浙江工业大学

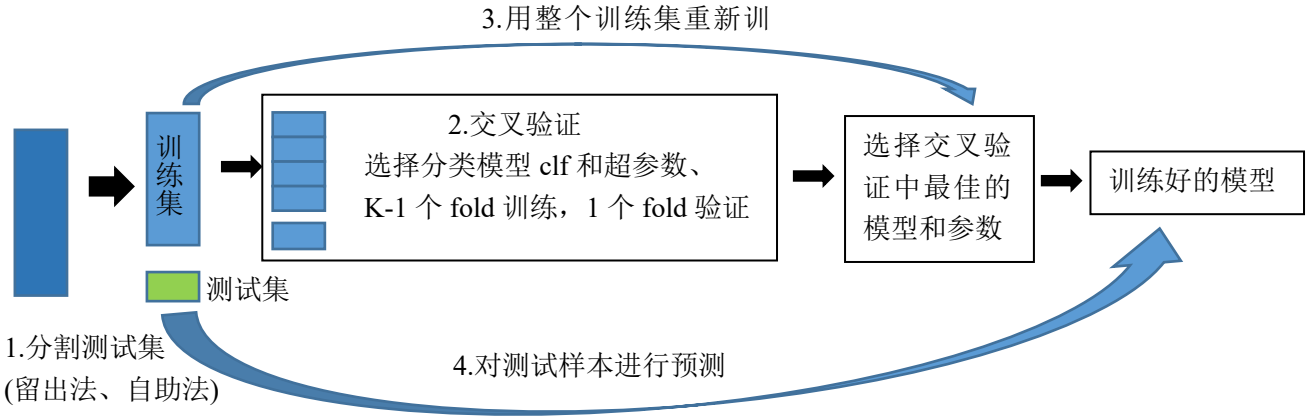
数据挖掘实验

计算机科学与技术学院

分类模型评估、k 近邻分类

一、实验目的

熟悉分类模型的评估和性能度量方法，掌握分类器调参和训练的一般步骤。



二、实验内容

1、编写以下性能度量函数

准确率（get_accuracy）、查准率（get_precision）、查全率（get_recall）、F1 度量（get_F1-measure）。以上每个函数输入(y, y_predicted)，两个参数分别为真实标签和预测出来的标签。

2、编程实现三种常用模型评估方法

2.1 编程实现分层采样的留出法，该函数按照给定比例从每个类别划分出测试集：X_train, y_train, X_test, y_test=StrateSplit(X,y,test_ratio,random_state)。4 个输入分别为数据矩阵，标签，测试集比例，和随机种子，返回训练集数据、对应标签，以及测试集数据和对应标签。

2.2 编写有放回采样 BoostTr 划分测试集。X_train, y_train, X_test, y_test=BoostTr(X,y,random_state)。该方法有放回采样 n 次得到训练集，n 为 X 中样本数目。未被采样到的样本为测试样本。

2.3 编写 k 折随机划分：folds_index=KfoldSplit(n_sample, k, random_state)。输入为训练集样本数据，fold 个数，以及随机种子，返回每个 fold 样本的 index。

3、 编程实现 k 最近邻分类

编写 k-nearestN 函数，输入训练数据 X_train,其对应标签 y,近邻数 k，测试集 X_test，以及距离度量，输出测试集中每个样本的标签。默认距离度量为欧式距离，其他可选距离包括余弦相似度等。

4、基于 breast_cancer 数据集的分类实验。

3.1 从 sklearn.datasets 导入数据，查看样本数、属性数、类别数、以及每个类别的大小，得到数据 X 和标签 y。通过观察类别大小，你发现什么？这种情况下 accuracy 是否是一种合适的度量，为什么？

3.2 调用 StrateSplit 以 20%划分出测试集,剩下的为训练集,设置随机种子为 42，即 X_train, y_train, X_test, y_test=StrateSplit (X,y,0.2,42)

3.3 调用 KfoldSplit，对训练集得到 k=10 个随机 fold 的样本索引。设置随机种子为 42。

3.4 基于以上划分，对 k-nearestN 进行交叉验证，即对每个 fold 轮流作为验证集,剩下为训练集,假设近邻数 k=2,输出 K 次的平均 precision, recall 和 F1。改变 k=4,对比 k=2 时 K 折交叉验证结果。

3.5 选择交叉验证结果更好的 k 值，用 3.2 中得到的训练集和测试集，得到测试集上的 precision, recall 和 F1，对比该 k 取值下交叉验证的结果。

随机采样可以用 pandas 的 sample 方法直接得到随机样本集，或者 numpy 的 random.choice 得到随机整数作为下标后再通过下标得到对应的数据。以下为对上面两种方法的简单例子和说明，更详细的用法自行查阅其他资料。

方法一：用 pandas 的 sample 方法直接得到随机样本集(**默认无放回**)

假设 x 为 pandas 的 DataFrame 对象

x.sample(n)：从 x 中随机抽取 n 个不同的记录

x.sample(frac=0.2)：从 x 中随机抽取 20%记录（不同）

x.sample(n,replace=True)：从 x 中采用有放回采样得到 n 个记录（可能包含重复记录）

方法二：用 numpy 的 random.choice 得到随机整数作为记录的下标（**默认有放回**）

np.random.choice(range(n1),n)：得到 n 个[0,n1)之间的随机整数

np.random.choice(range(n1),n,replace=False)

以下为整体框架伪代码(仅供参考)：

```
from sklearn import datasets,svm
# 导入 breast_cancer 数据
breastcan = datasets.load_breast_cancer()
#x 为对象-属性矩阵
x= breastcan.data
#y 为标签
y= breastcan.target
# 打印相关统计信息、是否有缺失值等
#第 1 步：把数据分成训练集 x_train 和测试集 x_test,以及对应的标签 y_train, y_test
。。。此处调用某种评估方法得到训练集合测试集的划分
X_train, y_train, X_test, y_test = StrateSplit(X, y, ratio)
#设置超参数
k=2
#第 2 步： 交叉验证进行模型和超参数选择
调用 KfoldSplit 把训练集 X_train 和对应 y_train 分成 k 个随机 fold
#对每个 fold,留出一个为验证集，其他为训练集进行分类器训练和验证
for …
        X_train_cv, y_train_cv, X_val, y_val=…
        # 得到验证集的预测结果
        y_pred = k-nearestN (X_train_cv, y_train_cv, k,X_val,’ Euclidean’ )
        # 计算 precision, recall 和 F1，然后对每个取 k 次的平均值
        p= get_precision(y_val, y_pred)
        r= get_recall(y_val, y_pred)
        f1= get_F1(y_val, y_pred)
p.mean()
r.mean()
f1.mean()
#换一个 k 值，重复上面步骤
#第 3 步：选择交叉验证 F1 最大的 k 值，用整个训练集（k 个 fold）的到测试集的结果
y_pred = k-nearestN (X_train, y_train, k, X_test,’ Euclidean’ )
p, r, f1..
```