

## 2 Наивный байесовский классификатор

20% баллов за задание, оценочное время выполнения: 40 минут

Начало 00:36

Конец 01:25

Загрузите датасеты `digits` и `breast_cancer` из `sklearn.datasets`. Выведите несколько строчек из обучающих выборок и посмотрите на признаки. С помощью `sklearn.model_selection.cross_val_score` с настройками по умолчанию и вызова метода `mean()` у возвращаемого этой функцией `numpy.ndarray`, сравните качество работы наивных байесовских классификаторов на этих двух датасетах.

Для сравнения предлагается использовать `BernoulliNB`, `MultinomialNB` и `GaussianNB`. Насколько полученные результаты согласуются с вашими ожиданиями?

Два датасета, конечно, еще не повод делать далеко идущие выводы, но при желании вы можете продолжить исследование на других выборках (например, из UCI репозитория).

Ответьте (прямо в `ipynb` блокноте с вашими экспериментами) на вопросы:

1. Каким получилось максимальное качество классификации на датасете `breast_cancer`?
2. Каким получилось максимальное качество классификации на датасете `digits`?
3. Какие утверждения из приведенных ниже верны?

(a) На вещественных признаках лучше всего сработал наивный байесовский классификатор с распределением Бернулли

(b) На вещественных признаках лучше всего сработал наивный байесовский классификатор с мультиномиальным распределением

(c) Мультиномиальное распределение лучше показало себя на выборке с целыми неотрицательными значениями признаков

(d) На вещественных признаках лучше всего сработало нормальное распределение

In [12]:

```
import numpy as np
import matplotlib.pyplot as plt

from sklearn import cross_validation, datasets, metrics, neighbors
from sklearn.model_selection import cross_val_score
from sklearn.naive_bayes import GaussianNB, BernoulliNB, MultinomialNB
```

In [13]:

```
digits = datasets.load_digits()
```

In [14]:

```
digits.keys()
```

Out[14]:

```
['images', 'data', 'target_names', 'DESCR', 'target']
```

In [15]:

```
print len(digits.images)
print "images: {}".format(digits.images[:2])
print "data: {}".format(digits.data[:2])
print "target names: {}".format(names = digits.target_names)
```

1797

```
images: [[[ 0.  0.  5. 13.  9.  1.  0.  0.]
```

```
[ 0.  0. 13. 15. 10. 15.  5.  0.]
```

```
[ 0.  3. 15.  2.  0. 11.  8.  0.]
```

```
[ 0.  4. 12.  0.  0.  8.  8.  0.]
```

```
[ 0.  5.  8.  0.  0.  9.  8.  0.]
```

```
[ 0.  4. 11.  0.  1. 12.  7.  0.]
```

```
[ 0.  2. 14.  5. 10. 12.  0.  0.]
```

```
[ 0.  0.  6. 13. 10.  0.  0.  0.]]
```

```
[[ 0.  0.  0. 12. 13.  5.  0.  0.]
```

```
[ 0.  0.  0. 11. 16.  9.  0.  0.]
```

```
[ 0.  0.  3. 15. 16.  6.  0.  0.]
```

```
[ 0.  7. 15. 16. 16.  2.  0.  0.]
```

```
[ 0.  0.  1. 16. 16.  3.  0.  0.]
```

```
[ 0.  0.  1. 16. 16.  6.  0.  0.]
```

```
[ 0.  0.  1. 16. 16.  6.  0.  0.]
```

```
[ 0.  0.  0. 11. 16. 10.  0.  0.]]]
```

```
data: [[ 0.  0.  5. 13.  9.  1.  0.  0.  0.  0. 13. 15.
```

```
10. 15.
```

```
 5.  0.  0.  3. 15.  2.  0. 11.  8.  0.  0.  4. 12.
```

```
 0.
```

```
 0.  8.  8.  0.  0.  5.  8.  0.  0.  9.  8.  0.  0.
```

```
 4.
```

```
11.  0.  1. 12.  7.  0.  0.  2. 14.  5. 10. 12.  0.
```

```
 0.
```

```
 0.  0.  6. 13. 10.  0.  0.  0.]
```

```
[ 0.  0.  0. 12. 13.  5.  0.  0.  0.  0.  0. 11. 16.
```

```
 9.
```

```
 0.  0.  0.  0.  3. 15. 16.  6.  0.  0.  0.  7. 15.
```

