基于 Python 和 Scikit-Learn 的机器学习介绍

2015/07/21 · 工具与框架 · Python, Scikit-Learn, 机器学习

你好，%用户名%！

我叫Alex，我在机器学习和网络图分析（主要是理论）有所涉猎。我同时在为一家俄罗斯移动运营商开发大数据产品。这是我第一次在网上写文章，不喜勿喷。

机构中领导了媒体和社交网络大数据分析工具的开发。我们然有一些我团队使用过的文档，我乐意与你们分享。前提是读者已经有很好的数学和机器学习方面的知识（我的团队主要由MIPT（莫斯科物理与技术大学）和数据分析学院的毕业生构成）。

这篇文章是对数据科学的简介，这门学科最近太火了。机器学习的竞赛也越来越多（如，Kaggle, TudedIT），而且他们的资金通常很可观。

R和Python是提供给数据科学家的最常用的两种工具。每一个工具都有其优缺点，但Python最近在各个方面都有所胜出（仅为鄙人愚见，虽然我两者都用）。这一切的发生是因为Scikit-Learn库的腾空出世，它包含有完善的文档和丰富的机器学习算法。

请注意，我们将主要在这篇文章中探讨机器学习算法。通常用Pandas包去进行主数据分析会比较好，而且这很容易你自己完成。所以，让我们集中精力在实现上。为了确定性，我们假设有一个特征-对象矩阵作为输入，被存在一个*.csv文件中。

数据加载

首先，数据要被加载到内存中，才能对其操作。Scikit-Learn库在它的实现用使用了NumPy数组，所以我们将用NumPy来加载*.csv文件。让我们从UCI Machine Learning Repository下载其中一个数据集。

```python
import numpy as np
import urllib
# url with dataset
url = "http://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.data"
# download the file
raw_data = urllib.urlopen(url)
# load the CSV file as a numpy matrix
dataset = np.loadtxt(raw_data, delimiter=",")
# separate the data from the target attributes
X = dataset[:,0:7]
y = dataset[:,8]
```

我们将在下面所有的例子里使用这个数据组，换言之，使用X特征物数组和y目标变量的值。

数据标准化

我们都知道大多数的梯度方法（几乎所有的机器学习算法都基于此）对于数据的缩放很敏感。因此，在运行算法之前，我们应该进行标准化，或所谓的规格化。标准化包括替换所有特征的名义值，让它们每一个的值在0和1之间。而对于规格化，它包括数据的预处理，使得每个特征的值有0和1

Python

```
from sklearn import preprocessing
# normalize the data attributes
normalized_X = preprocessing.normalize(X)
# standardize the data attributes
standardized_X = preprocessing.scale(X)
```

特征的选取

毫无疑问，解决一个问题最重要的是是恰当选取特征、甚至创造特征的能力。这叫做特征选取和特征工程。虽然特征工程是一个相当有创造性的过程，有时候更多的是靠直觉和专业的知识，但对于特征的选取，已经有很多的算法可供直接使用。如树算法就可以计算特征的信息量。

Python

```
from sklearn import metrics
from sklearn.ensemble import ExtraTreesClassifier
model = ExtraTreesClassifier()
model.fit(X, y)
# display the relative importance of each attribute
print(model.feature_importances_)
```

其他所有的方法都是基于对特征子集的高效搜索，从而找到最好的子集，意味着演化了的模型在这个子集上有最好的质量。递归特征消除算法（RFE）是这些搜索算法的其中之一，Scikit-Learn库同样也有提供。

Python

```
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
# create the RFE model and select 3 attributes
rfe = RFE(model, 3)
rfe = rfe.fit(X, y)
# summarize the selection of the attributes
print(rfe.support_)
print(rfe.ranking_)
```

正像我说的，Scikit-Learn库已经实现了所有基本机器学习的算法。让我来瞧一瞧它们中的一些。

逻辑回归

大多数情况下被用来解决分类问题（二元分类），但多类的分类（所谓的一对多方法）也适用。这个算法的优点是对于每一个输出的对象都有一个对应类别的概率。

Python

```python
from sklearn import metrics
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X, y)
print(model)
# make predictions
expected = y
predicted = model.predict(X)
# summarize the fit of the model
print(metrics.classification_report(expected, predicted))
print(metrics.confusion_matrix(expected, predicted))
```

朴素贝叶斯

它也是最有名的机器学习的算法之一，它的主要任务是恢复训练样本的数据分布密度。这个方法通常在多类的分类问题上表现的很好。

Python

```python
from sklearn import metrics
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X, y)
print(model)
# make predictions
expected = y
predicted = model.predict(X)
# summarize the fit of the model
print(metrics.classification_report(expected, predicted))
print(metrics.confusion_matrix(expected, predicted))
```

kNN（k-最近邻）方法通常用于一个更复杂分类算法的一部分。例如，我们可以用它的估计值做为一个对象的特征。有时候，一个简单的kNN算法在良好选择的特征上会有很出色的表现。当参数（主要是metrics）被设置得当，这个算法在回归问题中通常表现出最好的质量。

Python

```
from sklearn import metrics
from sklearn.neighbors import KNeighborsClassifier
# fit a k-nearest neighbor model to the data
model = KNeighborsClassifier()
model.fit(X, y)
print(model)
# make predictions
expected = y
predicted = model.predict(X)
# summarize the fit of the model
print(metrics.classification_report(expected, predicted))
print(metrics.confusion_matrix(expected, predicted))
```

