题目

1.现有两台服务器(S1, S2)，都会单向向用户 U 传送数据。服务器 S1 和 S2 之间也会有数据通讯，但无法确定它们之间的数据流向。数据包的传送只取两种可能值：T=1 (成功) 或 F=2 (失败)。假设贝叶斯网络由 S1、S2和U这三个节点构成，现采集了100条该网络的数据传送样本，如文件 server_data.txt 所给出。该文件中，每行代表一个三节点网络的样本， 试利用贝叶斯算法学习得到该网络的结构和参数。(30分)

参考资料

参考学习网址 https://blog.csdn.net/leida_wt/article/details/88743323 (https://blog.csdn.net/leida_wt/article/details/88743323)

自动设计网络结构的核心问题有两个，一个是评价网络好坏的指标，另一个是查找的方法。穷举是不可取的，因为组合数太大，只能是利用各种启发式方法或是限定搜索条件以减少搜索空间，因此产生两大类方法，Score-based Structure Learning与constraint-based structure learning 以及他们的结合hybrid structure learning。

In [1]:
```
import numpy as np
import pandas as pd
from pgmpy.models import BayesianModel
from pgmpy.estimators import MaximumLikelihoodEstimator, BayesianEstimator
from pgmpy.estimators import BdeuScore, K2Score, BicScore
import matplotlib.pyplot as plt
```
executed in 2.58s, finished 10:45:11 2019-12-29

In [2]:
```
data_list = []
with open('server_data.txt') as f:
    lines = f.readlines()
    for line in lines:
        data_list.append(line.strip().split())
data_list = np.array(data_list, dtype=np.int32)
data = pd.DataFrame(data_list, columns=['S1', 'S2', 'U'])
data
```
executed in 19ms, finished 10:45:11 2019-12-29

Out[2]:

	S1	S2	U
0	1	2	1
1	2	2	2
2	2	1	1
3	2	1	1
4	2	1	1
...	...	...	...
95	2	1	1
96	2	1	1
97	2	1	1
98	2	1	1
99	2	1	1

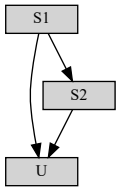
100 rows × 3 columns

In [3]:
```
def showBN(model, save=False):
    '''传入BayesianModel对象，调用graphviz绘制结构图，jupyter中可直接显示'''
    from graphviz import Digraph
    node_attr = dict(
        style='filled',
        shape='box',
        align='left',
        fontsize='12',
        ranksep='0.1',
        height='0.2'
    )
    dot = Digraph(node_attr=node_attr, graph_attr=dict(size="12,12"))
    seen = set()
    edges = model.edges()
    for a, b in edges:
        dot.edge(a, b)
    if save:
        dot.view(cleanup=True)
    return dot
```
executed in 5ms, finished 10:45:11 2019-12-29

根据题目要求分别定义出两种可能的网络

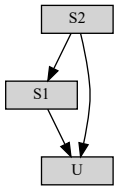
In [4]:
```
model_1 = BayesianModel([('S1', 'U'), ('S2', 'U'), ('S1', 'S2')])
model_1.fit(data)
showBN(model_1)
```
executed in 147ms, finished 10:45:11 2019-12-29

Out[4]:



In [5]:
```
model_2 = BayesianModel([('S1', 'U'), ('S2', 'U'), ('S2', 'S1')])
model_2.fit(data)
showBN(model_2)
```
executed in 100ms, finished 10:45:11 2019-12-29

Out[5]:



评分函数，使用k2，bdeu，bic进行评分

In [6]:
```
bdeu = BdeuScore(data, equivalent_sample_size=5)
k2 = K2Score(data)
bic = BicScore(data)
```
executed in 5ms, finished 10:45:11 2019-12-29

In [7]:
```
print(bdeu.score(model_1))
print(k2.score(model_1))
print(bic.score(model_1))
```
executed in 59ms, finished 10:45:11 2019-12-29

```
-127.81019191674014
-130.82411202574002
-129.03972756462477
```

In [8]:
```
print(bdeu.score(model_2))
print(k2.score(model_2))
print(bic.score(model_2))
```
executed in 34ms, finished 10:45:11 2019-12-29

```
-127.81019191674014
-130.99837093511061
-129.0397275646248
```

In [9]:
```
bdeu.score(model_1)>bdeu.score(model_2)
```
executed in 6ms, finished 10:45:11 2019-12-29

Out[9]: False

In [10]:
```
k2.score(model_1)>k2.score(model_2)
```
executed in 7ms, finished 10:45:11 2019-12-29

Out[10]: True

In [11]:
```
bic.score(model_1)>bic.score(model_2)
```
executed in 10ms, finished 10:45:11 2019-12-29

Out[11]: True

查看模型的概率转移表

In [12]:
```
print(model_1.get_cpds('S1'))
print(model_1.get_cpds('S2'))
print(model_1.get_cpds('U'))
```
executed in 7ms, finished 10:45:11 2019-12-29

```
+-------+------+
| S1(1) | 0.28 |
+-------+------+
| S1(2) | 0.72 |
+-------+------+
+-------+---------------------+-------+
| S1    | S1(1)               | S1(2) |
+-------+---------------------+-------+
| S2(1) | 0.17857142857142858 | 0.75  |
+-------+---------------------+-------+
| S2(2) | 0.8214285714285714  | 0.25  |
+-------+---------------------+-------+
+------+-------+-------+-------+-------+
| S1   | S1(1) | S1(1) | S1(2) | S1(2) |
+------+-------+-------+-------+-------+
| S2   | S2(1) | S2(2) | S2(1) | S2(2) |
+------+-------+-------+-------+-------+
| U(1) | 0.0   | 1.0   | 1.0   | 0.0   |
+------+-------+-------+-------+-------+
| U(2) | 1.0   | 0.0   | 0.0   | 1.0   |
+------+-------+-------+-------+-------+
```

In [13]:
```
print(model_2.get_cpds('S1'))
print(model_2.get_cpds('S2'))
print(model_2.get_cpds('U'))
```
executed in 8ms, finished 10:45:11 2019-12-29

```
+-------+---------------------+---------------------+
| S2    | S2(1)               | S2(2)               |
+-------+---------------------+---------------------+
| S1(1) | 0.0847457627118644  | 0.5609756097560976  |
+-------+---------------------+---------------------+
| S1(2) | 0.9152542372881356  | 0.43902439024390244 |
+-------+---------------------+---------------------+
+-------+------+
| S2(1) | 0.59 |
+-------+------+
| S2(2) | 0.41 |
+-------+------+
+------+-------+-------+-------+-------+
| S1   | S1(1) | S1(1) | S1(2) | S1(2) |
+------+-------+-------+-------+-------+
| S2   | S2(1) | S2(2) | S2(1) | S2(2) |
+------+-------+-------+-------+-------+
| U(1) | 0.0   | 1.0   | 1.0   | 0.0   |
+------+-------+-------+-------+-------+
| U(2) | 1.0   | 0.0   | 0.0   | 1.0   |
+------+-------+-------+-------+-------+
```

结论

分数差距不是很大，说明对这组数据来说，题目假定的两种网络的区分度不够高，说明这两种网络的结构可能性都很大。