form:

$$\hat{y} = g(x, \mathbf{w}) = \sum_{i=0}^{M} w_i x^i \tag{3.1}$$

Where M is the order of the polynomial and $\mathbf{w} = (w_0, \cdots w_M)^T$ is the vector of **polynomial coefficient** to be determined from the training data.

To find optimal coefficients $\mathbf{w}$, we minimize an **error function** that quantifies the discrepancy between $g(x, \mathbf{w})$ and the observed values y. We use the **sum-of-squares error (SSE)**:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{x} (\hat{y} - y)^2 = \frac{1}{2} \sum_{n=1}^{N} [g(x_n, \mathbf{w}) - f(x_n)]^2 \tag{3.2}$$

Let us define the polynomial basis vector:

$$p_M(x) = (1, x, x^2, \cdots, x^M)^T$$

This allows us to expression equation (3.1) in matrix form:

$$g(x, \mathbf{w}) = \mathbf{X}\mathbf{w} \tag{3.3}$$

Where $\mathbf{X}$ is the $N \times (M+1)$ design matrix whose n-th row is $p_M(x_n)^T$.

The equation 3.2 can then be rewritten as:

$$E(\mathbf{w}) = \frac{1}{2}||\mathbf{X}\mathbf{w} - \mathbf{y}||^2 = \mathbf{w}^T\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}\mathbf{w}\mathbf{y} + \mathbf{y}^T\mathbf{y} \tag{3.4}$$

To minimize $E(\mathbf{w})$, we set its gradient with respect to $\mathbf{w}$ to zero:

$$\frac{d}{d\mathbf{w}}E(\mathbf{w}) = 2\mathbf{X}^T\mathbf{X}\mathbf{w} - 2\mathbf{X}^T\mathbf{y} = 0$$

Solving for $\mathbf{w}$ yields:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{3.5}$$

In practice, observed data often contains noise, and higher-order polynomials are prone to overfitting. To mitigate overfitting, we introduce **regularization** by adding a penalty term to the error function:

$$E(\mathbf{w}) = \frac{1}{2}\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2 \tag{3.6}$$

where $\lambda > 0$ is the **regularization parameter** that controls the trade-off between fitting the data and keeping the polynomial coefficients small. This form of regularization is known as **L2 regularization** or **ridge regression**.

Setting the gradient to zero and solving for $\mathbf{w}$, we obtain:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_{M+1})^{-1}\mathbf{X}^T\mathbf{y} \tag{3.7}$$

where $\mathbf{I}_{M+1}$ is the $(M+1) \times (M+1)$ identity matrix.