【python数据挖掘课程】二十五.Matplotlib绘制带主题及聚类类标的散点图

这是《Python数据挖掘课程》系列文章，希望对您有所 帮助。当我们做聚类分析绘制散点图时，通常会遇到无法区分散点类标的情况，做主题分析时，可能会遇到无法将对应散点的名称（尤其中文名称）添加至图型中，为了解决这两个问题，本文提出了Matplotlib库的高级应用，主要是绘制带主题的散点图及聚类类标颜色进行区分，该方法被广泛应用于文本聚类和主题分析领域。

本篇文章为基础性文章，希望对你有所帮助，提供些思路，也是自己教学的内容。如果文章中存在错误或不足之处，还请海涵。同时，推荐大家阅读我以前的文章了解其他知识。

前文参考：

【Python数据挖掘课程】一.安装Python及爬虫入门介绍
【Python数据挖掘课程】二.Kmeans聚类数据分析及Anaconda介绍
【Python数据挖掘课程】三.Kmeans聚类代码实现、作业及优化
【Python数据挖掘课程】四.决策树DTC数据分析及鸢尾数据集分析
【Python数据挖掘课程】五.线性回归知识及预测糖尿病实例
【Python数据挖掘课程】六.Numpy、Pandas和Matplotlib包基础知识
【Python数据挖掘课程】七.PCA降维操作及subplot子图绘制
【Python数据挖掘课程】八.关联规则挖掘及Apriori实现购物推荐
【Python数据挖掘课程】九.回归模型LinearRegression简单分析氧化物数据
【python数据挖掘课程】十.Pandas、Matplotlib、PCA绘图实用代码补充
【python数据挖掘课程】十一.Pandas、Matplotlib结合SQL语句可视化分析
【python数据挖掘课程】十二.Pandas、Matplotlib结合SQL语句对比图分析
【python数据挖掘课程】十三.WordCloud词云配置过程及词频分析
【python数据挖掘课程】十四.Scipy调用curve_fit实现曲线拟合
【python数据挖掘课程】十五.Matplotlib调用imshow()函数绘制热图
【python数据挖掘课程】十六.逻辑回归LogisticRegression分析鸢尾花数据
【python数据挖掘课程】十七.社交网络Networkx库分析人物关系（初识篇）
【python数据挖掘课程】十八.线性回归及多项式回归分析四个案例分享

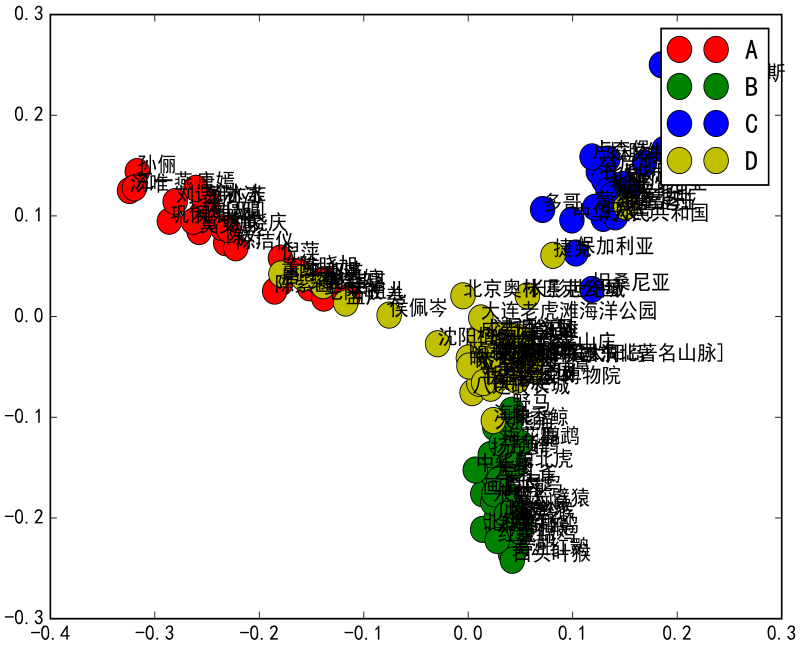
【python数据挖掘课程】十九.鸢尾花数据集可视化、线性回归、决策树花样分析
【python数据挖掘课程】二十.KNN最近邻分类算法分析详解及平衡秤TXT数据集读取
【python数据挖掘课程】二十一.朴素贝叶斯分类器详解及中文文本舆情分析
【python数据挖掘课程】二十二.Basemap地图包安装入门及基础知识讲解
【python数据挖掘课程】二十三.时间序列金融数据预测及Pandas库详解
【python数据挖掘课程】二十四.KMeans文本聚类分析互动百科语料

