机器学习实验 – 2023春

实验六 聚类和提示学习

20201060287 李昂

环境要求：

Python，numpy支持多维度的数组和矩阵运算，pandas数据处理和分析工具，Matplotlib图形化工具，openai 接口。

```
import openai # OpenAI提供的Python包
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

任务一：聚类

> 本任务中你将自己实现kmeans聚类算法，将数据完成聚类，针对数据选择合适的簇数，并且自己实现聚类算法过程，尝试不同的初始中心点，观察聚类结果，将结果可视化并分析。

文件ex6data.csv包含我们的线性可分类问题的数据集。x，y代表纵横坐标。

K-means算法的伪代码如下：

输入： 样本集$D=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m\}$;
聚类簇数k.

过程：

1: 从D中随机选择k个样本作为初始均值向量$\{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\ldots,\boldsymbol{\mu}_k\}$
2: **repeat**
3: 令$C_i=\emptyset\ (1\leq i\leq k)$
4: **for** $j=1,\ldots,m$ **do**
5: 计算样本$\boldsymbol{x}_j$与各均值向量$\boldsymbol{\mu}_i\ (1\leq i\leq k)$的距离：$d_{ji}=||\boldsymbol{x}_j-\boldsymbol{\mu}_i||_2$;
6: 根据距离最近的均值向量确定$\boldsymbol{x}_j$的簇标记：$\lambda_j=\arg\min_{i\in\{1,2,\ldots,k\}}d_{ji}$;
7: 将样本$\boldsymbol{x}_j$划入相应的簇：$C_{\lambda_j}=C_{\lambda_j}\bigcup\{\boldsymbol{x}_j\}$;
8: **end for**
9: **for** $i=1,\ldots,k$ **do**
10: 计算新均值向量：$\boldsymbol{\mu}_i'=\frac{1}{|C_i|}\sum_{\boldsymbol{x}\in C_i}\boldsymbol{x}$;
11: **if** $\boldsymbol{\mu}_i'\neq\boldsymbol{\mu}_i$ **then**
12: 将当前均值向量$\boldsymbol{\mu}_i$更新为$\boldsymbol{\mu}_i'$
13: **else**
14: 保持当前均值向量不变
15: **end if**
16: **end for**
17: **until** 当前均值向量均未更新
18: **return** 簇划分结果

输出： 簇划分$\mathcal{C}=\{C_1,C_2,\ldots,C_k\}$

分析

本任务难度不大，使用numpy实现如上伪代码，尝试多组起始点和聚类中心即可

代码实现

```
# -*- coding: utf-8 -*-
# @Time : 5/29/23 16:16
# @Author : ANG

import numpy as np
```

```
import matplotlib.pyplot as plt


def k_means(data, k, num_iterations):
    # 随机初始化聚类中心
    centroids = data[np.random.choice(range(len(data)), k,
replace=False)]

    for _ in range(num_iterations):
        # 计算每个样本点到聚类中心的距离
        distances = np.linalg.norm(data[:, np.newaxis] - centroids,
axis=2)

