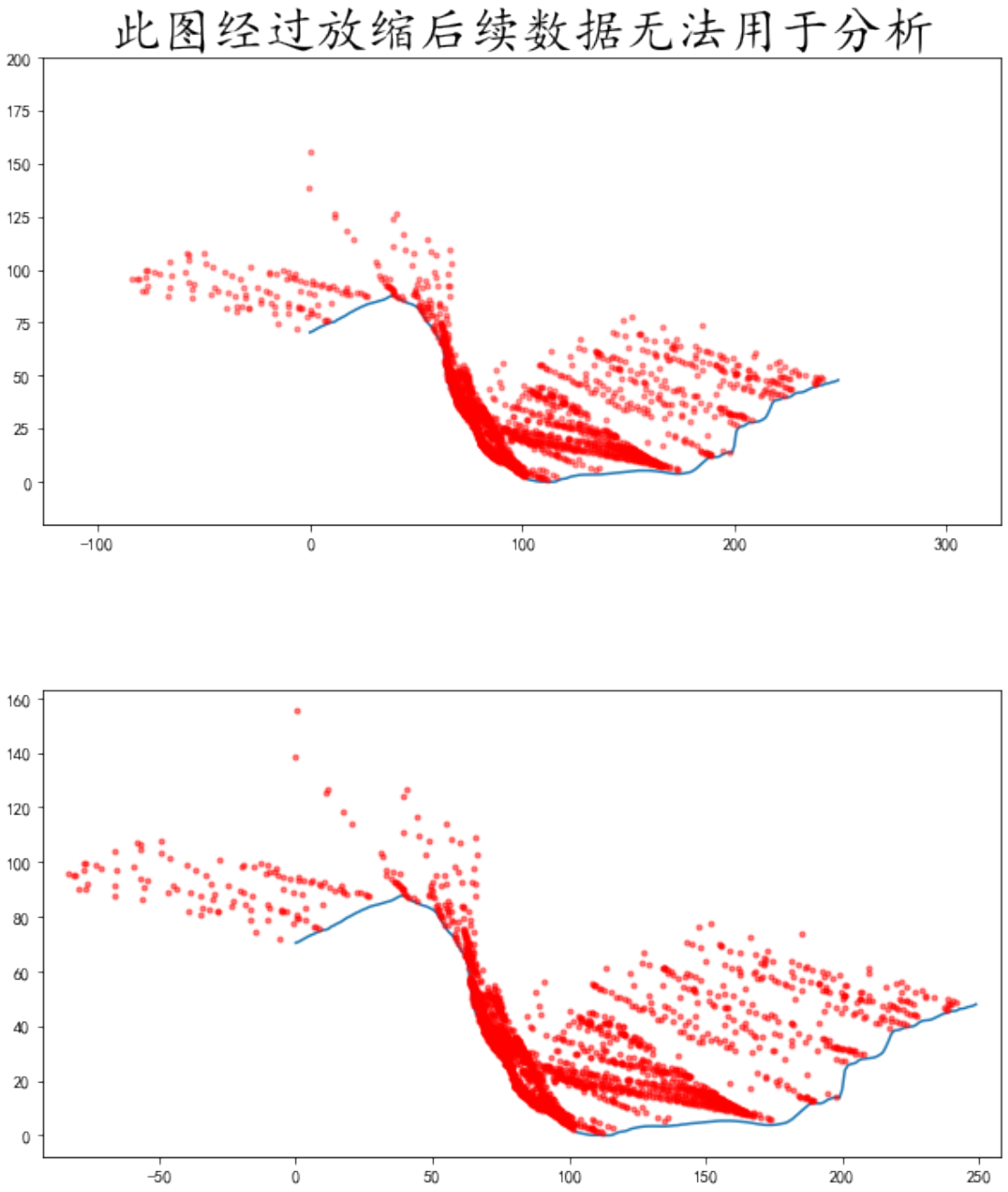
cal_center -0901

September 2, 2020

In [19]:
```
from ssea import *
from matplotlib import pyplot as plt
from sklearn.cluster import AffinityPropagation
%matplotlib inline

crossx_ls,crossy_ls,trash_m = cross_esi('')
train_set = np.c_[crossx_ls,crossy_ls]
#train_set,_ = filt_points(train_set)
#af = AffinityPropagation()
#af.fit(train_set)
#train_res = af.predict(train_set)

plt.figure(figsize = (10,5))
plt.rcParams['font.sans-serif'] = ['KaiTi']
plt.rcParams['axes.unicode_minus']=False
plt.plot(coast_x,coast)
#plt.plot(crossx_ls,crossy_ls,'r.')
plt.plot(train_set[:,0],train_set[:,1],'r.',alpha = 0.4)
plt.axis('equal')
plt.xlim(0,200)
plt.ylim(-20,200)
