《机器学习》读书笔记

一、算法概述（算法名称及原理）

最大期望算法在统计中被用于寻找参数的最大似然估计。在统计计算中，最大期望（EM）算法是在概率模型中寻找参数最大似然估计或者最大后验估计的算法，其中概率模型依赖于无法观测的隐性变量。最大期望算法经常用在机器学习和计算机视觉的数据聚类（Data Clustering）领域。

最大期望算法经过两个步骤交替进行计算，第一步是计算期望（E），利用对隐藏变量的现有估计值，计算其最大似然估计值；第二步是最大化（M），最大化在 E 步上求得的最大似然值来计算参数的值。M 步上找到的参数估计值被用于下一个 E 步计算中，这个过程不断交替进行。

二、算法设计（流程图及主要分段代码，附详细代码注释）

输入：观察数据$x = (x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(m)})$，联合分布$P(x^{(i)}, z^{(i)}; \theta)$，条件分布$P(z^{(i)}|x^{(i)}; \theta)$，最大迭代次数$J$。

- 随机初始化模型参数θ的初始值θ^0。
- $for\ j\ from\ 1\ to\ J$开始EM算法迭代。
  - E步:计算联合分布的条件概率期望

$$Q_i(z^{(i)}) = P(z^{(i)}|x^{(i)}; \theta)$$

$$L(\theta, \theta^j) = \sum_{i=1}^{m} \sum_{z^{(i)}} Q_i(z^{(i)}) logP(x^{(i)}, z^{(i)}; \theta)$$

  - M步:极大化$L(\theta, \theta^j)$，得到θ^{j+1}

$$\theta^{j+1} = \arg\max_x L(\theta, \theta^j)$$

  - 如果θ^{j+1}收敛，则算法结束，否则继续迭代。
- 输出模型参数θ。

```python

if __name__ == '__main__':

    wine = datasets.load_wine()

    feature_name = ['酒精','苹果酸','灰','灰的碱性','镁','总酚','类黄酮','非黄烷类酚类','花青素','颜色强度','色调'

    # and testing (25%) sets.
    skf = StratifiedKFold(n_splits=4)#分层采样，确保训练集，测试集中各类别样本的比例与原始数据集中相同。
    # Only take the first fold.
    train_index, test_index = next(iter(skf.split(wine.data, wine.target)))
    data = wine.data

    def maxminnorm(array):
        maxcols=array.max(axis=0)
        mincols=array.min(axis=0)
        data_shape = array.shape
        data_rows = data_shape[0]
        data_cols = data_shape[1]
        t=np.empty((data_rows,data_cols))
        for i in range(data_cols):
            t[:,i]=(array[:,i]-mincols[i])/(maxcols[i]-mincols[i])
        return t

    data=maxminnorm(data)

    y = wine.target
```

wine先进行归一化处理

```python
#    re = []
    feature_pairs=[[3,11]]
    for k, pair in enumerate(feature_pairs):
        x = data[:, pair]
        #print(x)   # y是目标值的列向量 它等于分类0，1,2时的值对应的X的位置 就可以算出每一类的实际均值
        m = np.array([np.mean(x[y == i], axis=0) for i in range(3)])  # 均值的实际值
#        print ('实际均值 = \n', m)
        num_iter = 100
        n, d = x.shape
        mu1 = x.min(axis=0)
        mu2 = x.max(axis=0)
        mu3 = np.median(x,axis=0)
        print( mu1, mu2,mu3)
        sigma1 = np.identity(d)
        sigma2 = np.identity(d)
        sigma3 = np.identity(d)
        pi = 1.0/3
        # EM
        for i in range(num_iter):
            # E Step
            norm1 = multivariate_normal(mu1, sigma1)
            norm2 = multivariate_normal(mu2, sigma2)
            norm3 = multivariate_normal(mu3, sigma3)
            tau1 = pi * norm1.pdf(x)
            tau2 = pi * norm2.pdf(x)
            tau3 = pi * norm3.pdf(x)
            gamma1 = tau1 / (tau1 + tau2 + tau3)
            gamma2 = tau2 / (tau1 + tau2 + tau3)
            gamma3 = tau3 / (tau1 + tau2 + tau3)
            # M Step
            mu1 = np.dot(gamma1, x) / np.sum(gamma1)
            mu2 = np.dot(gamma2, x) / np.sum(gamma2)
            mu3 = np.dot(gamma3, x) / np.sum(gamma3)
            sigma1 = np.dot(gamma1 * (x - mu1).T, x - mu1) / np.sum(gamma1)
            sigma2 = np.dot(gamma2 * (x - mu2).T, x - mu2) / np.sum(gamma2)
            sigma3 = np.dot(gamma3 * (x - mu3).T, x - mu3) / np.sum(gamma3)
            pi = (np.sum(gamma1)+np.sum(gamma2)+np.sum(gamma3)) / n
#            print (i, ":\t", mu1, mu2)
#        print(u'类别概率:\t', pi)
```

