lec09

k-NN

k 是这个算法的超参数。通常来说，更大的数据集应该拥有更大的 k；而对小数据集使用较大的 k 可能会超出类边界

训练集、测试集、验证集

在测试集外、训练集上划分一部分作为验证集，验证集的作用是设置超参数，减少过拟合

Cross-validation（交叉验证）

交叉验证有非常多的方法，她 PPT 只介绍了 *k-折交叉验证*

k-折交叉验证：数据集被随机分为k个大小相等的子集（称为“折”），每次留出一个子集用作测试集，而其余的k-1个子集合并用作训练集；这个过程重复k次，每个子集都有一次机会用作测试集。最终的模型性能是这k次测试结果的平均值。

k-NN 的复杂度

denotes:
n 个对象，d 个特征

训练

$O(1)$ 时间：它不需要进行显式的模型训练，只是简单地存储训练数据
$O(n \times d)$ 空间：即训练集的大小

分类

- 距离计算通常涉及到所有维度，每次距离计算的复杂度约为 $O(d)$
- 如果没有使用任何优化数据结构如 k-d 树或球树、ANN等，那么在最坏的情况下，找到最邻近的 k 个邻居的复杂度是 $O(n)$

因此，总复杂度为 $O(n \times d)$

k-NN 的 inductive bias 和 feature importance

归纳偏置

k-NN 分类器假设空间中邻近的点应具有相同的标签。这意味着分类器预期在数据集的同一区域中的样本将属于同一类别，这基于邻近性原则

特征重要性

k-NN 分类器默认所有特征具有同等重要性。这个假设可能导致性能问题，尤其是在存在许多无关特征的数据集中。如果数据集中只有少数几个特征是相关的，而其他许多特征都是无关的，k-NN 分类器很可能表现

不佳。这是因为无关的特征会干扰对邻近点的准确计算和判断

特征缩放：高斯归一化（Gaussian Normalization）

特征缩放目的是使得各个特征的尺度（scale）一致

高斯归一化:
数据集中的每一个特征 i，计算该特征的样本均值 μ_i 和样本标准差 σ_i
对于数据点 $X = (x_1, \ldots, x_d)$，每个特征 x_i 都会通过下面的特征进行转换：

$$\hat{x_i} = \frac{x_i - \mu_i}{\sigma_i}$$

$$\hat{X} = \left(\frac{x_1 - \mu_1}{\sigma_1}, \ldots, \frac{x_d - \mu_d}{\sigma_d} \right)$$

- 转换后数据将**围绕 0 居中**，每个特征都将具有零均值和单位标准差
- 所有特征现在具有相同的尺度，具有了**可比性**，无需考虑各特征的原始尺度
- 可以防止高尺度特征对结果产生过大影响，从而提高算法性能

k-NN 总结

- 数据预处理：对数据集中的特征进行规范化，使特征具有零均值和单位方差
- 高维数据处理：如果数据维度非常高，考虑使用降维技术
- 数据集分割：将训练数据分割成训练集（占 50%-90%）和验证集（占 10%-50%）
- 使用交叉验证：如果验证数据集太小，可以将训练数据分成多个部分，进行交叉验证
- 训练与评估：在验证数据集上训练并评估 k-NN 分类器，尝试不同的 k 值和不同类型的距离度量（如 L1 范数和 L2 范数）