

模式识别 KNN

22009200601 汤栋文

2024 年 10 月 31 日

目录

摘要	2
1. 方法	2
1.1 数据集	2
1.2 验证方式	2
2. 实验	2
1.1 将原数据直接进行 KNN	2
1.2 将原数据降维后进行 KNN	3
3. 结论	3

摘要

本研究使用 MNIST 和 CIFAR10 数据集，评估了 K 最近邻（KNN）算法在不同数据集上的分类性能，并在 GPU 并行计算的情况下，评估了计算成本。

1. 分析了 KNN 算法在多个数据集上的准确率。
2. 分析了 KNN 算法在不同规模数据集和距离度量下的计算成本。

1. 方法

1.1 数据集

本研究使用了 MNIST 和 CIFAR10 数据集。MNIST 数据集包含了 70,000 个手写数字的灰度图像（其中训练集 60,000 个，测试集 10,000 个），每个图像的大小为 28x28 像素，标记为 0 到 9 的数字类别。CIFAR-10 数据集包含了 60,000 张彩色图像（其中训练集 50,000 张，测试集 10,000 张），每张图像的大小为 32x32 像素，共有 10 个类别，包括飞机、汽车等。

为了消除特征间的量纲影响，我们对原始图像数据进行了标准化处理，即对每个像素值减去平均值并除以标准差，使得每个特征（像素值）具有零均值和单位方差。

1.2 验证方式

使用 MNIST 和 CIFAR 的官方数据集划分，划分为训练集和测试集。使用 KNN 算法分类后评估准确率。以及在相同的代码下的运行时间。

2. 实验

1.1 将原数据直接进行 KNN

MNIST 数据集

K	acc (%)
1	96.9
3	97.2
10	96.8
50	95.4
200	92.9

CIFAR10 数据集

K	acc (%)
1	35.4
3	35.6
10	34.7
50	32.7
200	29.7

metrics	acc (%)	time (s)
Euclidean	97.2	29.0
Manhattan	96.4	6.4
Cosine	97.4	21.9

metrics	acc (%)	time (s)
Euclidean	35.6	87.2
Manhattan	39.2	15.2
Cosine	37.7	58.7

1.2 将原数据降维后进行 KNN

CIFAR10 (K=3)

method	dim	acc (%)	time (s)
PCA	32	40.5	4.4
PCA	128	38.5	7.3
PCA	1024	35.8	30.3
LDA	8	31.0	11.8

结论：

1. KNN 算法简单直接，分类有效能够在简单的数据集上得到不错的结果。
2. KNN 算法的时间消耗较多，特别是当数据量较大时。

3. 结论

本研究表明，KNN 算法在简单数据集上表现良好，能够得到较高的分类准确率。然而，KNN 算法的计算时间较长，特别是在数据量较大时。我们在不同的距离度量下测试了 KNN 算法，发现欧几里得距离和余弦距离在 MNIST 数据集上的表现较优，而在 CIFAR10 数据集上曼哈顿距离的准确率更高。利用 PCA 和 t-SNE 对数据进行降维处理后，KNN 算法的准确率有所下降，但计算时间明显减少。综上所述，KNN 算法在某些场景下具有实用性，但其计算时间需在实际应用中加以考虑。