COMP9517: 计算机视觉 2023 T2

实验室 4 规格 可达到的最高分:

2.5 分

本实验**占课程总分的2.5%**。

实验文件应在网上提交。 提交说明将在接近截止日期时公布。 **提交的截止日期是第7周，2023年7月10日星期一，18:00:00。**

目的：本实验重温第五周讲座中涉及的重要概念，旨在使你熟悉实现特定的算法。

材料：你需要使用Python 3+，TensorFlow2，和Scikit-learn。本实验室使用的数据集是狗与猫的数据集，可从Kaggle网站获得： https://www.kaggle.com/competitions/dogs-vs-cats/data
该数据集由25,000张图片的训练集和12,500张图片的测试集组成。这些图像是RGB-颜色格式，可能有不同的尺寸，但每张图像只属于两个类别中的一个：狗和猫。在这个实验室中，我们将只使用训练集，因为测试集的真实类别标签无法从网站上获得。

提交：该任务可在实验后进行评估。在上述截止日期前以Jupyter笔记本（.ipynb）的形式提交你的源代码，包括所有的输出（见下面的编码要求）。提交链接将在适当的时候公布。

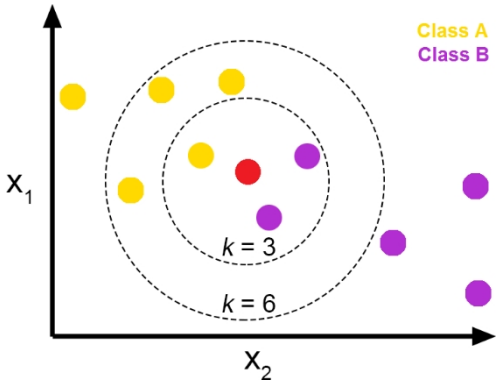
模式识别

本实验室的目标是实现并比较K-Nearest Neighbours（KNN）分类器、决策树（DT）分类器和随机梯度下降（SGD）分类器。下面我们在明确本实验室的任务之前，将对这些分类器进行简要介绍。

K-Nearest Neighbours (KNN)算法

KNN算法是非常简单和有效的。KNN的模型表示是整个训练数据集。通过在整个训练集中寻找最相似的K个实例（邻居）并总结这K个实例的输出变量，对一个新的数据点进行预测。对于回归问题，这可能是平均输出变量，对于分类问题，这可能是模式（或最常见）类值。诀窍在于如何确定

数据实例之间的**相似性**。



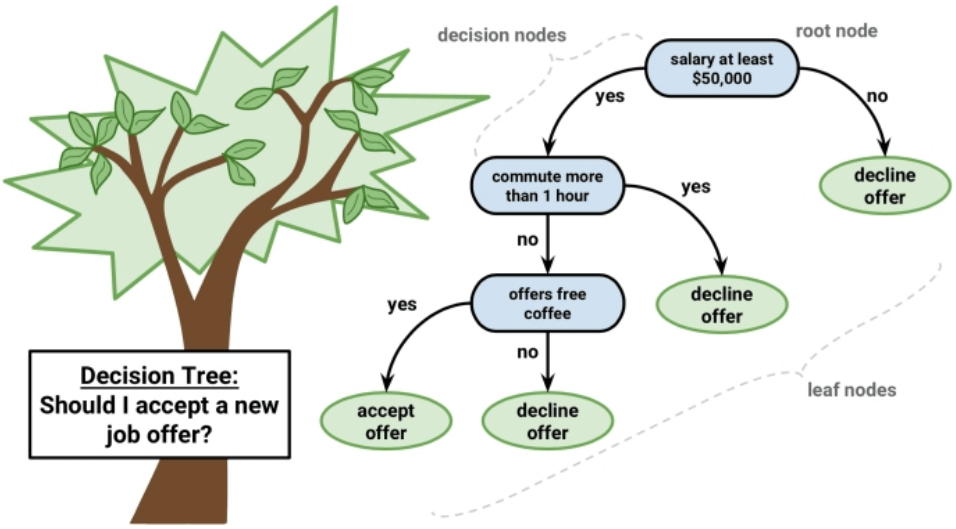
一个有3个和6个邻居的2类KNN例子（来自Towards Data Science）。

相似性：为了进行预测，我们需要计算任何两个数据实例之间的相似度。这样我们就可以在训练数据集中为测试数据集中的某个成员找到最相似的K个数据实例，进而做出预测。对于数字数据集，我们可以直接使用欧几里得距离测量。这被定义为两个数字数组之间的平方差之和的平方根。

参数：请参考Scikit-learn文档中的可用参数。
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

