MI - H10

January 19, 2017

In [58]:
```
from bayespy.nodes import Bernoulli, Mixture, MultiMixture
from bayespy.inference import VB
import itertools
```

In [46]:
```
# 2. Water sprinkler example using BayesPy
exp = 1e-20 # BayesPy has problems with 1.0 -> substract a small value
cloudy = Categorical([0.5, 0.5])
rain = Mixture(
    cloudy,
    Categorical,
    [
        [0.8, 0.2], # p(rain | not cloudy) = 0.2
        [0.2, 0.8], # p(rain | cloudy) = 0.8
    ]
)
sprinkler = Mixture(
    cloudy,
    Categorical,
    [
        [0.5, 0.5], # p(sprinkler on | not cloudy) = 0.5
        [0.9, 0.1], # p(sprinkler on | cloudy) = 0.1
    ]
)
wetGrass = MultiMixture(
    [rain, sprinkler],
    Categorical,
    [
        # No rain
        [
            # No sprinkler
            [1.0 - exp, 0.0 + exp],
            # Yes sprinkler
            [0.1, 0.9]
        ],
        # Yes rain
        [
            # No sprinkler
            [0.1, 0.9],
```

```
                    # Yes sprinkler
                    [0.01, 0.99]
                ]
            ]
        )

        wetGrass.observe(1)

        Q = VB(wetGrass, rain, sprinkler, cloudy)
        Q.update(repeat=100, verbose=True)

        print('P(s=1|w=1) =', sprinkler.get_moments()[0][1])
        print('P(r=1|w=1) =', rain.get_moments()[0][1])
        # FIXME: Returns wrong results
```

```
Iteration 1: loglike=-8.859004e-01 (0.046 seconds)
Iteration 2: loglike=-8.434570e-01 (0.019 seconds)
Iteration 3: loglike=-8.430205e-01 (0.026 seconds)
Iteration 4: loglike=-8.430174e-01 (0.026 seconds)
Converged at iteration 4.
P(s=1|w=1) = 0.148459431093
P(r=1|w=1) = 1.0
```

In [63]:
```
# Hotfix: Manually calculate values
p_c = np.array([0.5, 0.5])  # p(c)
p_cs = np.array([[0.5, 0.5], [0.9, 0.1]])  # p(s|c)
p_cr = np.array([[0.8, 0.2], [0.2, 0.8]])  # p(r|c)
p_rsw = np.array([[[1.0, 0.0], [0.1, 0.9]], [[0.1, 0.9], [0.01, 0.99]]])

def p(c=[0, 1], r=[0, 1], s=[0, 1], w=[0, 1]):
    p_sum = 0.0
    for ci, ri, si, wi in itertools.product(c, r, s, w):
        p_sum += p_c[ci] * p_cs[ci, si] * p_cr[ci, ri] * p_rsw[ri, si, wi]
    return p_sum

print('P(s=1|w=1) =', p(s=[1], w=[1]) / p(w=[1]))
print('P(r=1|w=1) =', p(r=[1], w=[1]) / p(w=[1]))
print('P(s=1|w=1, r=1) =', p(s=[1], w=[1], r=[1]) / p(w=[1], r=[1]))
```

```
P(s=1|w=1) = 0.429763560501
P(r=1|w=1) = 0.70792767733
P(s=1|w=1, r=1) = 0.194499017682
```