plt.title('',fontsize = 30)
plt.figure(figsize = (10,5))
plt.plot(coast_x,coast)
#plt.plot(crossx_ls,crossy_ls,'r.')
plt.plot(train_set[:,0],train_set[:,1],'r.',alpha = 0.5)
plt.axis('equal')
#plt.xlim(0,200)
#plt.ylim(-20,200)
plt.show()
```



0.1

91 1. 4 2. AP 3. 4. 5. AP

92:

0

kmeans

. . . K-means

sample_weightfrom sklearn.cluster import KMeans

0.2 KMeans

In [ ]:

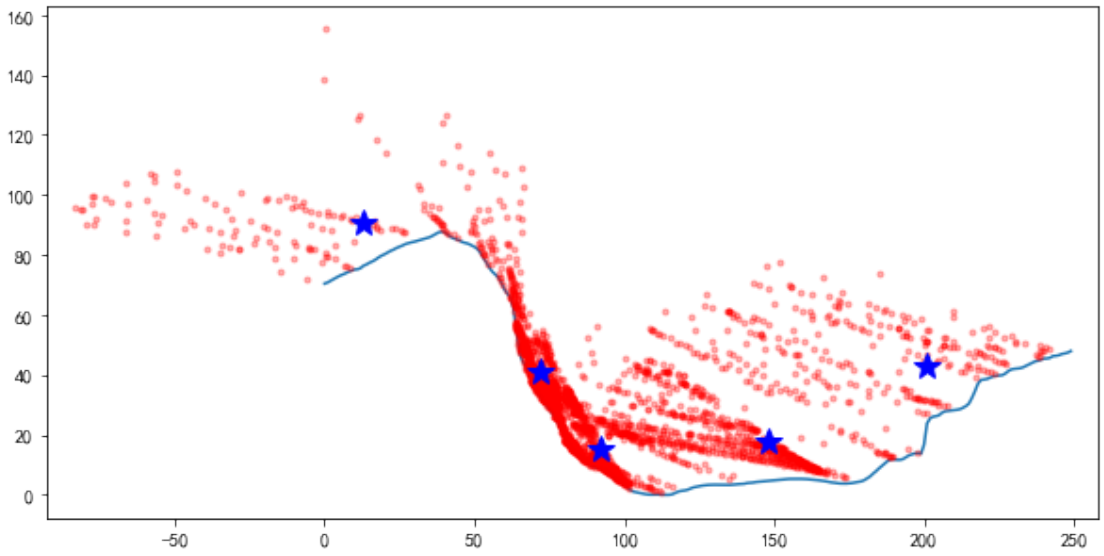
In [34]:
```
from sklearn.cluster import KMeans

okm = KMeans(n_clusters = 5)
okm.fit(train_set,sample_weight = trash_m)
```

Out[34]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=5, n_init=10, n_jobs=None, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

