In [70]:

```
# sprinkler.py
import pylab as pl
import pymc3 as pm
import numpy as np
import theano as tt
```

In [71]:

```
#Cancer - in lecture

a_prob = np.array([0.2,0.8])  # 2 choices
b_prob = np.array([[0.8,0.2],
                   [.2,.8]])  # (2x2)x2 choices
c_prob = np.array([[0.2,0.8],
                   [.05,.95]])  # (2x2)x2 choices
d_prob = np.array([[[0.8, 0.2],  # (2x2)x4 choices
                    [0.8, 0.2]],
                   [[0.8, 0.2],
                    [0.05, 0.95]]])

e_prob = np.array([[0.8,0.2],
                   [.6,.4]])  # (2x2)x2 choices

with pm.Model() as model:
    A = pm.Categorical('A',p=a_prob)

    B_prob = theano.shared(b_prob)  # make numpy-->theano
    B_0 = B_prob[A]  # select the prob array that "happened" thanks to parents
    B = pm.Categorical('B',p=B_0)
    #obs1 = pm.Bernoulli('obs1', p=B, observed=True)

    C_prob = theano.shared(c_prob)  # make numpy-->theano
    C_0 = C_prob[A]  # select the prob array that "happened" thanks to parents
    C = pm.Categorical('C',p=C_0)

    D_prob = theano.shared(d_prob)  # make numpy-->theano
    D_0 = D_prob[B,C]  # select the prob array that "happened" thanks to parents
    D = pm.Categorical('D',p=D_0)

    E_prob = theano.shared(e_prob)  # make numpy-->theano
    E_0 = E_prob[C]  # select the prob array that "happened" thanks to parents
    E = pm.Categorical('E',p=E_0)
    #obs2 = pm.Bernoulli('obs2', p=E, observed=False)

    trace = pm.sample(10000)

    print("summary=", pm.summary(trace, varnames=['A', 'B', 'C','D','E'], start=1000))
    #pm.traceplot(trace[1000:], varnames=['D1', 'D2', 'D3'])

    #map_estimate = pm.find_MAP(model=model) #doesn't work
    #map_estimate
```

```
Multiprocess sampling (4 chains in 4 jobs)
BinaryGibbsMetropolis: [A, B, C, D, E]
Sampling 4 chains: 100%|██████████| 42000/42000 [00:08<00:00, 5018.01draws/s]
The number of effective samples is smaller than 25% for some parameters.
```

```
summary=        mean        sd  mc_error  hpd_2.5  hpd_97.5         n_eff      Rhat
A  0.799417  0.400437  0.003281      0.0       1.0  12749.663707  1.000067
B  0.683028  0.465296  0.005683      0.0       1.0   6219.321961  1.000151
C  0.916139  0.277179  0.001956      0.0       1.0  20736.972469  1.000013
D  0.679278  0.466754  0.005456      0.0       1.0   6606.420108  1.000141
E  0.380111  0.485414  0.002023      0.0       1.0  67088.340937  0.999970
```

In [ ]:

```
x = np.array([0,1])
```

```
with pm.Model() as model:
    lambda_1 = pm.Exponential("lambda_1", 1.0)
    lambda_2 = pm.Exponential("lambda_2", 1.0)
    tau = pm.DiscreteUniform("tau", lower=0, upper=10)


    tmp = pm.Categorical('tmp',[0.7,0.3])

    X = pm.Deterministic('X',theano.shared(x)[tmp])

    p1 = pm.Uniform("p", 0, 1)
    p2 = 1 - p1
    p = tt.stack([p1, p2])

    assignment = pm.Categorical("assignment", p)

    trace = pm.sample()


new_deterministic_variable = lambda_1 + lambda_2
print("lambda1 = ", lambda_1.tag.test_value)
print("lambda2 = ", lambda_2.tag.test_value)
print("\nnew_deterministic_variable = ", new_deterministic_variable.tag.test_value)

print("\ntrace=", trace)
```

In [ ]:

```
d1_prob = np.array([0.3,0.7])  # 2 choices
d2_prob = np.array([0.6,0.3,0.1])  # 3 choices
d3_prob = np.array([[[0.1, 0.9],  # (2x3)x2 choices
                     [0.3, 0.7],
                     [0.4, 0.6]],
                    [[0.6, 0.4],
                     [0.8, 0.2],
                     [0.9, 0.1]]])

with pm.Model() as model:
    D1 = pm.Categorical('D1',p=d1_prob)
    D2 = pm.Categorical('D2',p=d2_prob)

    D3_prob = theano.shared(d3_prob)  # make numpy-->theano
    D3_0 = D3_prob[D1, D2]  # select the prob array that "happened" thanks to parents
    D3 = pm.Categorical('D3',p=D3_0)

    trace = pm.sample(10000)

    print("summary=", pm.summary(trace, varnames=['D1', 'D2', 'D3'], start=1000))
    #pm.traceplot(trace[1000:], varnames=['D1', 'D2', 'D3'])
```

In [ ]:

```
#Cancer with MCMC

a_prob = np.array([0.2,0.8])  # 2 choices
b_prob = np.array([[0.8,0.2],
                   [.2,.8]])  # (2x2)x2 choices
c_prob = np.array([[0.2,0.8],
                   [.05,.95]])  # (2x2)x2 choices
d_prob = np.array([[[0.9, 0.2],  # (2x2)x4 choices
                     [0.8, 0.2]],
                    [[0.8, 0.2],
                     [0.05, 0.95]]])

with pm.Model() as model:
    A = pm.Categorical('A',p=a_prob)

    B_prob = theano.shared(b_prob)  # make numpy-->theano
    B_0 = B_prob[A]  # select the prob array that "happened" thanks to parents
    B = pm.Categorical('B',p=B_0)
```

```
    C_prob = theano.shared(c_prob)   # make numpy-->theano
    C_0 = C_prob[A]  # select the prob array that "happened" thanks to parents
    C = pm.Categorical('C',p=C_0)

    #trace = pm.sample(50000)

    #pm.traceplot(trace[1000:], varnames=['D1', 'D2', 'D3'])
    #start = pm.find_MAP()
    step = pm.Metropolis()
    trace = pm.sample(10000, step=step)
    burned_trace = trace[1000::2]

    print("summary=", pm.summary(trace, varnames=['A', 'B', 'C'], start=1000))
```

In [ ]:

```
```