人工智能理论与实践

第四章 机器学习 实验

信息学部 信息与通信工程学院

主要内容

01 实验一 主成分分析

02 实验二 聚类

03 实验三 决策树

04 实验四 支持向量机

主要内容

01 实验一 主成分分析
02 实验二 聚类
03 实验三 决策树
04 实验四 支持向量机

1 主成分分析

1.1 实验目的

- 掌握主成分分析（PCA）原理
- 编程实现PCA对数据集进行降维

1.2 开发环境与编程语言

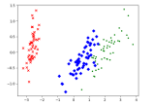
- Windows或Ubuntu操作系统
- Python编程语言

```
def self_pca(X, k):
    n_samples, n_features = X.shape
    mean = np.array([np.mean(X[:, i])
    norm_X = X - mean
    scatter_matrix = np.dot(np.transp
    eig_val, eig_vec = np.linalg.eig(
    eig_pairs = [(np.abs(eig_val[i]),
    eig_pairs.sort(reverse=True)
    feature = np.array([ele[1] for el
    return np.dot(norm_X, np.transpos


if __name__ == '__main__':
    # 以字典的形式加载鸢尾花数据集，使用y表示
    data = load_iris()
    y = data.target
    x = data.data

    # 调用PCA算法进行降维主成分分析
    # 指定主成分个数，即降维后数据维度，降维
    pca = PCA(n_components=2)
    reduced_x = pca.fit_transform(x)
```

1.3 实验原理

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

原数据集



降维并可视化

通过编程实现PCA，将数据集的多维数据降为二维数据并可视化

1.4 实验步骤

1）读取iris数据集

2）定义实现PCA的函数

3）使用PCA对数据进行降维

4）获得结果并将结果可视化

5）测试算法的正确性

1.5 实验报告

1）题目：利用主成分分析实现数据降维。

2）原代码。

3）总结程序运行中的报错原因并记录。

4）运行结果截图，要求输出降维可视化结果（3维/2维）。

5）尝试改变指定主成分个数，分析不同主成分数量的降维结果。

6）总结由本实验所获得心得体会。

主要内容

01 实验一 主成分分析
02 实验二 聚类
03 实验三 决策树
04 实验四 支持向量机

2.1 实验目的

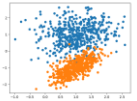
- 掌握聚类算法原理
- 编程实现聚类算法解决数据聚类问题

2.2 开发环境与编程语言

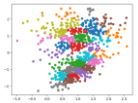
- Windows或Ubuntu操作系统
- Python编程语言

```python
def kmeans():
    model = KMeans(n_clusters=2)
    model.fit(X)
    return model.predict(X)


def minibatchKmeans():
    model = MiniBatchKMeans(n_clusters=2)
    model.fit(X)
    return model.predict(X)


def meanshift():
    model = MeanShift()
    return model.fit_predict(X)


def optics():
    model = OPTICS(eps=0.8, min_samples=10)
    return model.fit_predict(X)
```

2.3 实验原理



两类数据原始图



聚类后可视化

通过编程实现聚类算法，解决数据的聚类问题

2.4 实验步骤

1）随机初始化数据

2）定义实现聚类算法的函数

3）获得结果并将结果可视化

4）比较各聚类算法

2.5 实验报告

1）题目：利用多种聚类算法实现数据聚类

2）原代码。

3）产生至少1000个数据点，可视化原始分布，并对其进行聚类。

4）运行结果截图，要求输出不少于三种方法的聚类结果可视化结果图。

5）对比原始数据分布，分析不同聚类方法的聚类结果。

6）总结由本实验所获得心得体会。

主要内容

01 实验一 主成分分析
02 实验二 聚类
03 实验三 决策树
04 实验四 支持向量机

3.1 实验目的

- 掌握决策树算法原理
- 编程实现决策树算法解决二分类决策问题

3.2 开发环境与编程语言

- Windows或Ubuntu操作系统
- Python编程语言

```
def classify(inputTree, featLabels, testVec):
    firstStr = next(iter(inputTree))  # 获取决策树
    secondDict = inputTree[firstStr]  # 下一个字典
    featIndex = featLabels.index(firstStr)
    for key in secondDict.keys():
        if testVec[featIndex] == key:
            if type(secondDict[key]).__name__ ==
                classLabel = classify(secondDict
            else:
                classLabel = secondDict[key]
    return classLabel


if __name__ == '__main__':
    dataSet, labels = createDataSet()
    featLabels = []
    myTree = createTree(dataSet, labels, featLa

    print(myTree)
    visualization.createPlot(myTree)  # 决策树可视
```

