5.
(d)
In[85]:
score_dict
Out[85]:
{0.10000000000000001: 0.63428571428571423,
 0.20000000000000001: 0.7142857142857143,
 0.29999999999999999: 0.72571428571428576,
 0.40000000000000002: 0.7371428571428571,
 0.5: 0.73142857142857143,
 0.59999999999999998: 0.73142857142857143,
 0.69999999999999996: 0.7371428571428571,
 0.79999999999999993: 0.72571428571428576,
 0.89999999999999991: 0.74285714285714288,
 0.99999999999999989: 0.7371428571428571,
 1.0999999999999999: 0.7371428571428571,
 1.2: 0.7371428571428571,
 1.3: 0.74285714285714288,
 1.3999999999999999: 0.74857142857142855,
 1.5: 0.74857142857142855,
 1.5999999999999999: 0.74857142857142855,
 1.7: 0.75428571428571434,
 1.8: 0.75428571428571434,
 1.8999999999999999: 0.75428571428571434,
 2.0: 0.75428571428571434}

The best C is 1.7, 1.8, 1.9, 2.0.
The best accuracy is 0.77238805970149249.

The classification accuracy of hard margin SVM is 0.735074626866.

I believe that the best accuracy of soft margin is bigger than hard margin because soft margin can ignore some outliers, but the hard margin train all the noise.

And the code I implement shows below:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Nov  7 17:29:49 2017

@author: liuchangbai
```

```
"""


import pandas as pd
import numpy as np
import os,random

os.chdir("/Users/liuchangbai/Desktop/courses/Machine-Learning/Homework/HW3_export")

data = pd.read_csv("diabetes_scale.csv", sep = ",", names = ['label', 'feature1',
'feature2','feature3',
                                        'feature4','feature5','feature6','feature7','feature8'])

test = data[500:768]
data = data[0:500]

# cross validation
y = data['label']
x = data[['feature1', 'feature2','feature3','feature4','feature5','feature6','feature7','feature8']]

y_final = test['label']
x_final = test[['feature1',
'feature2','feature3','feature4','feature5','feature6','feature7','feature8']]

from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y,test_size=0.35, random_state=42)


# C value
c_list = np.linspace(0.1, 2, 20)
score_dict = {}

for c_value in c_list:
    # Support Vector Machine
    from sklearn import svm
    c_value = 2.0
    clf = svm.SVC(C = c_value)

    # fit
    clf.fit(x_train, y_train)
    y_pred = clf.predict(x_test)

    # get prediction score
    from sklearn import metrics
```

```
    score = metrics.accuracy_score(y_test,y_pred)
    print(score)

    score_dict[c_value] = score


c_value = 1.7
clf = svm.SVC(C = c_value)

y_predict = clf.predict(x_final)
soft_score = metrics.accuracy_score(y_final, y_predict)



# Hard Margin
hdm = svm.SVC(C = 1* np.exp(6))
hdm.fit(x_train, y_train)

y_pred = hdm.predict(x_final)

# get prediction score

print(metrics.accuracy_score(y_final,y_pred))
```