主要的Estep和Mstep

三、选用数据（数据集描述，包括来源，行数，列数，格式等）

iris 行数：150 列数：5

列属性及取值:

1) 萼片长度 cm，数值型
2) 萼片宽度 cm，数值型
3) 花瓣长度 cm，数值型
4) 花瓣宽度 cm 数值型

类别：
- Iris Setosa
- Iris Versicolour
- Iris Virginica

car，行数：1728，列数：6
列属性及取值：
1) buying: vhigh, high, med, low.
2) maint: vhigh, high, med, low.
3) doors: 2, 3, 4, 5more.
4) persons: 2, 4, more.
5) lug_boot: small, med, big.
6) safety: low, med, high.
类别：
unacc, acc, good, vgood

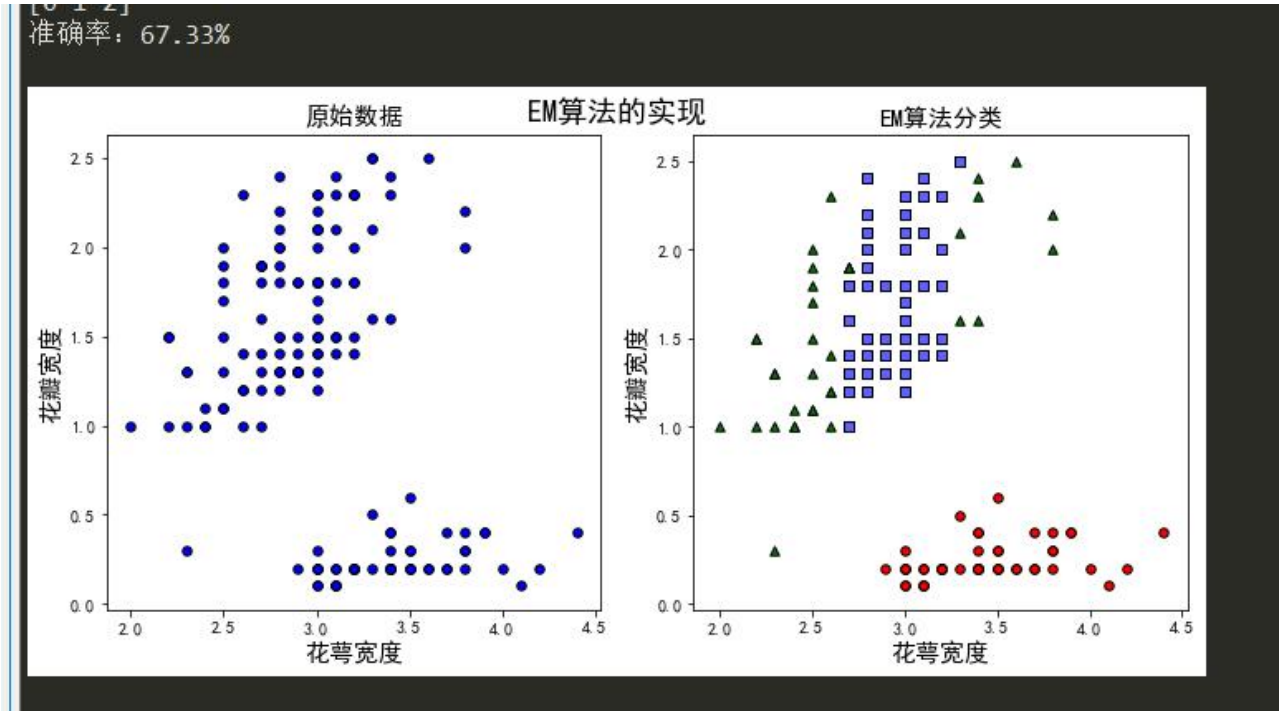
wine，行数：178，列数：13
属性：
1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10) Color intensity
11) Hue
12) OD280/OD315 of diluted wines
13) Proline
类别：
Alcohol 1，2，3

