

РОБОТА із ЗОБРАЖЕННЯМИ

Файл: Image_10_001

Класифікація зображень. Машина опорних векторів - Support Vector Machine (SVM)

Приклад класифікації облич

Дивись [Support Vector Machines](#)

```
# Завантаження пакетів
import numpy as np
import skimage.io as io
import matplotlib.pyplot as plt
plt.rcParams['font.size'] = 3

# Для статистичної обробки
from scipy import ndimage
from scipy import stats
```

```
# Завантаження пакету [Seaborn]
import seaborn as sns
sns.set()
```

```
# З sklearn імпорт лінійної регресії
from sklearn.linear_model import LinearRegression
from sklearn.datasets import make_blobs
```

Завантаження набору фотографій

```
# Приклади фото обличчь
from sklearn.datasets import fetch_lfw_people
faces = fetch_lfw_people(min_faces_per_person=60)
print('Всього Облич', faces.target_names.shape[0])
print(faces.target_names)
print('Всього фото', faces.images.shape[0])
print('Размір фото', faces.images.shape[1], ' x ', faces.images.shape[2])
```

```
Всього Облич 8
['Ariel Sharon' 'Colin Powell' 'Donald Rumsfeld' 'George W Bush'
 'Gerhard Schroeder' 'Hugo Chavez' 'Junichiro Koizumi' 'Tony Blair']
Всього фото 1348
Размір фото 62 x 47
```

!!! Маємо 1348 фото. Кожне фото по 2914 піксела (вектор 2914 компонент !)

```
# Як приклад - перші 15 фото
fig, ax = plt.subplots(3, 5)
for i, axi in enumerate(ax.flat):
    axi.imshow(faces.images[i], cmap='bone')
    # axi.set(xticks=[], yticks=[])
    axi.set(xticks=[], yticks=[],
            xlabel=faces.target_names[faces.target[i]])
```



Colin Powell



George W Bush



George W Bush



George W Bush



Hugo Chavez



George W Bush



Shinichi Koizumi



George W Bush



Tony Blair



Ariel Sharon



George W Bush



Donald Rumsfeld



George W Bush



George W Bush



George W Bus

```
# дaтaсeт пoдiляeмo нa тpeнyвaльнi тa вaлiдaцiйнi пocлiдoвнocтi
from sklearn.model_selection import train_test_split
Xtrain, Xtest, ytrain, ytest = train_test_split(faces.data, faces.target,
                                              random_state=42)
print('Тpeнyвaльнa мнoжинa', Xtrain.shape, ytrain.shape)
print('Вaлiдaцiйнa мнoжинa', Xtest.shape, ytest.shape)
```

Тренувальна множина (1011, 2914) (1011,)
Валідаційна множина (337, 2914) (337,)

!! Створення дескриптора

```
# Імпортуємо функцію формування дескрипторів (векторів)
# Використуємо метод головних компонент (PCA) для вилучення признаков
from sklearn.decomposition import PCA # імпорт методу PCA

# Будемо вилучати 150 признаков з 2914 пікселів кожної фотографії
# Тільки для демонстрації
pca = PCA(n_components=150, whiten=True, random_state=42) # створення екземпляру (моделі) PCA
pca.fit(Xtrain) # Обробляємо тренувальну множину
# Отримали 1011 векторів по 150 компонент кожний
print('кількість зразків для тренування', pca.n_samples_)
print('кількість компонент в векторі', pca.n_components_)
```

```
кількість зразків для тренування 1011
кількість компонент в векторі 150
```

Навчання SVM класифікатора

```
from sklearn.svm import SVC # "Support vector classifier"
```

```
# СОЗДАЕМ КОНВЕЙЕР
```

```
from sklearn.pipeline import make_pipeline
svc = SVC(kernel='rbf', class_weight='balanced') # модель класифікатора
model = make_pipeline(pca, svc) # конвейєр -> PCA -> SVM
```

```
# Навчання моделі
from sklearn.model_selection import GridSearchCV
param_grid = {'svc__C': [1, 5, 10, 50],
              'svc__gamma': [0.0001, 0.0005, 0.001, 0.005]}
# Пошук оптимальних параметрів на сітці
grid = GridSearchCV(model, param_grid)
grid.fit(Xtrain, ytrain)
print(grid.best_params_)
```

```
{'svc__C': 5, 'svc__gamma': 0.001}
```

Використуємо тесту множину

```
# Передаємо модлі знайдені оптимальні параметри
model = grid.best_estimator_
# Визначаємо мітки для тестової множини
yfit = model.predict(Xtest)
```

```
fig, ax = plt.subplots(4, 6)
for i, axi in enumerate(ax.flat):
    axi.imshow(Xtest[i].reshape(62, 47), cmap='bone')
    axi.set(xticks=[], yticks=[])
    axi.set_ylabel(faces.target_names[yfit[i]].split()[-1],
                   color='black' if yfit[i] == ytest[i] else 'red')
fig.suptitle('Визначено; Невірні - червоні', size=12);
```

Визначено; Невірні - червоні



```
# Метрики
from sklearn.metrics import classification_report
print(classification_report(ytest, yfit,
                           target_names=faces.target_names))
```

	precision	recall	f1-score	support
Ariel Sharon	0.65	0.87	0.74	15
Colin Powell	0.83	0.88	0.86	68
Donald Rumsfeld	0.70	0.84	0.76	31
George W Bush	0.97	0.80	0.88	126
Gerhard Schroeder	0.76	0.83	0.79	23
Hugo Chavez	0.93	0.70	0.80	20
Junichiro Koizumi	0.86	1.00	0.92	12
Tony Blair	0.82	0.98	0.89	42
accuracy			0.85	337
macro avg	0.82	0.86	0.83	337

weighted avg	0.86	0.85	0.85	337
--------------	------	------	------	-----

Матриця плутанини

```
from sklearn.metrics import confusion_matrix
mat = confusion_matrix(ytest, yfit)
sns.heatmap(mat.T, square=True, annot=True, fmt='d', cbar=True,
            xticklabels=faces.target_names,
            yticklabels=faces.target_names)
plt.xlabel('true label')
plt.ylabel('predicted label');
```