In [77]:
```
c_centers = okm.cluster_centers_

plt.figure(figsize = (10,5))
plt.rcParams['font.sans-serif'] = ['KaiTi']
plt.rcParams['axes.unicode_minus']=False
plt.plot(coast_x,coast)
plt.plot(train_set[:,0],train_set[:,1],'.',color = 'r',alpha = 0.3)
plt.axis('equal')
plt.plot(c_centers[:,0],c_centers[:,1],'b*',markersize = 15)
plt.show()
```



0.2.1 ...AP3-σ

0.2.2

In [6]:
```
train_set_bk = train_set.copy()
print('',train_set.shape[0])
```

17

```
In [7]: train_set = train_set_bk.copy()
        train_set = np.round(train_set)

        train_set = train_set.tolist()
        train_set_tuple = [tuple(i) for i in train_set]
        train_set = set(train_set_tuple)

In [8]: print('',len(train_set))
```

16

0.2.3

```
In [9]: train_set = list(train_set)
        num_ls = [ ]
        for i in train_set:
            num_ls.append(train_set_tuple.count(i))

In [10]: num_ls = np.array(num_ls)
         print('',num_ls[num_ls>1].size)
```

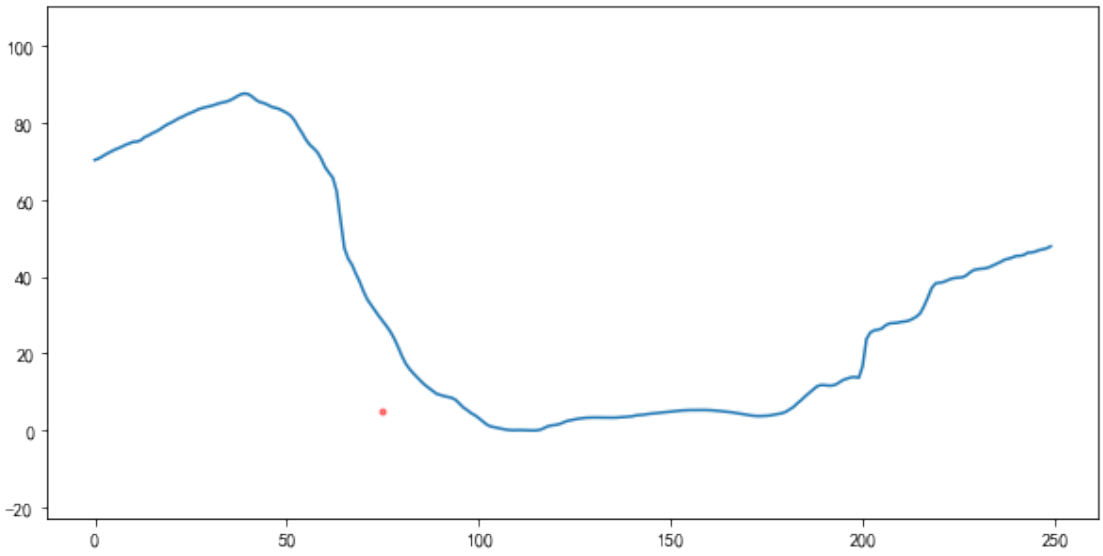