        # 分配每个样本点到最近的聚类中心
        labels = np.argmin(distances, axis=1)

        # 更新聚类中心为每个簇的均值
        for i in range(k):
            centroids[i] = np.mean(data[labels == i], axis=0)

    return centroids, labels


# 读取CSV数据
data = np.genfromtxt('../数据集/ex6data.csv', delimiter=',',
skip_header=1)

# 提取x和y数据
x = data[:, 0]
y = data[:, 1]

# 将x和y合并成一个二维数组
data = np.column_stack((x, y))

# 聚类的簇数和迭代次数
k = [2, 3, 4]
```

```
num_iterations = 10

# 调用 K-means 算法
for i in k:
    centroids, labels = k_means(data, i, num_iterations)
    print('k =', i, '时，聚类中心为：', centroids)
    # 绘制数据点和聚类中心
    plt.scatter(x, y, c=labels)
    plt.scatter(centroids[:, 0], centroids[:, 1], marker='X',
color='red', s=100)
    plt.xlabel('X')
    plt.ylabel('Y')
    plt.title('K-means Clustering')
    plt.show()
```

实验结果

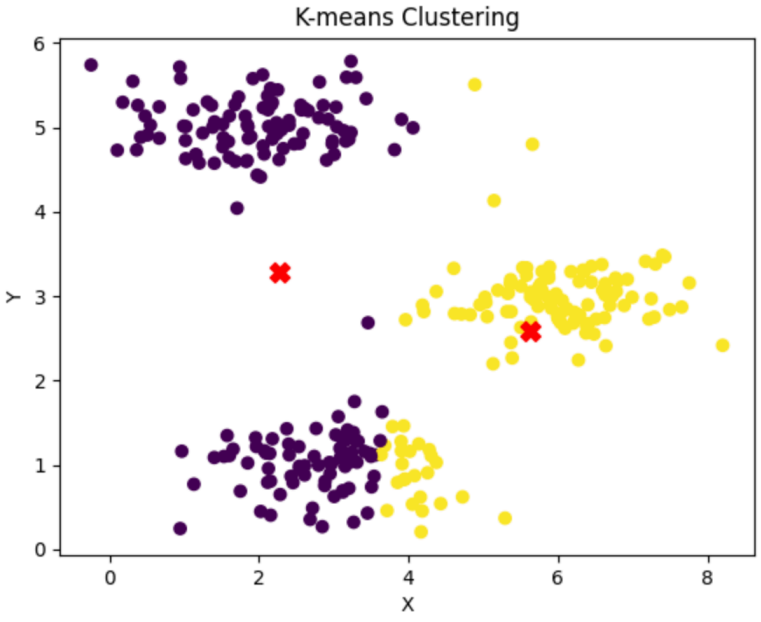


图 1 两个聚类簇

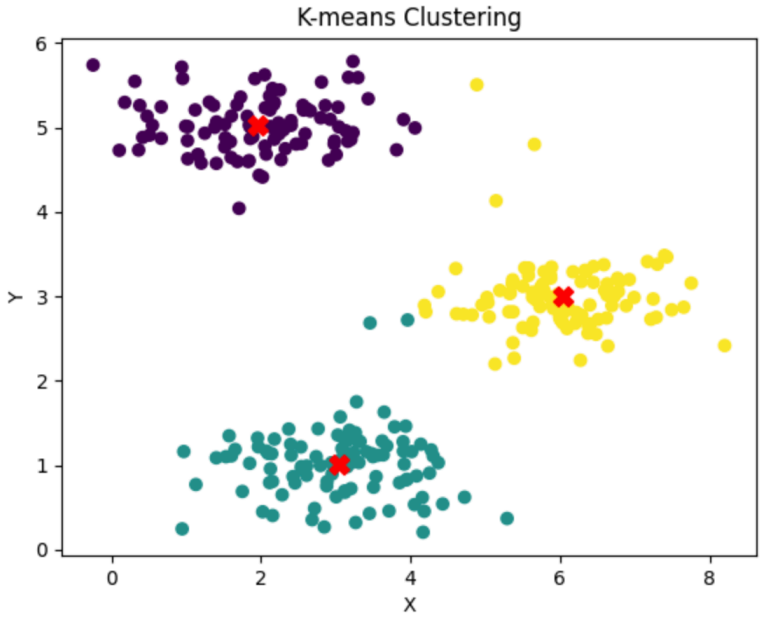