PSS：最近参加CSDN2018年博客评选，希望您能投出宝贵的一票。我是59号，Eastmount，杨秀璋。投票地址：https://bss.csdn.net/m/topic/blog_star2018/index

五年来写了314篇博客，12个专栏，是真的热爱分享，热爱CSDN这个平台，也想帮助更多的人，专栏包括Python、数据挖掘、网络爬虫、图像处理、C#、Android等。现在也当了两年老师，更是觉得有义务教好每一个学生，让贵州学子好好写点代码，学点技术，"师者，传到授业解惑也"，提前祝大家新年快乐。2019我们携手共进，为爱而生。

一. Matplotlib绘制带主题散点图

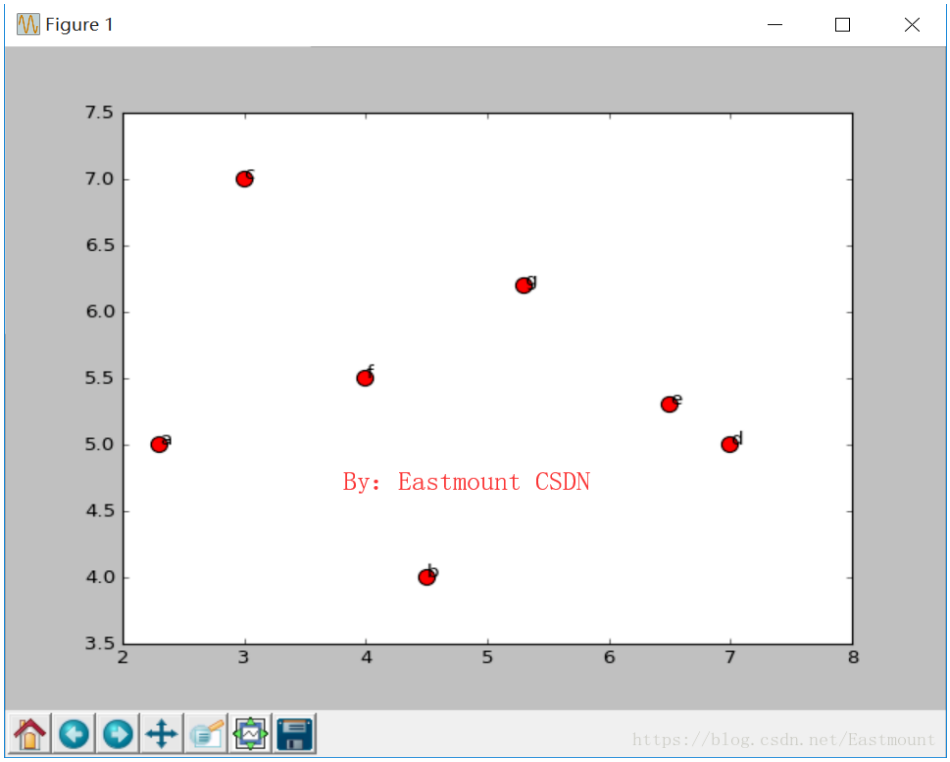
本文能帮助大家实现如下图所示的文本聚类分析或LDA主题模型分析，将相同主题的文章聚集在一起，也可以用于引文分析。图中包括人物、动物、景区和国家四个主题，将相似主题的文本聚集在一起，但也有预测错误的点，比如黄色“侯佩岑”被预测为黄色的景区主题。
文本聚类详见上一篇文章：【python数据挖掘课程】二十四.KMeans文本聚类分析互动百科语料。



