

EM-NoisyOR

November 14, 2017

```
In [19]: import numpy as np
         from math import log

         with open('spectX.txt') as f:
             content = f.readlines()
             content = [line.strip() for line in content]
             X = []
             for line in content:
                 x = line.split(' ')
                 x = list(map(int,x))
                 X.append(x)
             X = np.array(X)

         with open('spectY.txt') as f:
             content = f.readlines()
             content = [line.strip() for line in content]
             y = []
             for line in content:
                 yi = int(line)
                 y.append(yi)
             y = np.array(y)

         p = np.array([0.05]*X.shape[1])

In [52]: def inner(X,p):
         result = 1.0
         for i in range(len(X)):
             result *= (1-p[i])**X[i]
         return result

         def noisyOR_ll(X, y, p):
             ll = 0.0
             for i in range(X.shape[0]):
                 in_prod = inner(X[i],p)
                 if y[i]:
                     ll += log(1 - in_prod)
                 if not y[i]:
```

```

        ll += log(in_prod)
    return ll*1.0/X.shape[0]

def update_p(X, y, p):
    new_p = []
    t = X.sum(axis=0)
    for i in range(len(p)):
        up_val = 0.0
        for j in range(X.shape[0]):
            in_prod = inner(X[j],p)
            den = 1 - in_prod
            num = y[j]*p[i]*X[j][i]
            up_val += num*1.0/den
        new_p.append(up_val*1.0/t[i])
    return new_p

def calc_mistakes(X, y, p):
    mistakes = 0
    for i in range(X.shape[0]):
        in_prod = inner(X[i],p)
        if 1-in_prod >= 0.5 and y[i] == 0:
            mistakes += 1
        if 1-in_prod <= 0.5 and y[i] == 1:
            mistakes += 1
    return mistakes

def em_update(n, X, y, p):
    pfor = [0,1,2,4,8,16,32,64,128,256]
    mistake = []
    ll = []
    iterations = 0
    while iterations <= n:
        if iterations in pfor:
            log_likelihood = noisyOR_ll(X, y, p)
            ll.append(log_likelihood)
            mistakes = calc_mistakes(X, y, p)
            mistake.append(mistakes)
            p = update_p(X, y, p)
            iterations += 1
    return pfor, ll, mistake

```

In [53]: pfor, ll, mistakes = em_update(256, X, y, p)

```

In [59]: import pandas as pd
         pd.DataFrame({'Iterations':pfor, \
                      'Number of Mistakes':mistakes, \
                      'log-likelihood':ll})

```

Out[59]: Iterations Number of Mistakes log-likelihood

0	0	175	-0.958085
1	1	56	-0.495916
2	2	43	-0.408221
3	4	42	-0.364615
4	8	44	-0.347501
5	16	40	-0.334617
6	32	37	-0.322581
7	64	37	-0.314827
8	128	36	-0.311156
9	256	36	-0.310161