决策树（DT）算法

更多信息请参见https://en.wikipedia.org/wiki/Decision_tree_learning。



构建决策树的算法如下：

1. 选择一个要放在节点上的特征（第一个是根）。

2. 为每个可能的值做一个分支。

3. 对于每个分支节点，重复步骤1和2。

4. 如果一个节点的所有实例都有相同的分类，就停止开发树的这一部分。

如何确定在步骤1中分割哪个特征？一种方法是使用信息理论中的测量方法，如讲座中解释的**熵**或**信息增益**。

随机梯度下降（SGD）算法

更多信息见https://scikit-learn.org/stable/modules/sgd.html。

用不同的分类器进行实验

更多信息见https://scikit-learn.org/stable/modules/multiclass.html。还有更多的模型可供试验。下面是一个聚类模型的例子：

任务（2.5分）：在给定的数据集上进行图像分类。

开发一个程序，对狗与猫的数据集进行模式识别。使用KNN、DT和SGD三个分类器对图像进行分类，并比较分类结果。该程序应包含以下步骤：

设置

第1步：导入相关软件包

我们将在这个实验中主要使用**Scikit-learn**，所以在进入下一步之前，请确保你已经正确安装了

Scikit-learn库。你可以查看以下链接，了解更多关于该库和安装方法：

https://scikit-learn.org/stable/index.html

查看以下链接，了解如何导入KNN、DT和SGD分类器：
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html

第2步：加载数据集

在成功下载了狗与猫的数据集后，请熟悉它。具体来说，你可以检查整个数据集和每个类中有多少图像，每个图像的大小是多少。同时，显示每个类别的一些图像。我们鼓励你尝试对图像进行预处理，以改善你的分类结果。

第3步：提取数据集的一个子集

正如你所看到的，训练集里有25,000张图片（RGB-颜色格式），从文件名中可以看出正确的标签（狗与猫）。还有一个测试集，由另外12,500张图片组成，但是没有给出类别标签，所以我们将不使用这个集子。相反，我们把训练集分成一个用于实际训练的子集和一个用于测试的子集。

为了减少计算量，我们可以在完整的训练数据集的一个子集上工作。最初，将这个数据集分成10,000张图片用于训练，5,000张图片用于测试。在完成下面的所有步骤后，将训练图像的数量增加一倍到20000张，同时将测试图像的数量固定为5000张，并重复实验，看它是否对分类性能有影响。为了避免训练集中的类别不平衡，这可能会对性能产生负面影响，确保从狗类和猫类中取样的图像数量大致相等。

也可以参考Scikit-learn的内置函数`train_test_split()`，它可以自动洗刷数据集并帮助你分割数据：
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

第4步：为分类器进行必要的数据重塑

一旦你获得了数据集的一个子集，你需要重新塑造训练和测试数据，以便应用机器学习分类器。狗与猫数据集中的每张图片都应该有相同的大小，所以你需要调整它的大小。

分类

对每个分类器执行以下步骤：

第5步：初始化分类器模型

从Scikit-learn库中导入每个分类器后，你需要实例化（初始化）模型（分类器）对象。重要的是要阅读文档，找出初始化分类器模型时可以配置的各种参数。

第6步：将模型适合于训练数据

Scikit-learn库有一个拟合方法来从数据中学习。使用`fit()`方法，通过传递训练数据和训练标签作为参数来训练每个分类器。

第7步：在测试数据上评估训练好的模型

在你训练了一个分类器（也叫模型）之后，你可以用它来对测试数据进行预测。使用Scikit-learn库提供的预测（）方法。

评价

第8步：报告每个分类器的性能

为了量化训练好的分类器的性能，使用标准的分类指标，如**准确率**、**精确度**、**召回**率和**F1分数**。为了总结每个类别的正确和不正确的结果，使用**混淆矩阵**。Scikit-learn库提供了内置的方法，通过比较预测的标签和提供的地面真实标签来自动计算这些指标。点击以下链接，找到这些方法并导入：
https://scikit-learn.org/stable/modules/model_evaluation.html

对于每个分类器，在你的Jupyter笔记本中显示所有上述标准分类指标的值和混淆矩阵。同时，将你的分类器的准确性与狗与猫比赛的排行榜上的分数进行比较，并解释你的结果为什么更好或更差。
https://www.kaggle.com/competitions/dogs-vs-cats/leaderboard

编码要求

在你的Jupyter笔记本中，所有的单元格都应该已经被执行，这样导师/标记者就不需要再执行它们来查看结果。

参考文献

狗与猫的数据集：
https://www.kaggle.com/competitions/dogs-vs-cats/data

维基百科：K-Nearest Neighbors算法

https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

OpenCV-Python教程：K-Nearest Neighbour https://opencv24-python-tutorials.readthedocs.io/en/stable/py_tutorials/py_ml/py_knn/py_knn_index.html

走向数据科学：KNN（K-Nearest Neighbors）
https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d

SciKit-Learn: sklearn.neighbors.KNeighborsClassifier
https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

SciKit-Learn：在手写数字数据上的K-Means聚类演示 https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html

---

发布：2023年6月30日