十分钟上手sklearn：安装，获取数据，数据预处理

2018-02-05 | 机器学习 |
1,368 | 6

sklearn是机器学习中一个常用的python第三方模块，对常用的机器学习算法进行了封装

其中包括：

1.分类（Classification）

2.回归（Regression）

3.聚类（Clustering）

4.数据降维（Dimensionality reduction）

5.常用模型（Model selection）

6.数据预处理（Preprocessing）

本文将从sklearn的安装开始讲解，由浅入深，逐步上手sklearn。

sklearn官网：http://scikit-learn.org/stable/index.html

sklearn API：http://scikit-learn.org/stable/modules/classes.html#module-sklearn.preprocessing

skleran安装

sklearn的目前版本是0.19.1

依赖包：

Python (>=2.6或>=3.3)

NumPy(>=1.6.1)

SciPy(>=0.9)

使用pip安装，terminal直接执行即可

```
pip install -U scikit-learn
```

使用Anaconda安装，推荐Anaconda，因为里面已经内置了NumPy，SciPy等常用工具

```
conda install scikit-learn
```

安装完成后可以在python中检查一下版本，import sklearn不报错，则表示安装成功

```
>>import sklearn
>>sklearn.__version__
'0.19.1'
```

获取数据

机器学习算法往往需要大量的数据，在skleran中获取数据通常采用两种方式，一种是使用自带的数据集，另一种是创建数据集

导入数据集

sklearn自带了很多数据集，可以用来对算法进行测试分析，免去了自己再去找数据集的烦恼

其中包括：

鸢尾花数据集:load_iris()

手写数字数据集:load_digitals()

糖尿病数据集:load_diabetes()

乳腺癌数据集:load_breast_cancer()

波士顿房价数据集:load_boston()

体能训练数据集:load_linnerud()

这里以鸢尾花数据集为例导入数据集

```
#导入sklearn的数据集
import sklearn.datasets as sk_datasets
iris = sk_datasets.load_iris()
iris_X = iris.data #导入数据
iris_y = iris.target #导入标签
```

创建数据集

使用skleran的样本生成器(samples generator)可以创建数据，sklearn.datasets.samples_generator中包含了大量创建样本数据的方法。

这里以分类问题创建样本数据

```
import sklearn.datasets.samples_generator as sk_sample_generator
X,y=sk_sample_generator.make_classification(n_samples=6,n_features=5,n_informative=2,n_redundant=3,n_classes=2,n_c
for x_,y_ in zip(X,y):
    print(y_,end=": ")
    print(x_)
```

参数说明：

n_features :特征个数= n_informative（ ） + n_redundant + n_repeated

n_informative：多信息特征的个数

n_redundant：冗余信息，informative特征的随机线性组合

n_repeated ：重复信息，随机提取n_informative和n_redundant 特征

n_classes：分类类别

n_clusters_per_class ：某一个类别是由几个cluster构成的

random_state：随机种子，使得实验可重复

n_classes*n_clusters_per_class 要小于或等于 2^n_informative

打印结果：

```
0: [ 0.64459602  0.92767918 -1.32091378 -1.25725859 -0.74386837]
0: [ 1.66098845  2.22206181 -2.86249859 -3.28323172 -1.62389676]
0: [ 0.27019475 -0.12572907  1.1003977  -0.6600737   0.58334745]
1: [-0.77182836 -1.03692724  1.34422289  1.52452016  0.76221055]
1: [-0.1407289   0.32675611 -1.41296696  0.4113583  -0.75833145]
1: [-0.76656634 -0.35589955 -0.83132182  1.68841011 -0.4153836 ]
```

数据集的划分

机器学习的过程正往往需要对数据集进行划分，常分为训练集，测试集。sklearn中的model_selection为我们提供了划分数据集的方法。

以鸢尾花数据集为例进行划分

```
import sklearn.model_selection as sk_model_selection
X_train,X_test,y_train,y_test = sk_model_selection.train_test_split(iris_X,iris_y,train_size=0.3,random_state=20)
```

参数说明：

arrays：样本数组，包含特征向量和标签

test_size：

float-获得多大比重的测试样本 （默认：0.25）

int - 获得多少个测试样本

train_size: 同test_size

random_state:int - 随机种子（种子固定，实验可复现）

shuffle - 是否在分割之前对数据进行洗牌（默认True）

后面我们训练模型使用的数据集都基于此

数据预处理

我们为什么要进行数据预处理？

通常，真实生活中，我们获得的数据中往往存在很多的无用信息，甚至存在错误信息，而机器学习中有一句话叫做”Garbage in，Garbage out”，数据的健康程度对于算法结果的影响极大。数据预处理就是让那些冗余混乱的源数据变得能满足其应用要求。

当然，仅仅是数据预处理的方法就可以写好几千字的文章了，在这里只谈及几个基础的数据预处理的方法。

skleran中为我们提供了一个数据预处理的package：preprocessing，我们直接导入即可

```
import sklearn.preprocessing as sk_preprocessing
```

下面的例子我们使用:[[1, -1, 2], [0, 2, -1], [0, 1, -2]]做为初始数据。

数据的归一化

基于mean和std的标准化

```
scaler = sk_preprocessing.StandardScaler().fit(X)
new_X = scaler.transform(X)
print('基于mean和std的标准化:',new_X)
```

打印结果:

```
基于mean和std的标准化:
 [[ 1.41421356 -1.33630621  1.37281295]
 [-0.70710678  1.06904497 -0.39223227]
 [-0.70710678  0.26726124 -0.98058068]]
```

规范化到一定区间内 feature_range为数据规范化的范围

```
scaler = sk_preprocessing.MinMaxScaler(feature_range=(0,1)).fit(X)
```

```
new_X=scaler.transform(X)
print('规范化到一定区间内',new_X)
```

打印结果:

```
规范化到一定区间内
 [[1.         0.         1.        ]
 [ 0.         1.         0.25      ]
 [ 0.         0.66666667 0.        ]]
```

数据的正则化

首先求出样本的p-范数，然后该样本的所有元素都要除以该范数，这样最终使得每个样本的范数都为1

```
new_X = sk_preprocessing.normalize(X,norm='l2')
print('求二范数',new_X)
```

打印结果：

```
规范化到一定区间内
 [[0.40824829 -0.40824829  0.81649658]
 [ 0.         0.89442719 -0.4472136 ]
 [ 0.         0.4472136  -0.89442719]]
```

小结

本文介绍了sklearn的安装，sklearn导入数据集，创建数据集的基本方法，对数据预处理的常用方法进行了介绍。

下一篇，将重点讲解如何使用sklearn进行特征提取，使用sklearn实现机器学习经典算法，模型的保存等内容。

sklearn

‹ 决策树——ID3算法实现

十分钟上手sklearn：特征提取，常用模型，交叉验证 ›