1

0.2.4

0.2.5

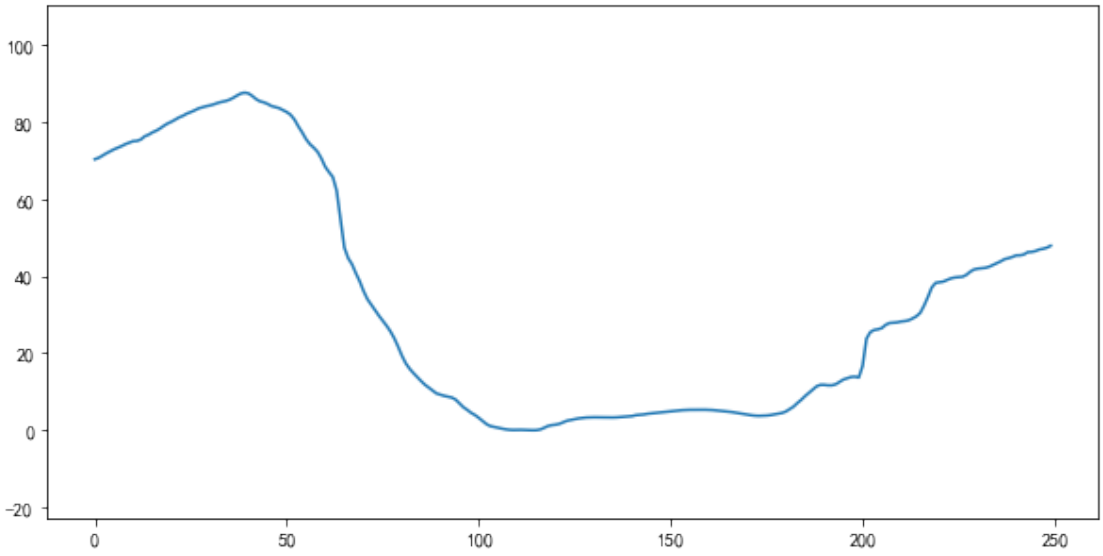
```
In [11]: coor = np.array(train_set)[num_ls[num_ls>1],:]

         plt.figure(figsize = (10,5))
         plt.plot(coast_x,coast)
         #plt.plot(crossx_ls,crossy_ls,'r.')
         plt.plot(coor[:,0],coor[:,1],'r.',alpha = 0.5)
         plt.axis('equal')
         #plt.xlim(0,200)
         #plt.ylim(-20,200)
         plt.show()
```



In [12]:
```
coor = np.array(train_set)[num_ls[num_ls>1],:]

coor,_ = filt_points(coor,distance = 'euc',factor = 1)
plt.figure(figsize = (10,5))
plt.plot(coast_x,coast)
#plt.plot(crossx_ls,crossy_ls,'r.')
plt.plot(coor[:,0],coor[:,1],'r.',alpha = 0.5)
plt.axis('equal')
#plt.xlim(0,200)
#plt.ylim(-20,200)
plt.show()
```



```
In [21]: coor

Out[21]: array([[  89.,   11.],
                [  89.,   11.],
                [  89.,   11.],
                ...,
                [  69.,   38.],
                [-415.,  153.],
                [  89.,   11.]])
```