16.

```
16.  2.  0.  0.  0.  0.  1. 16. 16.  3.  0.  0.  0.
```

```
 0.
```

```
 1. 16. 16.  6.  0.  0.  0.  0.  1. 16. 16.  6.  0.
```

```
 0.
```

```
 0.  0.  0. 11. 16. 10.  0.  0.]]]
```

```
target names: [0 1 2 3 4 5 6 7 8 9]
```

In [16]:

```
breast_cancer = datasets.load_breast_cancer()
```

In [17]:

```
breast_cancer.keys()
```

Out[17]:

```
['target_names', 'data', 'target', 'DESCR', 'feature_names']
```

In [18]:

```
print "feature names: {}".format(breast_cancer.feature_names[:2])
print "data: {}".format(breast_cancer.data[:2])
print "target names: {names}".format(names = breast_cancer.target_names)
```

```
feature names: ['mean radius' 'mean texture']
```

```
data: [[ 1.79900000e+01  1.03800000e+01  1.22800000e+02  1.0010
0000e+03
```

```
1.18400000e-01  2.77600000e-01  3.00100000e-01  1.47100000e-
```

```
01
```

```
2.41900000e-01  7.87100000e-02  1.09500000e+00  9.05300000e-
```

```
01
```

```
8.58900000e+00  1.53400000e+02  6.39900000e-03  4.90400000e-
```

```
02
```

```
5.37300000e-02  1.58700000e-02  3.00300000e-02  6.19300000e-
```

```
03
```

```
2.53800000e+01  1.73300000e+01  1.84600000e+02  2.01900000e+
```

```
03
```

```
1.62200000e-01  6.65600000e-01  7.11900000e-01  2.65400000e-
```

```
01
```

```
4.60100000e-01  1.18900000e-01]
```

```
[ 2.05700000e+01  1.77700000e+01  1.32900000e+02  1.32600000e+
```

```
03
```

```
8.47400000e-02  7.86400000e-02  8.69000000e-02  7.01700000e-
```

```
02
```

```
1.81200000e-01  5.66700000e-02  5.43500000e-01  7.33900000e-
```

```
01
```

```
3.39800000e+00  7.40800000e+01  5.22500000e-03  1.30800000e-
```

```
02
```

```
1.86000000e-02  1.34000000e-02  1.38900000e-02  3.53200000e-
```

```
03
```

```
2.49900000e+01  2.34100000e+01  1.58800000e+02  1.95600000e+
```

```
03
```

```
1.23800000e-01  1.86600000e-01  2.41600000e-01  1.86000000e-
```

```
01
```

```
2.75000000e-01  8.90200000e-02]]
```

```
target names: ['malignant' 'benign']
```

In [19]:

```
#С помощью sklearn.model_selection.cross_val_score с настройками по умолчанию  
#и вызова метода mean() у возвращаемого этой функцией numpy.ndarray,  
#сравните качество работы наивных байесовских классификаторов на этих двух датас  
етах.  
#Для сравнения предлагается использовать BernoulliNB, MultinomialNB и GaussianNB  
B.  
  
models = [BernoulliNB(), MultinomialNB(), GaussianNB()]  
names_models = ['BernoulliNB', 'MultinomialNB', 'GaussianNB']  
  
print 'Digits\n'  
for i in range(3):  
    print('{} {:.3f}'.format(names_models[i], np.mean(cross_val_score(models[i],  
digits.data, digits.target))))  
  
print '\n'  
print 'Breast cancer\n'  
for i in range(3):  
    print('{} {:.3f}'.format(names_models[i], np.mean(cross_val_score(models[i],  
breast_cancer.data, breast_cancer.target))))
```

Digits

BernoulliNB 0.826  
MultinomialNB 0.871  
GaussianNB 0.819

Breast cancer

BernoulliNB 0.627  
MultinomialNB 0.895  
GaussianNB 0.937

### 1 . Каким получилось максимальное качество классификации на датасете breast\_

GaussianNB 0.937

### 2 . Каким получилось максимальное качество классификации на датасете digits?

MultinomialNB 0.871

### 3 . Какие утверждения из приведенных ниже верны?

(с) Мультиномиальное распределение лучше показало себя на выборке с целыми неотрицательными значениями признаков

Выборка Digits содержит дискретные значения признаков. Мультиномиальное распределение дискретно - при его выборе достигается наибольшая точность.

(d) На вещественных признаках лучше всего сработало нормальное распределение

Выборка Breast cancer содержит вещественные признаки. Нормальное распределение непрерывно - при его выборе достигается наибольшая точность.