决策树

分类和回归树（CART）经常被用于这么一类问题，在这类问题中对象有可分类的特征且被用于回归和分类问题。决策树很适用于多类分类。

Python

```
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier
# fit a CART model to the data
model = DecisionTreeClassifier()
model.fit(X, y)
print(model)
# make predictions
expected = y
predicted = model.predict(X)
# summarize the fit of the model
print(metrics.classification_report(expected, predicted))
print(metrics.confusion_matrix(expected, predicted))
```

支持向量机

```python
from sklearn import metrics
from sklearn.svm import SVC
# fit a SVM model to the data
model = SVC()
model.fit(X, y)
print(model)
# make predictions
expected = y
predicted = model.predict(X)
# summarize the fit of the model
print(metrics.classification_report(expected, predicted))
print(metrics.confusion_matrix(expected, predicted))
```

除了分类和回归问题，Scikit-Learn还有海量的更复杂的算法，包括了聚类， 以及建立混合算法的实现技术，如Bagging和Boosting。

如何优化算法的参数

在编写高效的算法的过程中最难的步骤之一就是正确参数的选择。一般来说如果有经验的话会容易些，但无论如何，我们都得寻找。幸运的是Scikit-Learn提供了很多函数来帮助解决这个问题。

作为一个例子，我们来看一下规则化参数的选择，在其中不少数值被相继搜索了：

```python
import numpy as np
from sklearn.linear_model import Ridge
from sklearn.grid_search import GridSearchCV
# prepare a range of alpha values to test
alphas = np.array([1,0.1,0.01,0.001,0.0001,0])
# create and fit a ridge regression model, testing each alpha
model = Ridge()
grid = GridSearchCV(estimator=model, param_grid=dict(alpha=alphas))
grid.fit(X, y)
print(grid)
# summarize the results of the grid search
print(grid.best_score_)
print(grid.best_estimator_.alpha)
```

Python

```python
import numpy as np
from scipy.stats import uniform as sp_rand
from sklearn.linear_model import Ridge
from sklearn.grid_search import RandomizedSearchCV
# prepare a uniform distribution to sample for the alpha parameter
param_grid = {'alpha': sp_rand()}
# create and fit a ridge regression model, testing random alpha values
model = Ridge()
rsearch = RandomizedSearchCV(estimator=model, param_distributions=param_grid, n_iter=100)
rsearch.fit(X, y)
print(rsearch)
# summarize the results of the random parameter search
print(rsearch.best_score_)
print(rsearch.best_estimator_.alpha)
```

至此我们已经看了整个使用Scikit-Learn库的过程，除了将结果再输出到一个文件中。这个就作为你的一个练习吧，和R相比Python的一大优点就是它有很棒的文档说明。

在下一篇文章中，我们将深入探讨其他问题。我们尤其是要触及一个很重要的东西——特征的建造。我真心地希望这份材料可以帮助新手数据科学家尽快开始解决实践中的机器学习问题。最后，我祝愿那些刚刚开始参加机器学习竞赛的朋友拥有耐心以及马到成功！

关于作者：xeonqq

@钱小谦qq

个人主页 · 我的文章 · 10

相关文章

- Ruby 和 Python 分析器是如何工作的？
- 疏而不漏：随机森林 · 1
- Python NLP入门教程 · 5
- 三生万物：决策树 · 2
- 机器学习算法实践-树回归 · 1

可能感兴趣的话题

- 2年Java，想转 python · 2
- 自学为什么不合作共赢呢 · 2
- 计算机专业和培训出来的差别多大 · 7
- 是不是每一个男程序员都不会与女孩子表达与交流呢？ · 14
- 除了网站投简历外，还有什么方式可以拿到公司面试机会 · 5
- 前端未来规划 · 3

Python小组话题 **我有新话题**

Python自学，基础已经学完，现在学...
alexhan 发起 • 45 回复

Python学习，有哪些方向可以选择
小丑的哭笑 发起 • 7 回复

有没有非互联网行业的小伙伴自学编程...
叫我小K咯 发起 • 276 回复

在Python中，如何定义一个函数，来...
泡泡泡 发起 • 19 回复

[大概会吧](#) 发起 • 2 回复

[数据挖掘之面向对象的姑娘们都想要什...](#)
[小小小糖](#) 发起 • 15 回复

- [本周热门Python文章](#)
- [本月热门](#)
- [热门标签](#)

0 [Python Django 性能测试与优化指南](#)

1 [Ruby 和 Python 分析器是如何工...](#)

Python工具资源 **更多资源 »**

Tryton：一个通用商务框架
杂项

NLTK：一个先进的用来处理自然语言数据的Python程序。
自然语言处理 · 2

PyMC：马尔科夫链蒙特卡洛采样工具
科学计算与分析

statsmodels：统计建模和计量经济学
科学计算与分析

[Pylearn2：一个基于Theano的机器学习库](#)
[机器学习](#) · [1](#)

关于 Python 频道

Python频道分享 Python 开发技术、相关的行业动态。

快速链接
网站使用指南 »
加入我们 »
问题反馈与求助 »
网站积分规则 »
网站声望规则 »

关注我们

新浪微博：@Python开发者
RSS：订阅地址
推荐微信号

QQ： 2302462408（加好友请注明来意）

更多频道

小组 – 好的话题、有启发的回复、值得信赖的圈子
头条 – 分享和发现有价值的内容与观点
相亲 – 为IT单身男女服务的征婚传播平台
资源 – 优秀的工具资源导航
翻译 – 翻译传播优秀的外文文章
文章 – 国内外的精选文章
设计 – UI,网页，交互和用户体验
iOS – 专注iOS技术分享
安卓 – 专注Android技术分享
前端 – JavaScript, HTML5, CSS
Java – 专注Java技术分享
Python – 专注Python技术分享