0.2.6 AP

0.2.7 AP

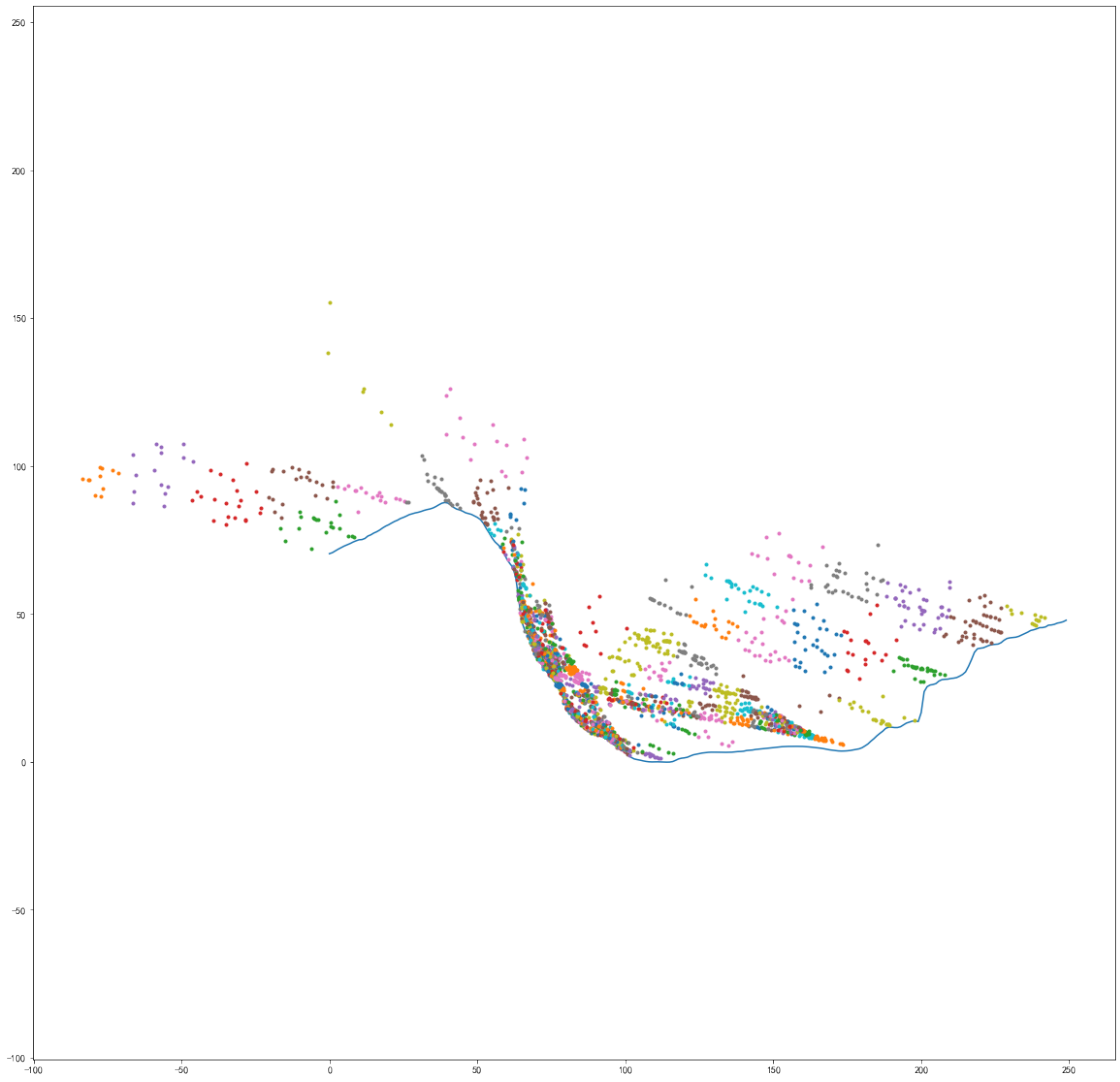
```
In [38]: from sklearn.cluster import AffinityPropagation

         af = AffinityPropagation()
         af.fit(train_set)
         train_res = af.predict(train_set)

In [42]: len(set(train_res))

Out[42]: 1948

In [58]: plt.figure(figsize = (20,20))
         plt.rcParams['font.sans-serif'] = ['KaiTi']
         plt.rcParams['axes.unicode_minus']=False
         plt.plot(coast_x,coast)
         #plt.plot(train_set[:,0],train_set[:,1],'r.',alpha = 0.5)
         plt.axis('equal')
         for i in np.arange(0,np.unique(train_res).size):
             plt.plot(train_set[train_res==i,0],train_set[train_res==i,1],'.')
         plt.show()
```

