云南大学数学与统计学院
《数据挖掘与决策支持实验》上机实践报告

课程名称： 运筹学数据挖掘与决策支持实验	**年级：** 2015 级	**上机实践成绩：**
指导教师： 彭程	**姓名：** 刘鹏	**专业：** 信息与计算科学
上机实践名称： 用 Relief 算法对 iris 数据进行特征选择	**学号：** 20151910042	**上机实践日期：** 2018-07-04
上机实践编号： 02	**组号：**	

一. 实验目的

学习使用 R 语言进行变量选择。

二. 实验内容

三. 实验平台

Windows 10 Pro 1803；
Microsoft© Visual Studio 2017 Enterprise。
Version 1.1.442 – © 2009-2018 RStudio, Inc.

四. 算法设计

数据预分析 ：安德森鸢尾花卉数据集（英文：Anderson's Iris data set），也称鸢尾花卉数据集（英文：Iris flower data set）或费雪鸢尾花卉数据集（英文：Fisher's Iris data set），是一类多重变量分析的数据集。它最初是埃德加·安德森从加拿大加斯帕半岛上的鸢尾属花朵中提取的地理变异数据，后由罗纳德·费雪作为判别分析的一个例子，运用到统计学中。

其数据集包含了 150 个样本，都属于鸢尾属下的三个亚属，分别是山鸢尾、杂色鸢尾和维吉尼亚鸢尾，每类 50 个数据，每个数据包含 4 个特征。4 个特征被用作样本的定量分析，它们分别是花萼和花瓣的长度和宽度。基于这四个特征的集合，费雪发展了一个线性判别分析以确定其属种。

Iris 数据集是常用的分类实验数据集。可通过花萼长度，花萼宽度，花瓣长度，花瓣宽度 4 个属性预测鸢尾花卉属于（Setosa，Versicolour，Virginica）三个种类中的哪一类。

该数据集包含了 5 个属性：

Sepal.Length（花萼长度），单位是 cm
Sepal.Width（花萼宽度），单位是 cm

Petal.Length（花瓣长度），单位是 cm
Petal.Width（花瓣宽度），单位是 cm
种类：Iris Setosa（山鸢尾）、Iris Versicolour（杂色鸢尾），以及 Iris Virginica（维吉尼亚鸢尾）

Relief 算法简介：这是一个基于特征加权的特征选择算法。Relief 算法最早由 Kira 提出，最初局限于两类数据的分类问题。Relief 算法是一种特征权重算法（Feature weighting algorithms），根据各个特征和类别的相关性赋予特征不同的权重，权重小于某个阈值的特征将被移除。Relief 算法中特征和类别的相关性是基于特征对近距离样本的区分能力。算法从训练集D中随机选择一个样本R，然后从和R同类的样本中寻找最近邻样本H，称为 Near Hit，从和R不同类的样本中寻找最近邻样本M，称为 Near Miss，然后根据以下规则更新每个特征的权重：如果R和 Near Hit 在某个特征上的距离小于R和 Near Miss 上的距离，则说明该特征对区分同类和不同类的最近邻是有益的，则增加该特征的权重；反之，如果R和 Near Hit 在某个特征的距离大于R和 Near Miss 上的距离，说明该特征对区分同类和不同类的最近邻起负面作用，则降低该特征的权重。以上过程重复m次，最后得到各特征的平均权重。特征的权重越大，表示该特征的分类能力越强，反之，表示该特征分类能力越弱。Relief 算法的运行时间随着样本的抽样次数m和原始特征个数N的增加线性增加，因而运行效率非常高。

Relief（Relevant Features）是著名的过滤式特征选择方法，Relief 为一系列算法，它包括最早提出的 Relief 以及后来拓展的 Relief-F 和 R*Relief*-F ，其中最早提出的 Relief 针对的是二分类问题，RRelief-F 算法可以解决多分类问题，RRelief-F 算法针对的是目标属性为连续值的回归问题。

4.1 原始的 Relief 算法

最早提出的 Relief 算法主要针对二分类问题，该方法设计了一个“相关统计量”来度量特征的重要性，该统计量是一个向量，向量的每个分量是对其中一个初始特征的评价值，特征子集的重要性就是子集中每个特征所对应的相关统计量之和，因此可以看出，这个“相关统计量”也可以视为是每个特征的“权值”。可以指定一个阈值τ，只需选择比τ大的相关统计量对应的特征值，也可以指定想要选择的特征个数k，然后选择相关统计量分量最大的k个特征。

