第一次大作业

赵家兴

2019 年 12 月 21 日

一. kNN 算法优劣分析

（一）. kNN 的优势

1）易于实现和理解。

2）训练复杂度极低（实际操作中与数据输入与预处理相当），这使 kNN 可能用于数据的初步分析，然后再使用一些较重量级的工具。

3）可以解决非线性问题。

4）超参数量少（k 和距离函数）。

5）没有需要训练的参数。

（二）. kNN 的劣势

1）训练数据必须全程存储在内存里，特别是训练数据量很大（有时 kNN 的预测准确率会随着训练数据的增多和完备而提高）而平台要求很严格时，这会是相当大的障碍。

2）预测复杂度高（是训练数据数目的线性函数）

3）距离函数和基于距离比较的模型受噪声影响很大，如对 CIFAR-10 数据集，由于背景的影响（蓝色背景下的一只猫很可能因为训练数据中的一张包括海里的船的图潘而被预测为船）。

二. L1 与 L2 距离的好坏比较

此处的 python 实现代码已在本人 github 公开：https://github.com/TheLitFire/ML-Researchera/blob/master/Proj1-kNN/knn.py

（一）. L1 与 L2 距离的特性

数学表达式上容易看出，由均值不等式，对于相同的两个向量，L2 会给出不小于 L1 的距离。或者说，L2 距离比 L1 距离更“不能容忍”偏差。这种区分，使得 L1 距离在噪声较大的数据场合比 L2 更占优势（CIFAR-10），但相反地，如果数据已经提前经过了各种处理（去噪，中心化等），L2 的精确度会更高（MNIST）。

（二）. L1 与 L2 的效果比较

由本人程序对 MNIST 的预测成功率来看，**L2 距离所建模的 kNN，预测失败率为 2.95%，而 L1 距离的失败率为 3.67%**（二者均通过一定的方式选择了对其最好的 k 的取值，下面介绍）。数据集 MNIST 正是上文所描述的，其图片是主要数据居中的，只有黑白位图（而非三原色）的，噪声（比如各种背景等）相当小的数据。

以上预测结果可以从链接：https://rec.ustc.edu.cn/share/fd45bc80-240e-11ea-94cc-d70f05087a77 中获得（文件名前缀为 MNIST）。

然而，当情况来到了 CIFAR-10 数据集时，情况正好相反：**L1 距离的预测成功率约 38%，而 L2 距离比这个要低几个百分点**（由于没有测试数据的标准答案，以上成功率来自对验证集的预测，下面介绍）。事实上，由于 CIFAR-10 图片中背景色、同类物品的不同取景（卡车的左右照片）等影响，对偏差“更能容忍”的 L1 距离占据了上风。

三. k 的选取

对已有训练数据，从其中随机抽取 10000 个作为验证集，剩余数据为新的训练集（对于 MNIST，训练集有 50000 条数据，而 CIFAR-10 为 40000 个）。

随后，使用新的训练集训练模型，并预测验证集中数据的标签，和预测集的答案标签比较，对于不同的 k，分别计算成功率，并选出成功率最高的 k 作为本次的选择结果。

最后，多次重复这一随机选择、训练、预测、评估的流程，在多次获得的结果中找出出现次数最多的“最佳的 k”，作为最终的选择。

选择的结果为，对于 MNIST 数据集的 L1 和 L2 距离，k 均为 3；而对 CIFAR-10 而言，距离不论 L1 或 L2，k 均选择 1。

值得说明的是，可以通过测试见得，在上述训练集数据量约数万条的情况下，k 超过几十、达到上百时，k 越大，预测准确率就越低。所以实际操作中，本人 k 的待选范围也不过几十。

此处的选择程序可见：https://github.com/TheLitFire/ML-Researchera/tree/master/Proj1-kNN/supporting%20codes