pomegranate 框架

——使用 predict_proba 函数运行推理

pomegranate 框架还支持使用 predict_proba 函数运行推理，要求是传入一个值字典，其中键是状态名，值是该状态的观察值。如果我们不提供任何值，我们得到图的边缘，也就是每个变量的值的频率除以从图中随机抽取的无限个样本值。

示例 1：假设我们不输入任何信息。

从返回的结果看，这是放三个离散分布对象，每个对象代表每个变量的边缘分布，且顺序与被放入模型的顺序相同，分别表示 guest、prize 和 monty 变量。从这个边缘分布结果可以看到，什么都不输入时每件事都是等可能的。

```
In [9]: model.predict_proba({})
```

```
Out[9]: array([{
            "frozen" :false,
            "dtype" :"str",
            "class" :"Distribution",
            "parameters" :[
                {
                    "A" :0.3333333333333333,
                    "C" :0.3333333333333333,
                    "B" :0.3333333333333333
                }
            ],
            "name" :"DiscreteDistribution"
        },
               {
            "frozen" :false,
            "dtype" :"str",
            "class" :"Distribution",
            "parameters" :[
                {
                    "A" :0.3333333333333333,
                    "C" :0.3333333333333333,
                    "B" :0.3333333333333333
                }
            ],
            "name" :"DiscreteDistribution"
        },
               {
            "frozen" :false,
            "dtype" :"str",
            "class" :"Distribution",
            "parameters" :[
                {
                    "A" :0.3333333333333333,
                    "C" :0.3333333333333333,
                    "B" :0.3333333333333333
                }
            ],
            "name" :"DiscreteDistribution"
        }], dtype=object)
```

示例 2：假设客人选择了门 A。为此，我们将这个字典信息传递给 predict_proba。返回的结果是奖品分布概率并没有改变，奖品在每扇门后面的可能性仍然是相等的，同时主持人不会打开“A”门，因为参赛者选择了它。

```
In [11]: model.predict_proba(['A', None, None])

Out[11]: array(['A',
        {
    "frozen" :false,
    "dtype" :"str",
    "class" :"Distribution",
    "parameters" :[
        {
            "A" :0.3333333333333333,
            "C" :0.3333333333333333,
            "B" :0.3333333333333333
        }
    ],
    "name" :"DiscreteDistribution"
},
        {
    "frozen" :false,
    "dtype" :"str",
    "class" :"Distribution",
    "parameters" :[
        {
            "A" :0.0,
            "C" :0.49999999999999983,
            "B" :0.49999999999999983
        }
    ],
    "name" :"DiscreteDistribution"
}], dtype=object)
```

示例 3：假设参赛者选择了“A”门，主持人打开了“C”门。

我们看到奖品在不同门后的概率发生了变化。现在，汽车在门后被标记为“B”的可能性增加了一倍。这表明，在游戏节目中，在主持人打开一扇门后改变最初的猜测总是更好的。

```
In [12]: model.predict_proba({'guest': 'A', 'monty': 'C'})
```

```
Out[12]: array(['A',
                {
            "frozen" :false,
            "dtype" :"str",
            "class" :"Distribution",
            "parameters" :[
                {
                    "A" :0.3333333333333334,
                    "C" :0.0,
                    "B" :0.6666666666666664
                }
            ],
            "name" :"DiscreteDistribution"
        },
               'C'], dtype=object)
```