有了 Relief 的基本思想，那么现在的问题就转换成如何得到一种有效的权值或者相关统计量类对特征进行度量，Relief 借用了“假设间隔”（hypothesis margin）的思想，我们知道在分类问题中，常常会采用决策面的思想来进行分类，“假设间隔”就是指在保持样本分类不变的情况下，决策面能够移动的最大距离，可以表示为：

$$\theta = \frac{1}{2}(\|x - M(x)\| - \|x - H(x)\|) \tag{4.1-1}$$

其中，$M(x)$和$H(x)$指的是与x同类的和与x非同类的最近邻点。我们知道，当一个属性对分类有利时，则该同类样本在该属性上的距离较近，而异类样本在该属性上的距离较远，因此，若将假设间隔推广到对属性的评价中，则对应于公式(4.1-1)圆括号中的第一项越小，第二项越大，则该属性对分类越有利。“假设间隔”能对各维度上的特征的分类能力进行评价，从而就可以近似地估计出对分类最有用的特征子集，

Relief 正是利用了这个特性。

假设训练集D为(x_1, y_1)，(x_2, y_2)，…，(x_m, y_m)，对每个样本x_i，计算与x_i同类别的最近邻$x_{i,nh}$，称为是“猜中近邻”（near−heat），然后计算与x_i非同类别的最近邻$x_{i,nm}$，称为是“猜错近邻”（near−miss），则属性j对应的相关统计量为：

$$\delta^j = \sum_i -\mathrm{diff}\left(x_i^j,\ x_{i,\ nh}^j\right)^2 + \mathrm{diff}\left(x_i^j,\ x_{i,\ nm}^j\right)^2 \tag{4.1-2}$$

其中，x_a^j代表样本x_a在属性j上的取值，$\mathrm{diff}\left(x_a^j,\ x_b^j\right)$的计算取决于属性$j$的类型。

对离散型属性：

$$\mathrm{diff}\left(x_a^j,\ x_b^j\right) = \begin{cases} 0, & x_a^j = x_b^j \\ 1, & \text{otherwise} \end{cases} \tag{4.1-3}$$

对连续型属性：

$$\mathrm{diff}\left(x_a^j,\ x_b^j\right) = \left|x_a^j - x_b^j\right| \tag{4.1-4}$$

