E11 Expected Maximize Algorithm

Suixin Ou

School of Computer Science
Sun Yat-sen University

December 21, 2021

Background

The Iris Data Set

- The UCI dataset (http://archive.ics.uci.edu/ml/index.php) is the most widely used dataset for machine learning. If you are interested in other datasets in other areas, you can refer to https://www.zhihu.com/question/63383992/answer/222718972.
- It is perhaps the best known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are NOT linearly separable from each other.

Background

The Iris Data Set

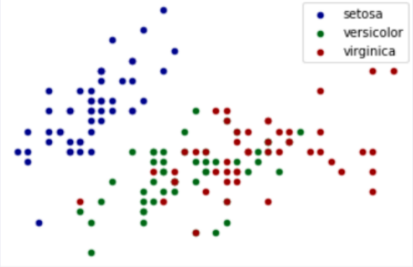
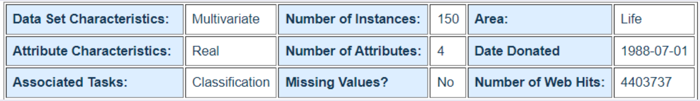


Figure 1: Visualization of Iris Dataset

Task

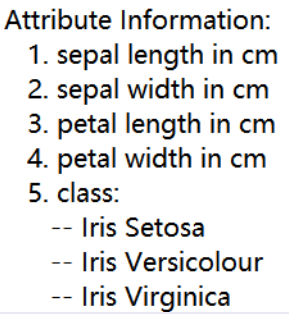
Description

- Dataset statistics



Data Set Characteristics:	Multivariate	**Number of Instances:**	150	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	4	**Date Donated**	1988-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	4403737

- Domain information

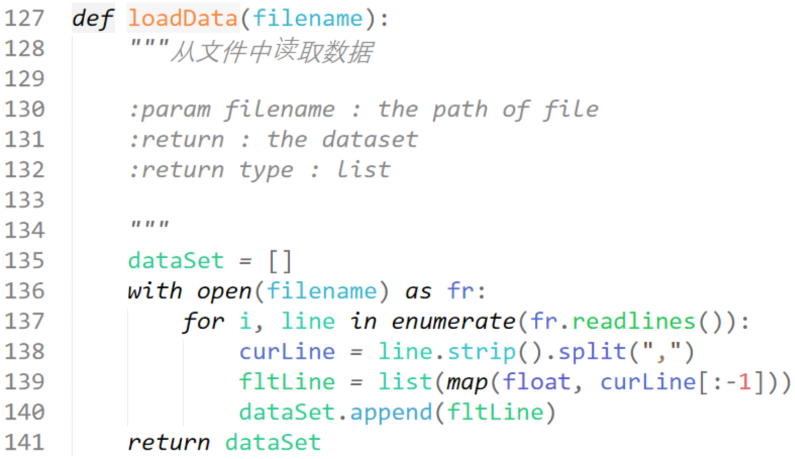


Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
   -- Iris Setosa
   -- Iris Versicolour
   -- Iris Virginica

Solution

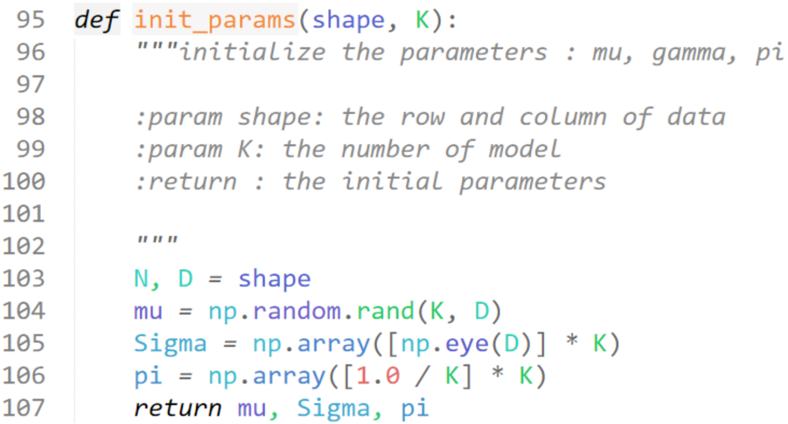
Read the file ”iris.data”



```python
def loadData(filename):
    """从文件中读取数据

    :param filename : the path of file
    :return : the dataset
    :return type : list

    """
    dataSet = []
    with open(filename) as fr:
        for i, line in enumerate(fr.readlines()):
            curLine = line.strip().split(",")
            fltLine = list(map(float, curLine[:-1]))
            dataSet.append(fltLine)
    return dataSet
```

Solution

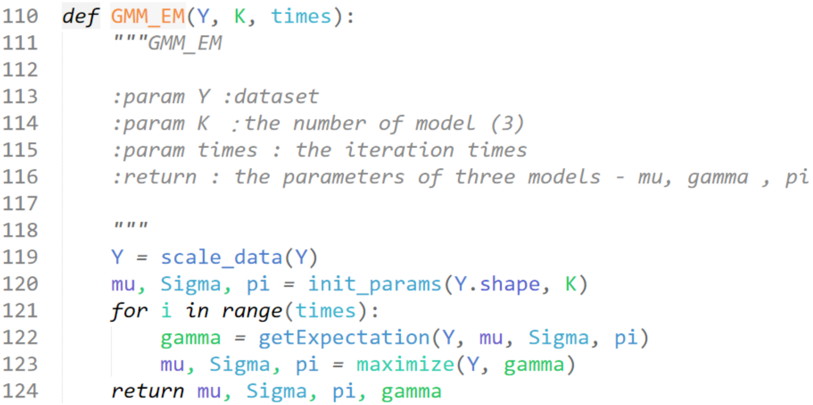
Initialize parameters



```python
def init_params(shape, K):
    """initialize the parameters : mu, gamma, pi

    :param shape: the row and column of data
    :param K: the number of model
    :return : the initial parameters

    """
    N, D = shape
    mu = np.random.rand(K, D)
    Sigma = np.array([np.eye(D)] * K)
    pi = np.array([1.0 / K] * K)
    return mu, Sigma, pi
```