四、评价方法（说明训练集和测试集分配方法及评价指标）

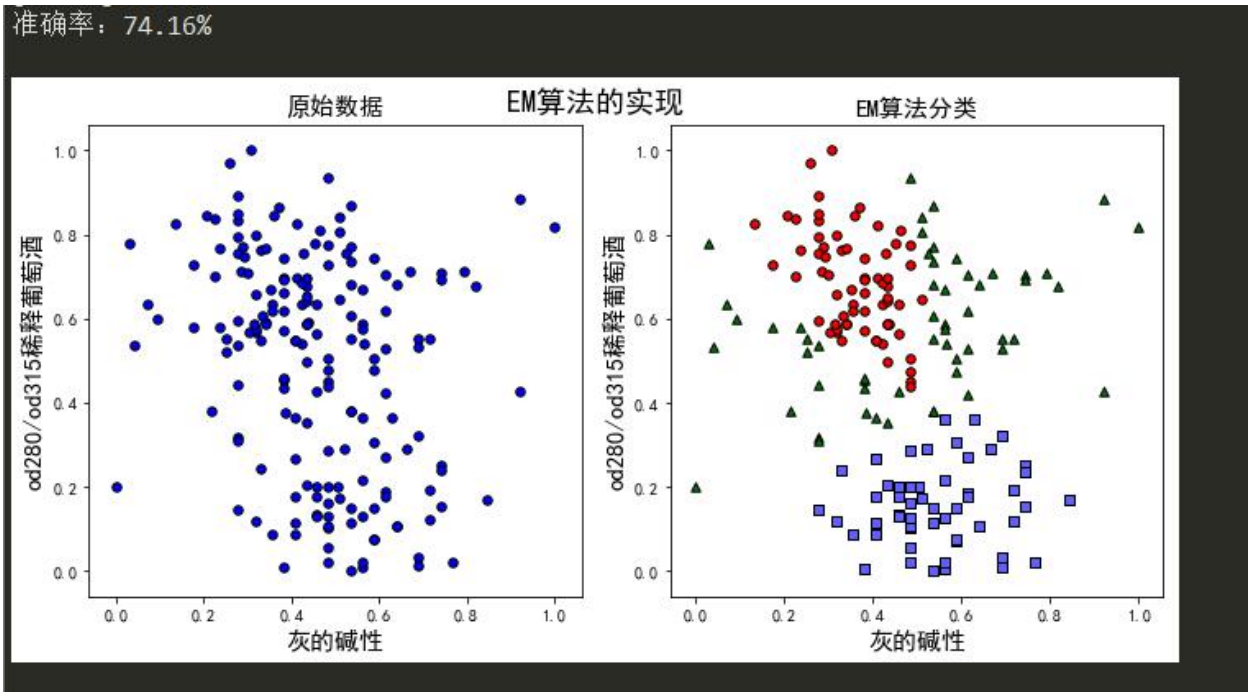
使用准确率进行评价

五、实验结果截图

Iris：



Wine：



六、实验结果分析及比较

本实验使用两两组合的方式找出最佳的组合，即准确率最高的组合，对于同样的算法，wine 的聚类效果可能会更好些。

七、遇到的问题及解决方法，实践心得

EM 算法的优点：

1) 聚类。

2) 算法计算结果稳定、准确。
3) EM 算法自收敛，既不需要事先设定类别，也不需要数据间的两两比较合并等操作。

EM 算法的缺点：

1）对初始化数据敏感。

2）EM 算法计算复杂，收敛较慢，不适于大规模数据集和高维数据。

3）当所要优化的函数不是凸函数时，EM 算法容易给出局部最优解，而不是全局最优解。

关于 CAR 离散属性，使用 EM 聚类，存在问题