五. 程序代码

5.1 程序描述

Iris 数据量太少，难以进行特征提取，所以这里用 k-means 方法进行一下聚类。

5.2 程序代码

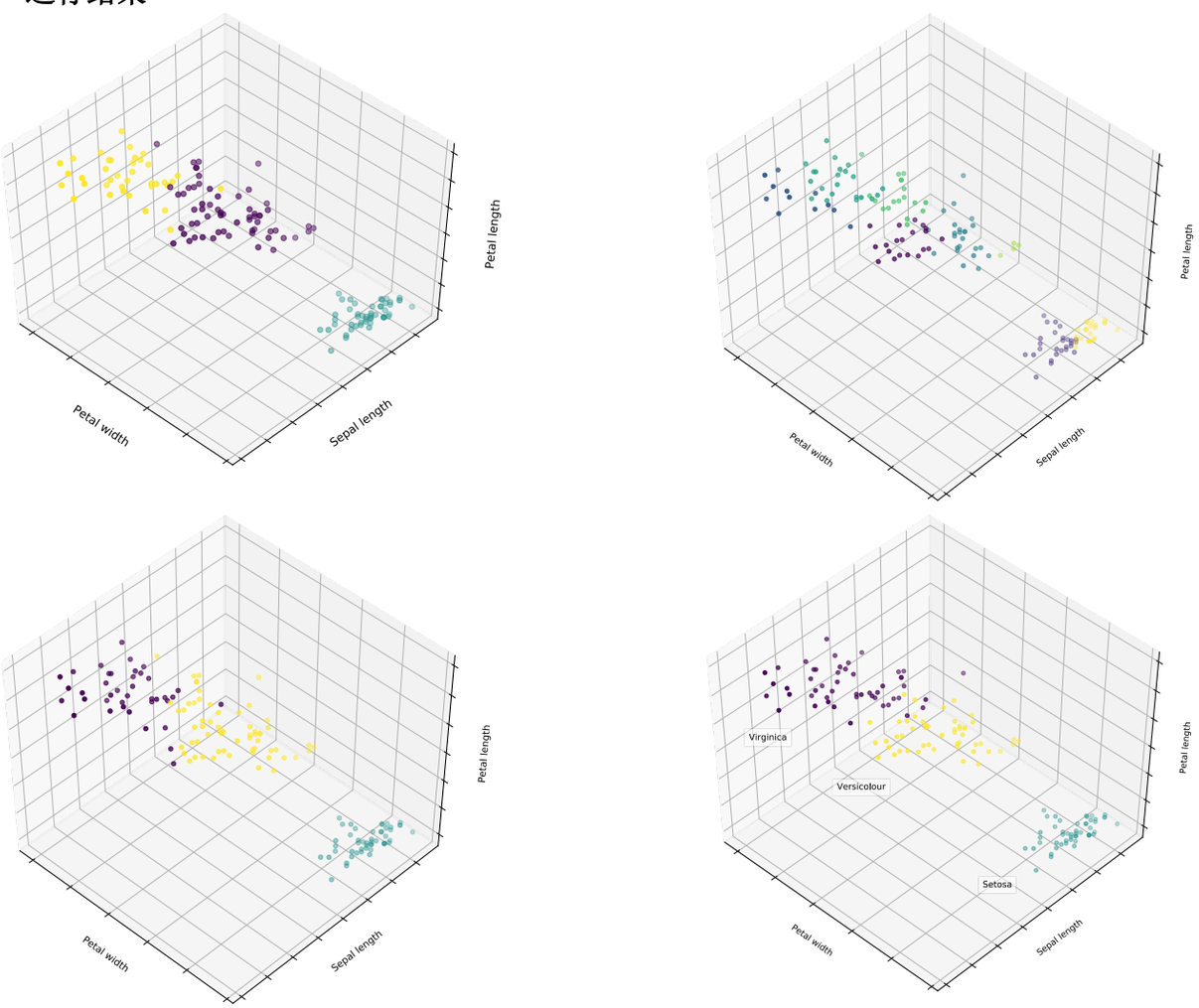
```
# Code source: Gaël Varoquaux
# Modified for documentation by Jaques Grobler
# License: BSD 3 clause

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D


from sklearn.cluster import KMeans
from sklearn import datasets

np.random.seed(5)

centers = [[1, 1], [-1, -1], [1, -1]]
iris = datasets.load_iris()
X = iris.data
```

```
y = iris.target

estimators = {'k_means_iris_3': KMeans(n_clusters=3),
              'k_means_iris_8': KMeans(n_clusters=8),
              'k_means_iris_bad_init': KMeans(n_clusters=3, n_init=1,
                                              init='random')}


fignum = 1
for name, est in estimators.items():
    fig = plt.figure(fignum, figsize=(4, 3))
    plt.clf()
    ax = Axes3D(fig, rect=[0, 0, .95, 1], elev=48, azim=134)

    plt.cla()
    est.fit(X)
    labels = est.labels_

    ax.scatter(X[:, 3], X[:, 0], X[:, 2], c=labels.astype(np.float))

    ax.w_xaxis.set_ticklabels([])
    ax.w_yaxis.set_ticklabels([])
    ax.w_zaxis.set_ticklabels([])
    ax.set_xlabel('Petal width')
    ax.set_ylabel('Sepal length')
    ax.set_zlabel('Petal length')
    fignum = fignum + 1

# Plot the ground truth
fig = plt.figure(fignum, figsize=(4, 3))
plt.clf()
ax = Axes3D(fig, rect=[0, 0, .95, 1], elev=48, azim=134)

plt.cla()

for name, label in [('Setosa', 0),
                    ('Versicolour', 1),
                     ('Virginica', 2)]:
    ax.text3D(X[y == label, 3].mean(),
              X[y == label, 0].mean() + 1.5,
              X[y == label, 2].mean(), name,
              horizontalalignment='center',
              bbox=dict(alpha=.5, edgecolor='w', facecolor='w'))
```

```
# Reorder the labels to have colors matching the cluster results
y = np.choose(y, [1, 2, 0]).astype(np.float)
ax.scatter(X[:, 3], X[:, 0], X[:, 2], c=y)

ax.w_xaxis.set_ticklabels([])
ax.w_yaxis.set_ticklabels([])
ax.w_zaxis.set_ticklabels([])
ax.set_xlabel('Petal width')
ax.set_ylabel('Sepal length')
ax.set_zlabel('Petal length')
plt.show()
```

程序代码 1

5.3 运行结果



六. 实验体会

本次实验作为先期实验，难度比较大。代码基本是从开源社区的支持者处获得。

鸢尾花数据作为数据知识获取界的 Hello World，比较适合用来进行分类，用在特征选择上不是很合适，因为其数据量太小了，而且维度也不高。[1]

七. 参考文献

[1] TAN P-N, STEINBACH M, KUMAR V. 数据挖掘导论 [M]. 2nd ed. 北京: 人民邮电出版社, 2011.