kNN 算法实验报告

卢珍妮

摘要： k 近邻法(k-nearest neighbor, k-NN)是一种基本分类与回归方法。k 近邻法假设给定一个训练数据集，其中的实例类别已定。分类时，对新的实例，根据其 k 个最近邻的训练实例的类别通过多数表决等方式进行预测。k 近邻法实际上利用训练数据集对特征向量空间进行划分，并作为其分类的“模型”。k 值的选择、距离度量及分类决策规则是 k 近邻法的三个基本要素。本文通过对 kNN 算法的实验，详细介绍 kNN 算法的思想、原理以及具体的代码实现。

关键词： k 近邻算法；kNN 算法；分类；基本要素

1 简介

k 近邻(kNN，k-NearestNeighbor)分类算法是数据挖掘分类技术中最简单的方法之一。所谓 K 近邻，就是 k 个最近的邻居的意思，说的是每个样本都可以用它最接近的 k 个邻居来代表。

kNN 算法的核心思想是如果一个样本在特征空间中的 k 个最相邻的样本中的大多数属于某一个类别，则该样本也属于这个类别，并具有这个类别上样本的特性。该方法在确定分类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。kNN 方法在类别决策时，只与极少量的相邻样本有关。由于 kNN 方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因此对于类域的交叉或重叠较多的待分样本集来说，kNN 方法较其他方法更为适合。

2 算法

2.1 算法流程

对每一个未知点执行：

1.计算未知点到所有已知类别点的距离

2.按距离排序（升序）

3.选取其中前 k 个与未知点离得最近的点

4.统计 k 个点中各个类别的个数

5.上述 k 个点里类别出现频率最高的作为未知点的类别

2.2 三要素

2.2.1 距离度量

1.Lp 距离：

$$L_p(x_i, x_j) = \left(\sum_{l=1}^{n} |x_i^{(l)} - x_j^{(l)}|^p \right)^{\frac{1}{p}} \quad (1)$$

2.欧式距离：

$$d_{euc}(x,y) = [\sum_{j=1}^{d} (x_j - y_j)^2]^{\frac{1}{2}} = [(x-y)(x-y)^T]^{\frac{1}{2}} \quad (2)$$

3.曼哈顿距离：

$$d = |x_1 - x_2| + |y_1 - y_2| \quad (3)$$

2.2.2 k 值的选择

如果选择较小的 k 值，“学习”的近似误差会减小，只有与输入实例较近的(相似的)训练实例才会对预测结果起作用。但缺点是“学习”的估计误差会增大，预测结果会对近邻的实例点非常敏感。如果邻近的实例点恰巧是噪声， 预测就会出错.换句话说，k 值的减小就意味着整体模型变得复杂，容易发生过 拟合。

如果选择较大的 k 值，正好相反，k 值的增大就意味着整体的模型变得简单.

如果 k=N，那么无论输入实例是什么，都将简单地预测它属于在训练实例 中最多的类。这时，模型过于简单，完全忽略训练实例中的大量有用信息，是不 可取的.

在应用中，k 值一般取一个比较小的数值.通常采用交叉验证法来选取最优的 k 值.

2.2.3 分类决策规则

K 近邻法中的分类决策规则往往是多数表决，即由输入实例的 k 个邻近的训练实例中的多数类决定输入实例的类。

2.3 算法优缺点

kNN 算法简单有效、易理解。但由于需要保存全部数据集，因此对内存消耗大，当数据集较大时对设备要求非常高。同时需要计算每个未知点到全部已知点的距离，可能会很耗时。kNN 算法是懒惰学习方法，一些积极学习算法要快很多。计算量较大。目前常用的解决方法是事先对已知样本点进行剪辑，事先去除对分类作用不大的样本。该算法在分类时有个主要的不足是，当样本不平衡时，如一个类的样本容量很大，而其他类样本容量很小时，有可能导致当输入一个新样本时，该样本的 K 个邻居中大容量类的样本占多数。该算法只计算“最近的”邻居样本，某一类的样本数量很大，那么或者这类样本并不接近目标样本，或者这类样本很靠近目标样本。无论怎样，数量并不能影响运行结果。可以采用权值的方法（和该样本距离小的邻居权值大）来改进。

3. 实验

实验部分，使用鸢尾花数据集，距离度量采用欧式距离，分类决策规则使用多数表决规则。实验通过设置不同的 k 值，比较对测试数据集预测的准确率。

3.1 算法流程

kNN 算法
输入：训练数据集，测试数据集，K 值 输出：测试数据集的标签类型，准确率 步骤： 1.读取数据：从 txt 文件中读取数据，并将字符串类型转换为数字。 2.计算距离：计算两个数据实例之间的距离。 3.确定 K 近邻：确定最相近的 k 个实例。 4.预测结果：根据 k 个实例的类别，生成预测结果。 5.准确度：计算预测的准确度。

3.2 实验数据与设置

3.2.1 数据集

鸢尾花数据集由对 3 个不同品种的鸢尾花的 150 组观察数据组成。对于这些花有 4 个测量维度：萼片长度、萼片宽度、花瓣长度、花瓣宽度，所有的数值都以厘米为单位。需要预测的属性是品种，品种的可能值有：清风藤、云芝、锦葵。本文选取 104 项数据作为训练集，46 项数据作为测试集，通过计算测试数据集与训练数据集的所有实例的 4 个属性的欧式距离，选取前 k 个欧式距离最小的数据，根据多数表决规则确定测试数据集的类别。

3.2.2 评价指标

为了验证 k 值的选择对于算法的影响，通过设置不同的 k 值，比较准确率的不同。表 1 展示了不同 k 值下 kNN 算法的准确率，从实验结果可以看出，随着 k 值的增大，准确率增大。但当 k 为 40 时，准确率降低。选择较大的 k 值，使得与测试集数据实例较远的训练数据也对预测产生了作用，使得预测错误，导致准确率降低。因此，k 值的选择对于 kNN 算法的性能至关重要。

表 1.展示了不同 k 值下 kNN 算法的准确率

K 值	准确率
1	93.48%
5	95.65%
10	95.65%
20	97.83%
40	91.30%

4 总结

本文对 kNN 算法的思想，基本要素以及优缺点进行了介绍，在鸢尾花数据集上实现了 kNN 算法，通过选择不同的 k 值，对比准确率的变化，从而发现 k 值的选择对于 kNN 算法的性能至关重要。