

费斯曼算法 (Fano Algorithm)

算法介绍

费斯曼算法是一种无前缀编码方法，通过递归地将符号集合划分为两组，使得两组的概率尽可能接近相等，从而生成高效的二进制编码。其核心步骤包括排序、划分和递归编码。

算法步骤

- 排序**：按符号的概率降序排列。
- 划分**：将符号划分为两组，使两组的概率总和尽可能接近相等。
- 赋码**：给第一组的符号添加前缀 0，给第二组的符号添加前缀 1。
- 递归**：对每一组重复步骤 2 和 3，直到每组只剩一个符号。
- 完成**：将最终生成的编码组合起来，得到每个符号的二进制表示。

示例

假设有以下符号及其概率分布：

符号	概率
A	0.4
B	0.2
C	0.2
D	0.1
E	0.1

1. 排序

按概率降序排列：

A(0.4), B(0.2), C(0.2), D(0.1), E(0.1)

2. 第一次划分

将符号集合划分为两组，使两组的概率总和尽可能接近：

- 第一组： A(0.4) （概率 0.4）
- 第二组： B(0.2), C(0.2), D(0.1), E(0.1) （概率 0.6）

赋码：

- 第一组前缀为 0

- 第二组前缀为 1

3. 对第二组递归划分

第二组: $B(0.2), C(0.2), D(0.1), E(0.1)$

- 划分:
 - 第一子组: $B(0.2), C(0.2)$ (概率 0.4)
 - 第二子组: $D(0.1), E(0.1)$ (概率 0.2)
- 赋码:
 - 第一子组前缀为 10
 - 第二子组前缀为 11

对每个子组继续递归

- 子组 $B(0.2), C(0.2)$:
 - 划分: $B(0.2)$ 和 $C(0.2)$
 - 赋码:
 - B: 100
 - C: 101
- 子组 $D(0.1), E(0.1)$:

- 划分： $D(0.1)$ 和 $E(0.1)$
- 赋码：
 - D: 110
 - E: 111

最终编码结果

符号	概率	编码
A	0.4	0
B	0.2	100
C	0.2	101
D	0.1	110
E	0.1	111

特点

- 无前缀性：生成的编码是前缀码，解码时不会产生歧义。
- 压缩效率：适用于概率分布较均匀的数据，但在某些情况下不如霍夫曼编码高效。
- 简单实现：基于递归划分的思想，易于手动操作或教学演示。

代码实现

以下是 Python 的费斯曼算法实现：

```
# Fano Algorithm Implementation
def fano_encoding(symbols, probabilities):
    # 排序符号和概率
    symbols_probs = sorted(zip(symbols,
probabilities), key=lambda x: x[1], reverse=True)
    codes = {symbol: "" for symbol, _ in
symbols_probs}

    def divide_and_encode(sub_symbols_probs,
prefix):
        if len(sub_symbols_probs) == 1:
            codes[sub_symbols_probs[0][0]] =
prefix
            return

        # 计算分割点
        total_prob = sum(prob for _, prob in
sub_symbols_probs)
        split_index, cum_prob = 0, 0
        for i, (_, prob) in
enumerate(sub_symbols_probs):
            cum_prob += prob
            if cum_prob >= total_prob / 2:
```

```
        split_index = i + 1
        break

    # 递归处理两组

    divide_and_encode(sub_symbols_probs[:split_index], prefix + "0")

    divide_and_encode(sub_symbols_probs[split_index:], prefix + "1")

    divide_and_encode(symbols_probs, "")
    return codes

# 示例
symbols = ['A', 'B', 'C', 'D', 'E']
probabilities = [0.4, 0.2, 0.2, 0.1, 0.1]
codes = fano_encoding(symbols, probabilities)
print("费斯曼编码结果: ", codes)
```

输出

```
费斯曼编码结果:  {'A': '0', 'B': '100', 'C': '101',
'D': '110', 'E': '111'}
```