

Statistics: Expectation

Hoàng-Nguyên Vũ

1. Lý thuyết về Expectation (Kỳ vọng)

Trong xác suất, **Expectation** của một biến ngẫu nhiên X là giá trị trung bình của X nếu thí nghiệm được lặp đi lặp lại nhiều lần.

1.1 Kỳ vọng của biến ngẫu nhiên rời rạc (Discrete)

$$\mathbb{E}[X] = \sum_i x_i \cdot P(X = x_i)$$

1.2 Kỳ vọng của biến liên tục (Continuous)

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x \cdot f(x) dx$$

Trong đó:

- x_i : các giá trị có thể xảy ra của biến ngẫu nhiên
- $P(X = x_i)$: xác suất của x_i
- $f(x)$: hàm mật độ xác suất

1.3 Ứng dụng trong AI

Expectation được sử dụng rộng rãi trong:

- Dự đoán giá trị trung bình đầu ra
- Học tăng cường: tính phần thưởng kỳ vọng
- Loss Function trong huấn luyện mô hình

1.4 Tính toán bằng NumPy

```
1 import numpy as np
2
3 x = np.array([1, 2, 3])
4 p = np.array([0.2, 0.5, 0.3])
5
6 expectation = np.sum(x * p)
7 print("E[X] =", expectation)
```

2. Bài tập thực hành Expectation với NumPy

Bài 1: Tính kỳ vọng của biến rời rạc

Cho biến ngẫu nhiên $X = [2, 4, 6, 8]$ với xác suất $P = [0.1, 0.3, 0.4, 0.2]$.

Yêu cầu: Viết code Python để tính $\mathbb{E}[X]$.

Bài 2: Expectation từ phân phối chuẩn

Tạo dữ liệu gồm 10,000 mẫu tuân theo phân phối chuẩn $\mu = 5, \sigma = 2$. Tính giá trị kỳ vọng gần đúng từ dữ liệu.

```
1 data = np.random.normal(5, 2, 10000)
2 expectation = np.mean(data)
```

Bài 3: Kỳ vọng của loss function trong AI

Cho:

```
1 y_pred = np.array([0.1, 0.4, 0.6, 0.9])
2 y_true = np.array([0, 0, 1, 1])
```

Tính kỳ vọng của Binary Cross-Entropy Loss:

$$\text{Loss} = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

Bài 4: Kỳ vọng phần thưởng trong Reinforcement Learning

Giả sử phần thưởng từ 0 đến 10, phân phối:

```
1 rewards = np.random.choice(
2     np.arange(0, 11), size=100,
3     p=[0.05, 0.1, 0.1, 0.15, 0.1, 0.1, 0.1, 0.1, 0.05, 0.1, 0.05]
4 )
```

Tính kỳ vọng của phần thưởng.

Bài 5: So sánh kỳ vọng của hai mô hình AI

Dự đoán xác suất:

```
1 model1 = np.array([[0.2, 0.5, 0.3],
2                    [0.1, 0.7, 0.2]])
3
4 model2 = np.array([[0.3, 0.4, 0.3],
5                    [0.2, 0.6, 0.2]])
6
7 labels = np.array([1, 1]) # ground truth
```

Tính kỳ vọng cross-entropy loss cho từng mô hình và chọn mô hình tốt hơn.