详细代码如下所示，通过(x,y)绘制散点图，再调用annotate()函数增加每个点对应的名称。注意：聚类分析通过scatter()绘制图形，通常包括：x坐标、y坐标、点名称、聚类类标。

```
#-*- coding:utf-8 -*-
import os
import codecs
import numpy as np
import matplotlib
import matplotlib.pyplot as plt

x = [2.3, 4.5, 3, 7, 6.5, 4, 5.3]
y = [5, 4, 7, 5, 5.3, 5.5, 6.2]

num = np.arange(7)
name = ["a", "b", "c", "d", "e", "f", "g"]

fig, ax = plt.subplots()
ax.scatter(x,y,c='r',s=100)

for i,txt in enumerate(name):  #n
    ax.annotate(txt,(x[i],y[i]))

plt.show()
```

输出结果如下所示：

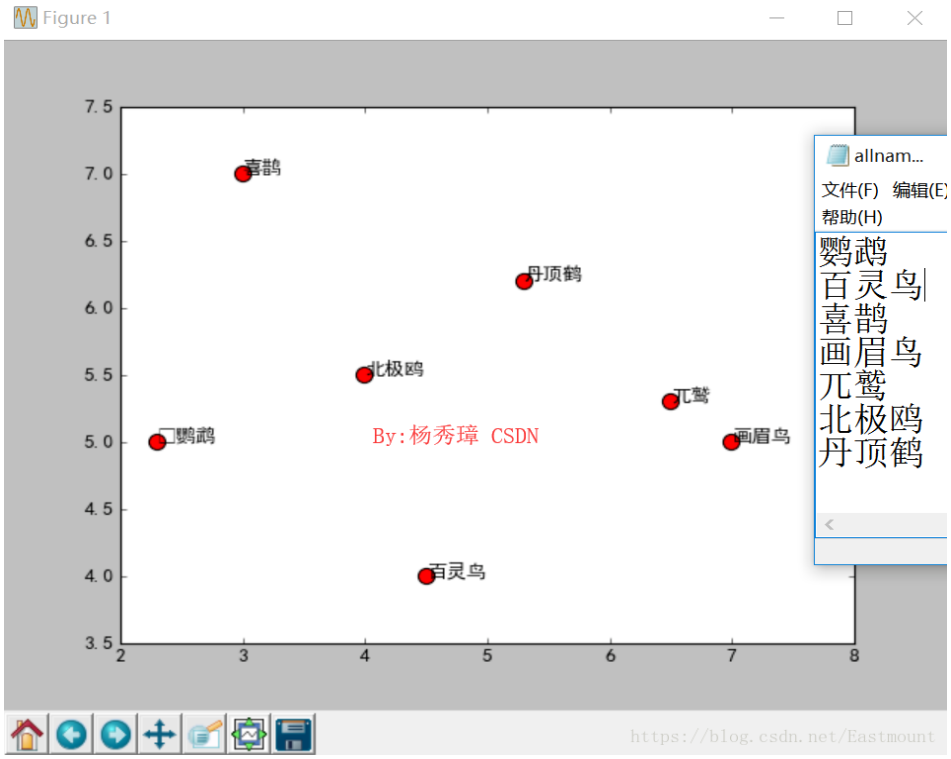


这里是通过 "name = ["a", "b", "c", "d", "e", "f", "g"]" 或 "num = np.arange(7)" 数组设置名称，而实际情况数据很多，比如文本聚类，我们可以通过TXT文本或CSV文件读入数据进行绘制，尤其是中文名称。详细代码如下所示：

```
#-*- coding:utf-8 -*-
import os
import codecs
import numpy as np
import matplotlib
import matplotlib.pyplot as plt

x = [2.3, 4.5, 3, 7, 6.5, 4, 5.3]
y = [5, 4, 7, 5, 5.3, 5.5, 6.2]

n=np.arange(7)
name = ["a", "b", "c", "d", "e", "f", "g"]

fig, ax = plt.subplots()
ax.scatter(x,y,c='r',s=100)

#定义数组读取名称
corpus = []
result = codecs.open('allname.txt', 'r', 'utf-8')
for u in result.readlines():
    print u.strip()
    corpus.append(u.strip())
```

```
    #解决中文和负号'-'显示为方块的问题
matplotlib.rcParams['font.sans-serif'] = ['SimHei']
matplotlib.rcParams['font.family']='sans-serif'
matplotlib.rcParams['axes.unicode_minus'] = False

for i,txt in enumerate(corpus): #n  name
    ax.annotate(txt,(x[i],y[i]))

result.close()
plt.savefig('plot.png', dpi=1200)
plt.show()
```