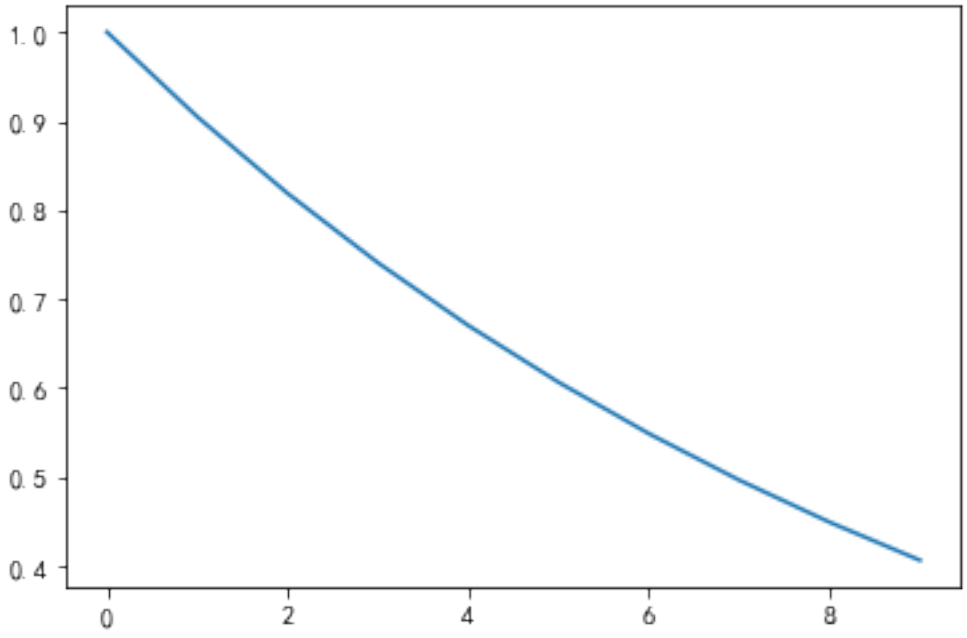


0.3

$$A\sqrt{e^{-\beta d}}$$

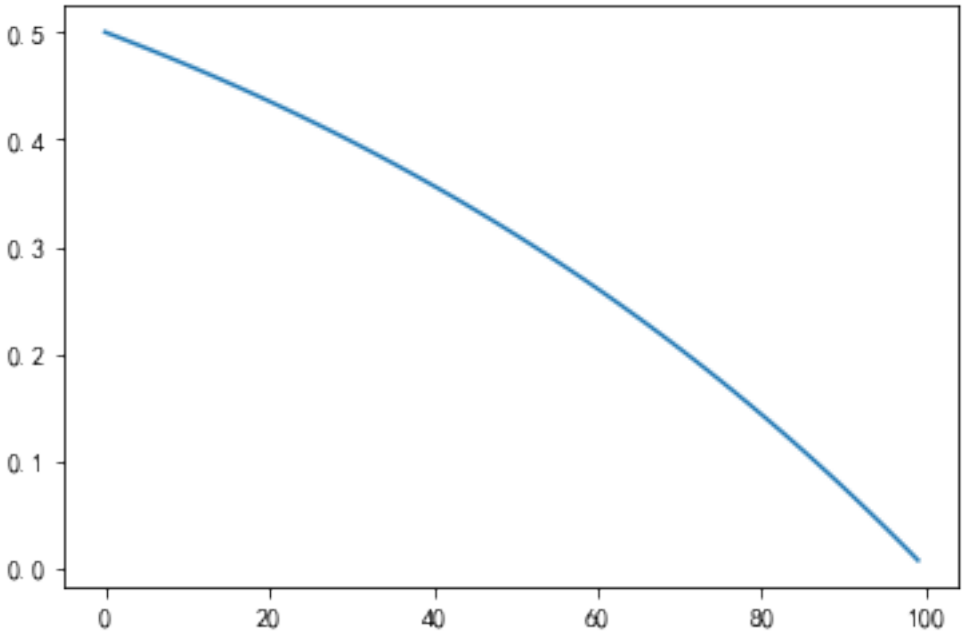
...

```
In [10]: xxx = np.arange(10)
         yyy = np.sqrt(np.exp(-0.2*xxx))
         plt.plot(xxx,yyy)
         plt.show()
```



0.3.1

```
In [57]: d = 100
         a = 0.01
         b = 0.5/(-1+np.exp(a*d))
         c = 0.5+b
         xxx = np.arange(100)
         yyy = -b*np.exp(a*xxx)+c
         plt.plot(xxx,yyy)
         plt.show()
```




```
In [58]: a,b,c,d
```

```
Out[58]: (0.01, 0.2909883534346632, 0.7909883534346632, 100)
```

```
In [ ]:
```