3.3 实验原理

```python
def createDataSet():
    dataSet = [[0, 0, 0, 0, 'no'],  # 数据集
               [0, 0, 0, 1, 'no'],
               [0, 1, 0, 1, 'yes'],
               [0, 1, 1, 0, 'yes'],
               [0, 0, 0, 0, 'no'],
               [1, 0, 0, 0, 'no'],
               [1, 0, 0, 1, 'no'],
               [1, 1, 1, 1, 'yes'],
               [1, 0, 1, 2, 'no'],
               [1, 0, 1, 2, 'yes'],
               [2, 0, 1, 2, 'yes'],
               [2, 0, 1, 1, 'yes'],
               [2, 1, 0, 1, 'yes'],
               [2, 1, 0, 2, 'yes'],
               [2, 0, 0, 0, 'no']]
    labels = ['年龄', '有工作', '有自己的房子', '信贷情况']  # 分类属性
    return dataSet, labels  # 返回数据集和分类属性
```


原始数据集

决策树可视化

通过编程实现决策树算法，构造决策树并可视化

3.4 实验步骤

1）读取数据

2）定义实现决策树算法的函数

3）获得结果并将决策树可视化

4）验证决策树算法的正确性

3.5 实验报告

1）题目：利用决策树算法实现二分类决策。

2）原代码。

3）总结程序运行中的报错原因并记录。

4）运行结果截图，要求输出生成的决策树可视化图。

5）尝试自己构建一类决策数据，生成决策树。

6）总结由本实验所获得心得体会。

主要内容

01 实验一 主成分分析

02 实验二 聚类

03 实验三 决策树

04 **实验四 支持向量机**

4.1 实验目的

- 掌握支持向量机算法原理
- 编程实现支持向量机算法解决

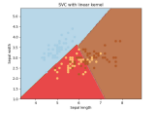
4.2 开发环境与编程语言

- Windows或Ubuntu操作系统
- Python编程语言

```
# 将所得参数进行模型训练
svc = svm.SVC(kernel='linear', C=1, gamma='auto').
# 建图
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() +
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() +
h = (x_max / x_min) / 100
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
                     np.arange(y_min, y_max, h))
plt.subplot(1, 1, 1)  # 将显示界面分割成1*1 图形标号为1
Z = svc.predict(np.c_[xx.ravel(), yy.ravel()])  #
Z = Z.reshape(xx.shape)  # 重新构造行列
plt.contourf(xx, yy, Z, cmap=plt.cm.Paired, alpha=
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Pai
plt.xlabel('Sepal length')  # x轴标签
plt.ylabel('Sepal width')  # y轴标签
plt.xlim(xx.min(), xx.max())  # 设置x轴的数值显示范围
plt.title('SVC with linear kernel')  # 设置显示图像的
plt.savefig('./test1.png')  # 存储图像
plt.show()  # 显示
```

4.3 实验原理

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa



原始数据集

向量机可视化

通过编程实现支持向量机算法，构造向量机并可视化

4.4 实验步骤

1. 读取数据
2. 定义实现支持向量机算法的函数
3. 获得结果并将结果可视化
4. 验证支持向量机算法的正确性

4.5 实验报告

1）题目：利用支持向量机实现数据分类。

2）原代码。

3）总结程序运行中的报错原因并记录。

4）运行结果截图，要求输出数据分类可视化图。

5）并对支持向量机的分类结果进行分析。

6）总结由本实验所获得心得体会。

人工智能理论与实践

第四章 机器学习 实验

信息学部 信息与通信工程学院