图 2 三个聚类簇

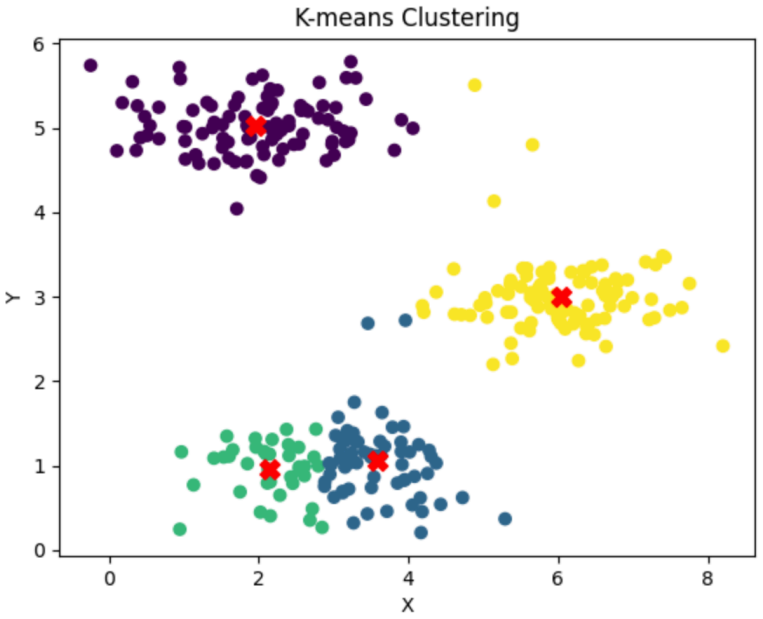


图 3 四个聚类簇

可见，初始点和聚类簇数不同，聚类结果略有不同，本数据集中选取聚类簇数为聚类效果最好。

任务二：提示学习（Prompt Learning）

本任务中你讲尝试使用提示学习完成文本情感分析。提示学习通过构造提示，引导预训练语言模型完成文本分析、生成等相关任务。本任务中，你将以预训练语言模型ChatGLM为基础，对实验五中的评论数据集进行情感分类，考虑到模型推理时间，本次实验只使用20%的数据作为测试，并与实验五中LSTM的结果进行对比。

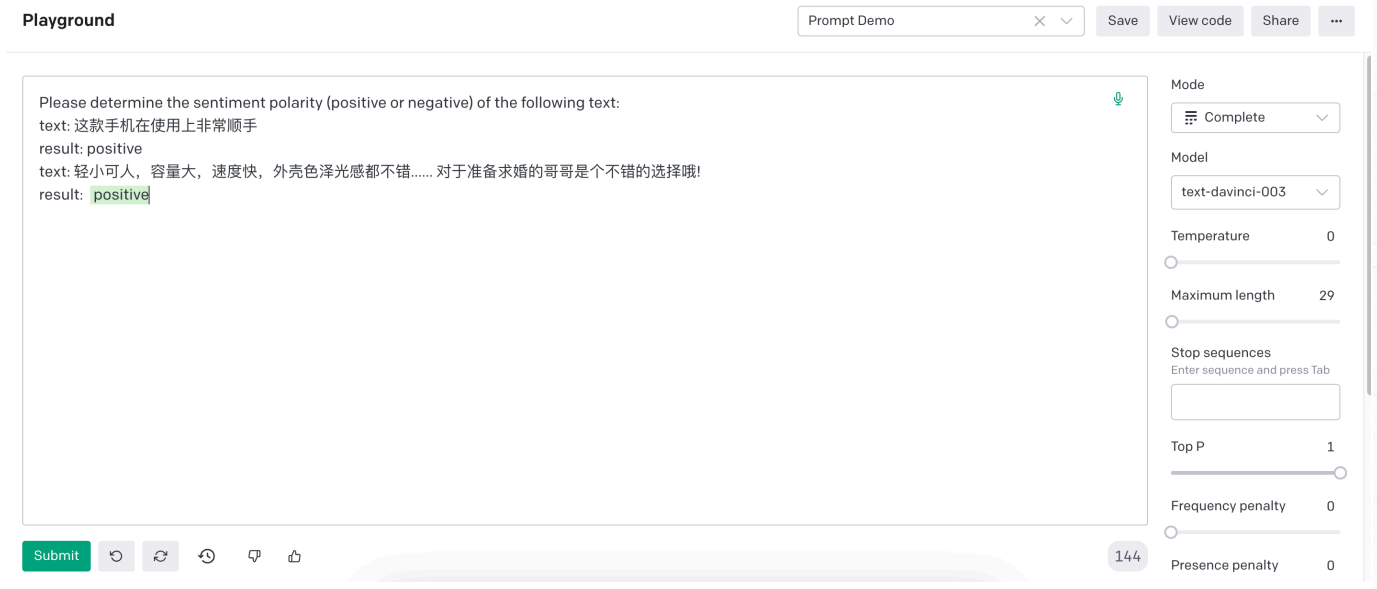
文件comments.csv包含我们的情感分析的数据集。

分析

本任务使用提示学习的方式，通过构建Prompt，完成相应任务。由于实验设备限制，本次实验不采用CahtGLM作为预训练模型，直接调用Open AI API的，通过http请求进行数据交互，完成相关工作。

下面是本次实验的说明：

1. 本次实验使用gpt-3.5-turbo模型完成验证
2. 本次实验构建了能够结构华输出结果的Prompt，并在OpenAI Playground进行验证，能够得到所需结构华输出（Playground仅支持davinci接口，再此只做展示）



OpenAI Playground存档：

https://platform.openai.com/playground/p/Cuvl4qxtFyPtCremze0wiAJX?model=text-davinci-003

3. 若要运行程序，请替换`openai.api_key`为您的API KEY，并保证文件所在位置和文件路径相对应，若遇到`[RateLimitError: That model is currently overloaded with other requests]`说明此时服务器繁忙，请在夜晚/其他服务器空闲时间进行验证。

代码实现