输出结果如下所示：

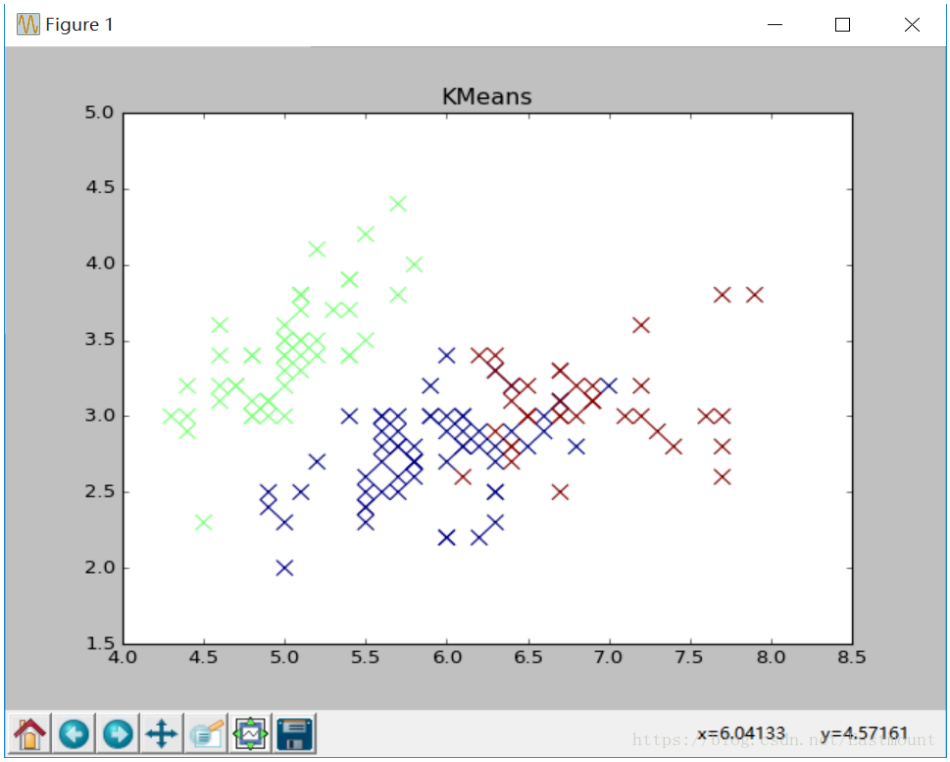


二. Matplotlib聚类类标设置散点图

假设现在对鸢尾花数据集进行KMeans聚类分析，代码如下所示：

```
# -*- coding: utf-8 -*-
#载入数据集
from sklearn.datasets import load_iris
iris = load_iris()
print iris.data            #输出数据集
print iris.target          #输出真实标签
print len(iris.target)
print iris.data.shape    #150个样本 每个样本4个特征


#导入决策树DTC包
from sklearn.cluster import KMeans
clf = KMeans(n_clusters=3)
pre = clf.fit_predict(iris.data)
print pre

#获取花卉两列数据集
X = iris.data
L1 = [x[0] for x in X]
print L1
L2 = [x[1] for x in X]
print L2

#绘图
import numpy as np
import matplotlib.pyplot as plt
plt.scatter(L1, L2, c=pre, marker='x', s=100)
plt.title("KMeans")
plt.show()
```

