

人 工 智 能 实 验 报 告

实验名称: 贝叶斯网络编程实验

学员姓名:	程景愉
学 号:	202302723005
实验日期:	2025.12.31

国防科技大学教育训练部制

《本科实验报告》填写说明

实验报告内容编排应符合以下要求：

(1) 采用 A4 (21cm×29.7cm) 白色复印纸，单面黑字。上下左右各侧的页边距均为 3cm；缺省文档网格：字号为小 4 号，中文为宋体，英文和阿拉伯数字为 Times New Roman，每页 30 行，每行 36 字；页脚距边界为 2.5cm，页码置于页脚、居中，采用小 5 号阿拉伯数字从 1 开始连续编排，封面不编页码。

(2) 报告正文最多可设四级标题，字体均为黑体，第一级标题字号为 4 号，其余各级标题为小 4 号；标题序号第一级用“一、”、“二、”……，第二级用“（一）”、“（二）”……，第三级用“1.”、“2.”……，第四级用“（1）”、“（2）”……，分别按序连续编排。

(3) 正文插图、表格中的文字字号均为 5 号。

目录

实验介绍	4
实验内容	4
实验情况	4
实验步骤与实现	4
网络定义与概率表	4
拒绝采样 (Rejection Sampling)	5
似然加权 (Likelihood Weighting)	5
实验结果	6
实验总结	6

实验介绍

本次实验旨在实现并比较两种贝叶斯网络近似推理算法：拒绝采样（Rejection Sampling）和似然加权（Likelihood Weighting）。实验使用的贝叶斯网络模型包含四个布尔变量：Cloudy (C), Sprinkler (S), Rain (R), WetGrass (W)。网络结构为 $C \rightarrow S, C \rightarrow R, S \rightarrow W, R \rightarrow W$ 。实验目标是根据给定的条件概率表（CPTs），在已知证据 $S = T$ 和 $W = T$ 的情况下，估计 $R = T$ 的后验概率 $P(R = T \mid S = T, W = T)$ 。

实验内容

实验主要包含以下内容：

1. 根据题目描述的贝叶斯网络结构和条件概率表，构建数据结构表示该网络。
2. 实现拒绝采样算法：从先验分布中采样，拒绝与证据不一致的样本，利用保留样本中查询变量的频率作为后验概率的估计。
3. 实现似然加权算法：固定证据变量的值，对非证据变量进行采样，并根据证据变量的条件概率计算样本权重，利用加权频率估计后验概率。
4. 比较两种算法在不同采样数量（ $N=100, 1000, \dots$ ）下的估计结果与真实值的偏差，分析算法收敛性。

实验情况

1. 使用 Python 语言实现算法。
2. 代码需清晰易读，包含必要的注释。
3. 计算并输出 $P(R = T \mid S = T, W = T)$ 的精确值以便对比。
4. 撰写实验报告，展示核心代码和实验结果。

实验步骤与实现

网络定义与概率表

首先定义网络的条件概率表（CPTs）。

```
# CPTs
P_C = {True: 0.5, False: 0.5}

# P(S | C)
P_S_given_C = {
    True: {True: 0.10, False: 0.90}, # C=T
    False: {True: 0.50, False: 0.50} # C=F
}
```

```
# P(R | C)
P_R_given_C = {
    True: {True: 0.80, False: 0.20}, # C=T
    False: {True: 0.20, False: 0.80} # C=F
}

# P(W | S, R)
P_W_given_SR = {
    (True, True): {True: 0.99, False: 0.01},
    (True, False): {True: 0.90, False: 0.10},
    (False, True): {True: 0.90, False: 0.10},
    (False, False): {True: 0.00, False: 1.00}
}
```

拒绝采样 (*Rejection Sampling*)

拒绝采样通过生成先验分布的样本，并丢弃那些与证据不符的样本来工作。

```
def rejection_sampling(query_var, query_val, evidence, num_samples):
    consistent_count = 0
    query_match_count = 0

    for _ in range(num_samples):
        sample = get_prior_sample() # 从联合分布采样

        # 检查样本是否与证据一致
        is_consistent = True
        for var, val in evidence.items():
            if sample[var] != val:
                is_consistent = False
                break

        if is_consistent:
            consistent_count += 1
            if sample[query_var] == query_val:
                query_match_count += 1

    if consistent_count == 0: return 0.0
    return query_match_count / consistent_count
```

似然加权 (*Likelihood Weighting*)

似然加权强制证据变量取观测值，并将样本权重设为证据变量在其父节点给定情况下的似然概率乘积。

```
def likelihood_weighting(query_var, query_val, evidence, num_samples):
    weighted_counts = {True: 0.0, False: 0.0}

    for _ in range(num_samples):
        weight = 1.0
        sample = {}
```

```
# 以 C, S, R, W 的拓扑顺序采样/赋值
# 若变量在证据中, 则赋值为证据值并更新权重
# 若变量不在证据中, 则根据父节点采样
# (代码细节略, 参见附件源程序)
# ...

# 累加权重
weighted_counts[sample[query_var]] += weight

total_weight = sum(weighted_counts.values())
if total_weight == 0: return 0.0
return weighted_counts[query_val] / total_weight
```

实验结果

首先通过精确计算（枚举联合概率分布）得到理论真实值： $P(R = T \mid S = T, W = T) \approx 0.320388$

下表展示了在不同采样数量 N 下，两种算法的估计结果：

样本数量 N	拒绝采样结果	似然加权结果
100	0.480000	0.377902
1000	0.353741	0.311148
10000	0.320734	0.317552
100000	0.318791	0.321722
1000000	0.319916	0.319471

实验总结

- 收敛性：从实验结果可以看出，随着采样数量 N 的增加，拒绝采样和似然加权算法的结果都逐渐收敛于真实值 0.320388。
- 算法比较：
 - 拒绝采样：实现简单，直接利用先验分布采样。但在证据概率较低时（本例中证据发生的概率约为 0.278，尚可），会拒绝大量样本，导致计算资源浪费。在样本量较小（如 $N=100$ ）时波动较大。
 - 似然加权：利用了证据信息，所有生成的样本都是有效的（权重不为 0），因此在相同采样数量下，通常能提供比拒绝采样更稳定的估计，尤其是当证据发生概率很低时优势更明显。

3. 结论：两种算法均能有效进行近似推理。对于本实验中的简单网络和非极端概率证据，两者在 $N = 10000$ 以上时均能给出较好的估计结果。