```
# -*- coding: utf-8 -*-
# @Time : 5/29/23 17:10
# @Author : ANG

import openai
import pandas as pd

# 设置你的 OpenAI API 密钥
openai.api_key = 'sk-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'


# 读取 CSV 文件
def read_csv(dataset):
    data = pd.read_csv(dataset)
    # 截取 20% 的数据进行测试
    size = int(len(data) * 0.2)
    data = data.sample(n=size)
    return data


# 生成提示文本
def generate_prompt(text):
    """
    为了提高正确率，使用英文提示文本，并增加了一些提示文本（均来自comments.csv文件）
    该prompt已经在OpenAI Playground进行验证，能够输出格式话结果，便于后续的结果比对。
```

```
    """
    return """Please determine the sentiment polarity (positive or
negative) of the following text:
text: 这款手机在使用上非常顺手
result: positive
text: {}
result:""".format(
        text.capitalize()
    )


# 情感语义分析
def analyze_sentiment(text):
    prompt = generate_prompt(text)

    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "user", "content": generate_prompt(text)},
        ],
        max_tokens=20,
        temperature=0,
        frequency_penalty=0,
        presence_penalty=0,
        stop=None,
        n=1,
    )

    sentiment = response.choices[0].message.content.strip()

    return sentiment


# 分析情感并与正确结果比对正确率
def analyze_and_evaluate(dataset):
    data = read_csv(dataset)
```

```
    # 总行数
    total_count = len(data)
    # 正确行数
    correct_count = 0

    for index, row in data.iterrows():
        text = row['comments']
        expected_sentiment = int(row['label'])

        sentiment = analyze_sentiment(text)
        predicted_sentiment = 1 if sentiment == "positive" else 0

        if predicted_sentiment == expected_sentiment:
            correct_count += 1

    accuracy = correct_count / total_count * 100
    print("准确率：{:.2f}%".format(accuracy))


# 测试情感分析并评估准确率
dataset = '../数据集/comments.csv'
analyze_and_evaluate(dataset)
```

代码结构：

`read_csv`函数用于读取csv文件，并截取20%进行实验；`generate_prompt`生成在OpenAI Playground验证过的Prompt；`analyze_sentiment`使用`openai`提供的`ChatCompletion`类完成验证；`analyze_and_evaluate`比对正确率。

实验结果

```
Python 控制台>>> runfile('/Users/wallanceleon/Desktop/机器学习/
准确率：98.42%

>>>
```

图 4 提示学习正确率

后记

本次任务一较为简单，再此不在赘述。

任务二提示学习中遇到并解决了以下问题：

1. 首次使用OpenAI Python API 参考了官网Demo ， github地址如下：
   https://github.com/openai/openai-quickstart-python
2. 网络环境问题，由于访问量巨大和OpenAI服务器问题，经常出现不同的http报错，这里参考了：
   https://help.openai.com/en/?q=Error+Code
3. 由于gpt-3.5-turbo并不支持`openai.Completion`类，只能使用`openai.ChatCompletion`，此处参考了：https://platform.openai.com/docs/api-reference/chat/create
4. Prompt设计部分思路来源：

   https://github.com/taishan1994/ChatABSA