输出图形如下所示：

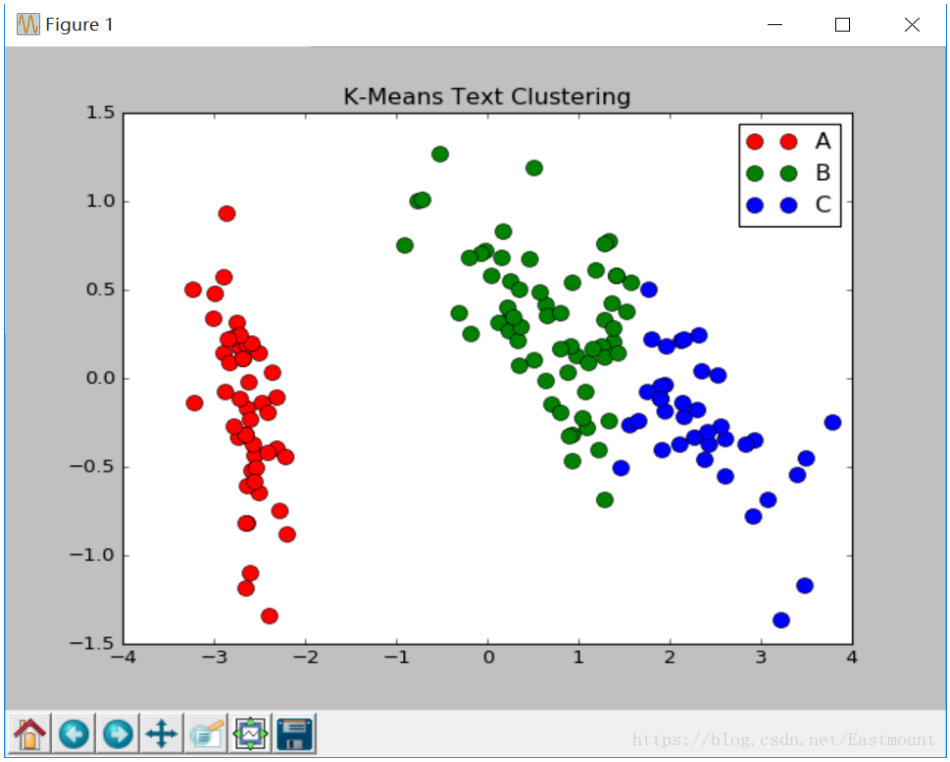


上图却不知道每种颜色的散点对应的类标或名称。这是聚类分析常见的一个问题，如何解决这个问题呢？需要通过循环获取不同类标，再绘制散点图并增加图例。完整代码如下所示：

```
# -*- coding: utf-8 -*-
#载入数据集
from sklearn.datasets import load_iris
iris = load_iris()
print iris.data             #输出数据集
print iris.target           #输出真实标签
print len(iris.target)
print iris.data.shape    #150个样本 每个样本4个特征


#导入决策树DTC包
from sklearn.cluster import KMeans
clf = KMeans(n_clusters=3)
y_pred = clf.fit_predict(iris.data)
print y_pred

#降维绘图
from sklearn.decomposition import PCA
pca = PCA(n_components=2)              #输出两维
newData = pca.fit_transform(iris.data)   #载入N维
print newData
```

```
x = [n[0] for n in newData]
                            y = [n[1] for n in newData]

x1, y1 = [], []
x2, y2 = [], []
x3, y3 = [], []

#分别获取类标为0、1、2的数据 赋值给(x1,y1) (x2,y2) (x3,y3)
i = 0
while i < len(newData):
    if y_pred[i]==0:
        x1.append(newData[i][0])
        y1.append(newData[i][1])
    elif y_pred[i]==1:
        x2.append(newData[i][0])
        y2.append(newData[i][1])
    elif y_pred[i]==2:
        x3.append(newData[i][0])
        y3.append(newData[i][1])
    i = i + 1


import matplotlib.pyplot as plt

#三种颜色
plot1, = plt.plot(x1, y1, 'or', marker="o", markersize=10)
plot2, = plt.plot(x2, y2, 'og', marker="o", markersize=10)
plot3, = plt.plot(x3, y3, 'ob', marker="o", markersize=10)
plt.title("K-Means Text Clustering")  #绘制标题
plt.legend((plot1, plot2, plot3), ('A', 'B', 'C'))

#plt.scatter(x1, x2, c=clf.labels_,  s=100)
plt.show()
```

输出结果如下所示，可以对每类散点样式进行设置，同时绘制标注图形。



希望基础性文章对您有所帮助，如果文章中有错误或不足之处还请海涵。
最后推荐作者的最新出版书籍：

本书主要包括上下两册：

《Python网络数据爬取及分析从入门到精通（爬取篇）》
《Python网络数据爬取及分析从入门到精通（分析篇）》

(By:Eastmount 2018-07-18 深夜12点 http://blog.csdn.net/eastmount/)