Solution

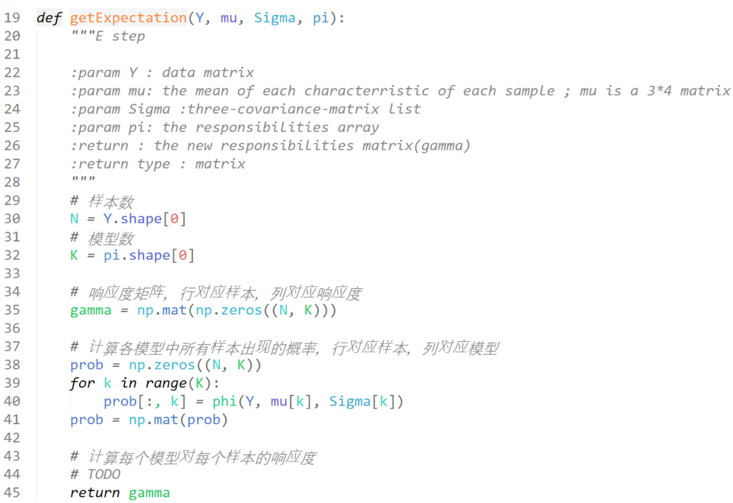
Expected Maximize algorithm framework



```python
def GMM_EM(Y, K, times):
    """GMM_EM

    :param Y :dataset
    :param K  :the number of model (3)
    :param times : the iteration times
    :return : the parameters of three models - mu, gamma , pi

    """
    Y = scale_data(Y)
    mu, Sigma, pi = init_params(Y.shape, K)
    for i in range(times):
        gamma = getExpectation(Y, mu, Sigma, pi)
        mu, Sigma, pi = maximize(Y, gamma)
    return mu, Sigma, pi, gamma
```

Solution

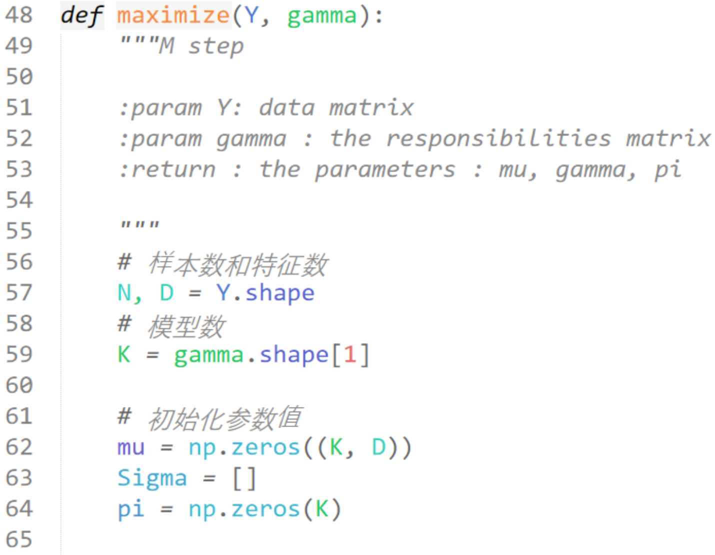
Please Finish the getExpectation function.



```python
def getExpectation(Y, mu, Sigma, pi):
    """E step

    :param Y : data matrix
    :param mu: the mean of each characterristic of each sample ; mu is a 3*4 matrix
    :param Sigma :three-covariance-matrix list
    :param pi: the responsibilities array
    :return : the new responsibilities matrix(gamma)
    :return type : matrix
    """
    # 样本数
    N = Y.shape[0]
    # 模型数
    K = pi.shape[0]

    # 响应度矩阵，行对应样本，列对应响应度
    gamma = np.mat(np.zeros((N, K)))

    # 计算各模型中所有样本出现的概率，行对应样本，列对应模型
    prob = np.zeros((N, K))
    for k in range(K):
        prob[:, k] = phi(Y, mu[k], Sigma[k])
    prob = np.mat(prob)

    # 计算每个模型对每个样本的响应度
    # TODO
    return gamma
```

Solution

Please Finish the maximize function.



```python
def maximize(Y, gamma):
    """M step

    :param Y: data matrix
    :param gamma : the responsibilities matrix
    :return : the parameters : mu, gamma, pi

    """
    # 样本数和特征数
    N, D = Y.shape
    # 模型数
    K = gamma.shape[1]

    # 初始化参数值
    mu = np.zeros((K, D))
    Sigma = []
    pi = np.zeros(K)

```

Submission

Submission

pack your report E11_YourNumber.pdf and source code into zip file E11_YourNumber.zip, then send it to ai_course2021@163.com.

The End