test

April 27, 2015

In [16]:

In [1]:
```
%%file Bayhiecluster.py

import numpy as np
from scipy.stats import multivariate_normal
import scipy
from scipy.special import gamma

def marginal_likelihood_NIW(points):
    n=len(points)
    p=len(points[0])
    mean_data = np.mean(points, axis=0)
    sum_squares = np.sum([np.array(np.matrix(x - mean_data).T * np.matrix(x - mean_data)) for x
    k0=3.0
    v0=10
    sigma0=np.eye(p)
    mu0=[0]*p
    kn=k0+n
    vn=v0+n
    sigman=sigma0+sum_squares+(float(k0*n)/(k0+n))*np.dot((mean_data-mu0).reshape(p,1),(mean_da
    gammadn=reduce(lambda x,y: x*y, map(lambda x:gamma((vn+1-x)*0.5),range(1,p+1)))
    gammad0=reduce(lambda x,y: x*y, map(lambda x:gamma((v0+1-x)*0.5),range(1,p+1)))
    mar_like=np.pi**(-p*n*0.5)*(np.linalg.det(sigma0))**(v0*0.5)*(k0/kn)**(0.5*p)*gammadn/(floa

    return mar_like

class bicluster:
    def __init__(self, point, left=None,right=None,probability=None,d=None,id=None):
        self.left = left
        self.right = right
        self.point = point
        self.id = id
        self.probability = probability
        self.d=d

def yezi(clust):
    if clust.left == None and clust.right == None :
        return [clust.id]
    return yezi(clust.left) + yezi(clust.right)


#cluster function
def bcluster(data,function) :
```

```
    dim=len(data[0])
    alpha=3
    biclusters = [ bicluster(point = [data[i]], id = i ,probability=0.0000001,d=alpha) for i in

    flag = None;
    currentclusted = -1 #id for the new cluster

    if len(biclusters) == 1:
        clusters = [yezi(biclusters[i]) for i in range(len(biclusters))]
        return biclusters,clusters

    while(len(biclusters) > 1) :
        max_prob = 0;
        biclusters_len = len(biclusters)
        for i in range(biclusters_len-1) :
            for j in range(i + 1, biclusters_len) :

                #calculate P_H1: MC method
                temp_cluster_points= biclusters[i].point + biclusters[j].point
                P_H1=function(temp_cluster_points)
                #P_H1=marginal_likelihood_DW(temp_cluster_points)
                pi=float(scipy.misc.factorial(len(temp_cluster_points)-1))*alpha/(float(scipy.mi
                marginal_prob=pi*P_H1+(1-pi)*biclusters[i].probability*biclusters[j].probabilit
                r = pi*P_H1/marginal_prob
                if r > max_prob :
                    max_prob = r
                    flag = (i,j)

        if max_prob<0.5:
            break

        bic1,bic2 = flag

        newpoint = biclusters[bic1].point + biclusters[bic2].point #combine the points of two c
        P_H1=function(newpoint)
        #P_H1=marginal_likelihood_DW(newpoint)

        newprob=pi*P_H1+(1-pi)*biclusters[bic1].probability*biclusters[bic2].probability
        newd=float(scipy.misc.factorial(len(newpoint)-1))*alpha+biclusters[bic1].d*biclusters[bi
        newbic = bicluster(point=newpoint, left=biclusters[bic1], right=biclusters[bic2], probal
        currentclusted -= 1

        del biclusters[bic2]
        del biclusters[bic1]
        biclusters.append(newbic)
        clusters = [yezi(biclusters[i]) for i in range(len(biclusters))]

    return biclusters,clusters
```

```
Overwriting Bayhiecluster.py
```

In [2]:
```
%%file test_Bayhiecluster.py
import numpy as np
from scipy.stats import multivariate_normal
import scipy
```

```
from scipy.special import gamma
from numpy.testing import assert_almost_equal
from Bayhiecluster import bcluster, marginal_likelihood_NIW, yezi

#parameters for test data
mean0=(0,0)
cov0=np.eye(2)

mean2=(0,0)
cov2=0.5*np.eye(2)

mean3=(2,2)
cov3=0.5*np.eye(2)

mean4=(8,8)
cov4=0.5*np.eye(2)


def test_atleast_onecluster():
    for i in range(20):
        data2=np.random.multivariate_normal(mean2,cov2,3)
        data3=np.random.multivariate_normal(mean3,cov3,4)
        data4=np.random.multivariate_normal(mean4,cov4,3)
        sample_NIW=np.concatenate((data2, data3,data4), axis=0)
        k,l=bcluster(sample_NIW,marginal_likelihood_NIW)
        assert len(l) >= 1

def test_max_cluster_number():
    for i in range(20):
        data2=np.random.multivariate_normal(mean2,cov2,3)
        data3=np.random.multivariate_normal(mean3,cov3,4)
        data4=np.random.multivariate_normal(mean4,cov4,3)
        sample_NIW=np.concatenate((data2, data3,data4), axis=0)
        k,l=bcluster(sample_NIW,marginal_likelihood_NIW)
        n=len(sample_NIW)
        assert len(l) <= n

def test_if_all_the_same():
    data0=np.random.multivariate_normal(mean0,cov0,1)
    sample0=np.concatenate((data0, data0, data0, data0, data0, data0, data0, data0, data0, data0
    k,l=bcluster(sample0,marginal_likelihood_NIW)
    assert len(l) == 1

def test_only_one_point():
    special=[[np.random.multivariate_normal(mean0,cov0,1)]]
    k,l=bcluster(special,marginal_likelihood_NIW)
    assert len(l) == 1
```

```
Overwriting test_Bayhiecluster.py
```

```
In [3]: ! py.test
```

```
============================= test session starts ==============================
platform linux2 -- Python 2.7.9 -- py-1.4.25 -- pytest-2.6.3
collected 4 items
```

```
test_Bayhiecluster.py ...

========================== 4 passed in 2.49 seconds ==========================
```

In []: