数据科学与工程学院

统计方法与机器学习课程实验报告

基于降维算法的 K 近邻算法手写数字识别

2021 年 6 月 13 日

摘要

如何高效处理和传输海量图像数据是大数据处理中的一个重要问题，而图像压缩技术是其中必不可少的一环。主成分分析（PCA）算法是一种用于数据降维的算法，其在数据压缩，消除冗余和数据噪音消除等领域都有较为广泛的的应用。其核心思想是在减少数据集的维数的同时保留数据集当中对方差贡献最大的特征，而这是通过保留低维主成分，忽略高维次要成分做到的。这样在低维空间的成分往往能够保留住数据的最重要部分。在本实验中，我们使用该算法对 MNIST 手写数字数据集进行数据降维，再使用 K 近邻算法进行分类。本实验在 Python3.7(Jupyter-Lab) 环境下进行，并且对一些重要指标与分类精度其进行了可视化展示。

关键词: 降维，KNN，手写数字识别，图像分类

目录

第 1 章　项目概述与数据概览

本实验的目标在于使用主成分分析算法对 MNIST 手写数字数据集进行压缩降维，随后使用 K 近邻算法对手写数字数据集进行分类，再分析降维后产生的信息损失对分类精确度造成的影响。

1.1　主成分分析

主成分分析算法（Principal Components Analysis）最主要的是应用在于对数据集进行维数约减（Dimensionality Reduction）。PCA 算法流程为输入一个 n 维数据集 $X = (x_1, x_2, \ldots, x_m)$，将数据降维到的维数 n' 后再输出降维后的样本集 X'。如图 1.1 所示，执行 PCA 算法时，首先对所有的样本进行去中心化 $x_i = x_i - \frac{1}{m}\sum_{j=1}^{m} x_j$，再计算样本的协方差矩阵 $CovMat = \frac{1}{m}XX^{\top}$，接着求出协方差矩阵的特征值及对应的特征向量。最后将特征向量按对应特征值大小排列成矩阵，取矩阵的前 k 行（行数取决于维度约的程度）组成矩阵 P，$X' = PX$ 即为降低维度后的数据。

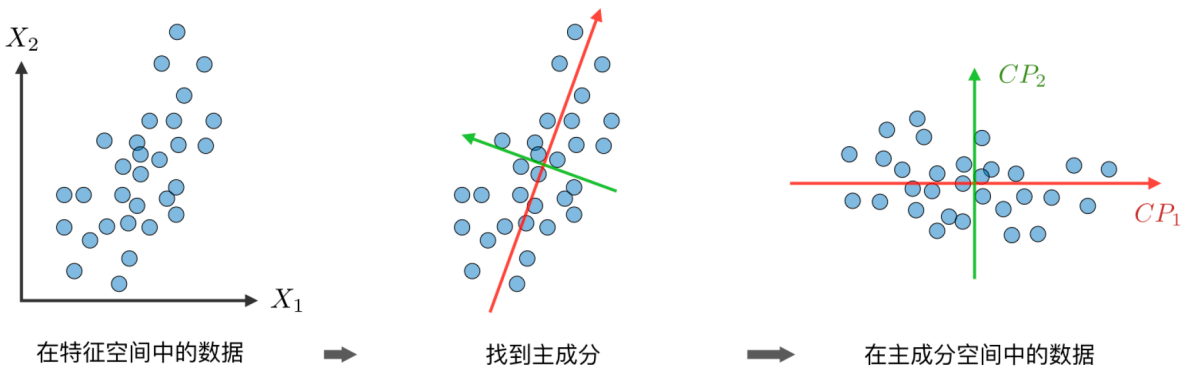

图 1.1: 主成分分析算法的执行过程

1.2　K 近邻算法

K 近邻算法 (K- Nearest Neighbor) 是一种经典的分类算法，如图 1.3 所示

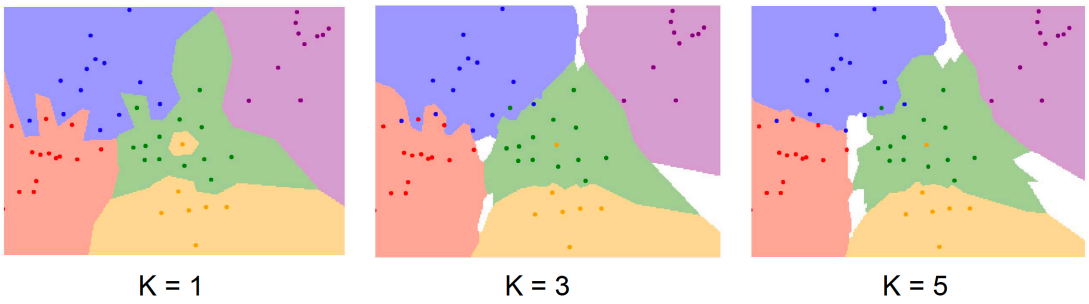

图 1.2: K 近邻算法

KNN 算法（K-近邻算法）是一种用于分类和回归的非参数统计方法。在这两种情况下，输入包含特征空间中的 k（k 为超参数）个最接近的训练样本。KNN 最大的特点是它是一种惰性算法，只记忆用于训练的数据，而非提取特征，故其训练阶段时间复杂度为 O(1)。KNN 决策流程的重要指标为测试数据点与训练集数据点的距离，但因 KNN 只单纯 1 记忆数据，故其预测阶段时间复杂度为 O(n)。使用 k-d 树等算法能优化 KNN 分类器的预测时间开销。

1.3 MNIST 手写数字数据集

图 1.3: MNIST Examples

该数据集可在 http://archive.ics.uci.edu/ml/ 获得，也可从 Kaggle 上的 MNIST in CSV 页面直接下载 csv 文件。

第 2 章　算法

2.1　主成分分析算法流程

主成分分析算法的核心思想是数据降维，通过最大化数据的方差而将数据投影到低维度上，步骤如下所示:

(1) 规范化输入数据，先取列均值 $\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$，对数据进行去中心化可得 $x_j^{(i)} \leftarrow \frac{x_j^{(i)}-\mu_j}{\sigma_j}$ 和 $\sigma_j^2 = \frac{1}{m}\sum_{i=1}^{m}\left(x_j^{(i)}-\mu_j\right)^2$

(2) 计算协方差矩阵 $\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}x^{(i)T} \in \mathbb{R}^{n\times n}$

(3) 计算协方差矩阵的主特征值和相应特征向量 $u_1, \ldots, u_k \in \mathbb{R}^n$

(4) 将数据投影到 $\mathrm{span}_{\mathbb{R}}(u_1, \ldots, u_k)$，最大化 k 维空间的方差

2.2　K 近邻算法流程

设训练数据集 T（MNIST 数据集中有 60,000 个训练样本）

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)\}$$

其中，$x_i \in \mathcal{X} \subseteq \mathbf{R}^n$ 为实例的特征向量，$y_i \in \mathcal{Y} = \{c_1, c_2, \cdots, c_K\}$ 为实例的类别，$i = 1, 2, \cdots, N$;

该算法输出为实例 x 所属的类 y，在 MNIST 数据集分类问题中，共 10 个类，算法执行流程如下:

(1) 根据给定的距离度量（如 L2 距离），在训练集 T 中找出与 x 最邻近的 k 个点（k 为超参数），涵盖这 k 个点的 x 的邻域记作 $N_k(x)$;

(2) 在 $N_k(x)$ 中根据分类决策规则（如多数表决）决定 x 的类别 y;

$$y = \arg\max_{c_j} \sum_{x_i \in N_k(x)} I(y_i = c_j), \quad i = 1, 2, \cdots, N; j = 1, 2, \cdots, K$$

k 近邻法的特殊情况是 $k = 1$ 的情形，称为最近邻算法。对于输入的实例点/特征向量 x，最近邻法将训练数据集中与 x 最邻近点的类作为 x 所在的的类，在后续实验的比较中我们可以看到不同邻居数量对分类精确度造成的影响。

第 3 章　实验过程与结果

本实验的过程主要包括数据的读取、数据降维、机器学习训练模型、结果分析和数据的可视化展示等几个主要步骤。这部分代码请见 *Project*1.*ipynb* 文件。

3.1　图像数据读取和可视化展示

我们从手写数字数据集中抽取一部分数据进行可视化展示，原始数据的大小为 28*28，。虽然是黑白图片，但是在计算距离时依然会有大量的时间开销，我们可以通过降维算法来对时间开销进行压缩。手写数字样本如图 3.1 所示。

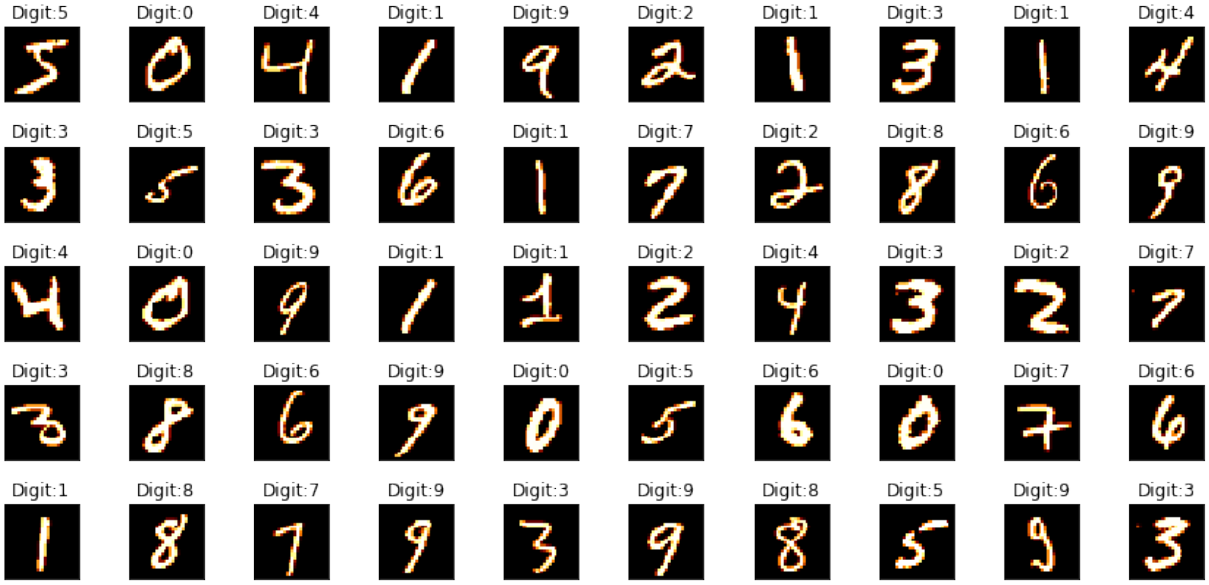

图 3.1: MNIST 数据展示

3.2　主成分可视化展示

我们先执行 PCA 算法，对其特征值分布进行可视化展示，如图 3.2 所示。随着特征数量选择的变多，其在二维空间的分布逐渐由简单变复杂。

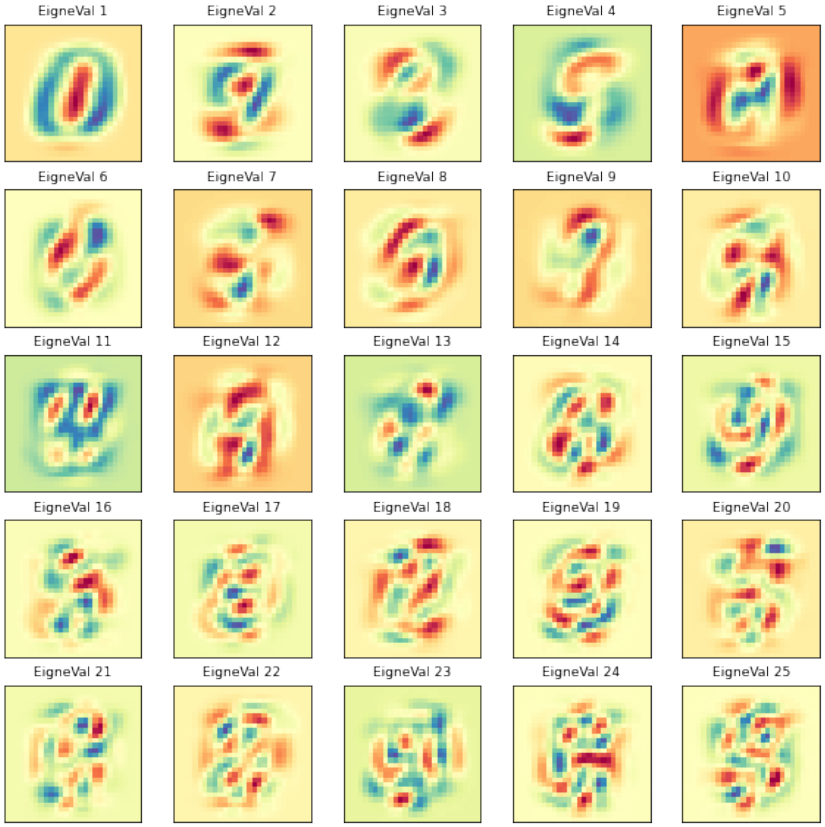

图 3.2: Eigne Value Visualization

随后降维 MNIST 手写数字数据集中的数据至二维，随后进行可视化展示（这部分用于降维可视化的代码在 visualize.py 文件中），可以看到仅保留两个主成分的数据点在 xy 平面上的分布。

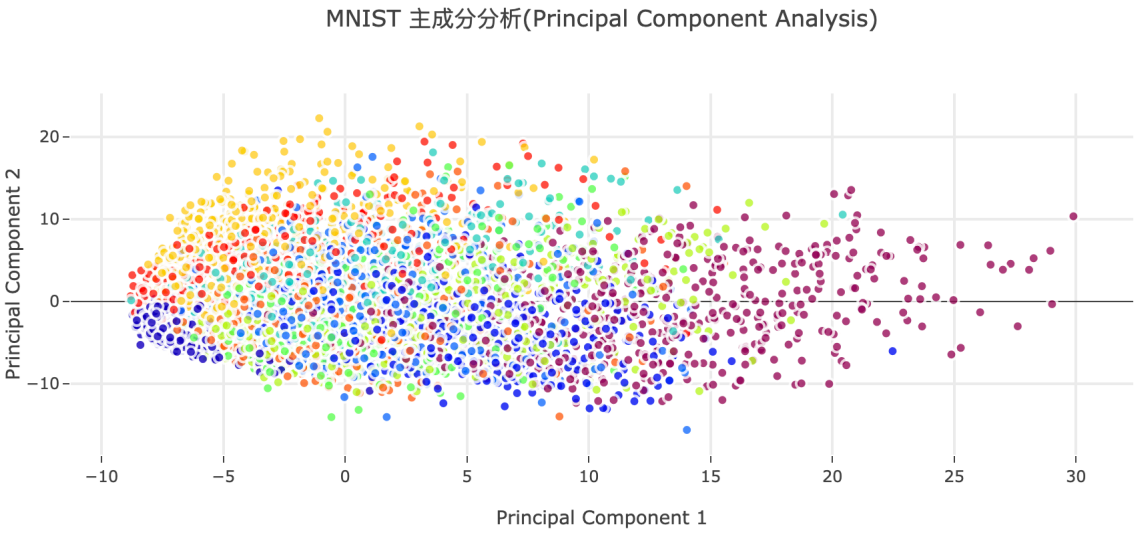

图 3.3: MNIST PCA Visualization

3.3 其他降维算法可视化展示

除了主成分分析算法外，还有另外两种常用的降维算法可供选择，分别是 LDA 降维和 t-SNE 降维算法。此部分代码存放于 *visualize.py* 文件中。

3.3.1 LDA 降维算法

LDA 算法的思想是将高维数据投影到低维空间之后，使得同一类数据尽可能的紧凑，不同类的数据尽可能分散，这是一种监督学习算法。

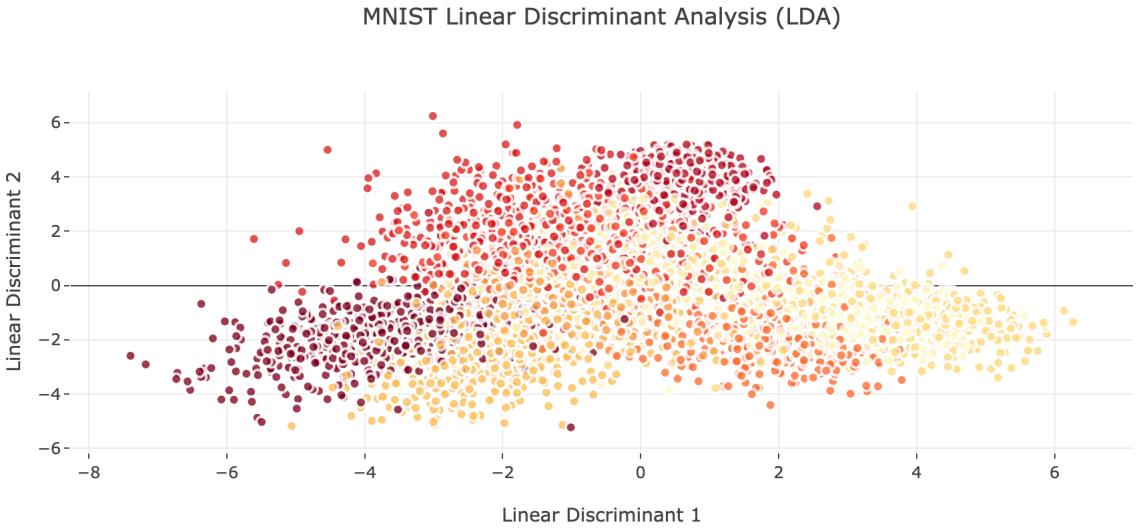

图 3.4: MNIST LDA Visualization

3.3.2 t-SNE 降维算法

使用与上述两种降维方法同样的可视化手段，我们可以发现 t-SNE 降维算法能将降维后的数据点在二维平面上表达的非常清楚。t-SNE 设计的初衷就是用于在二维或三维的低维空间中表示高维数据集，从而使其可视化。

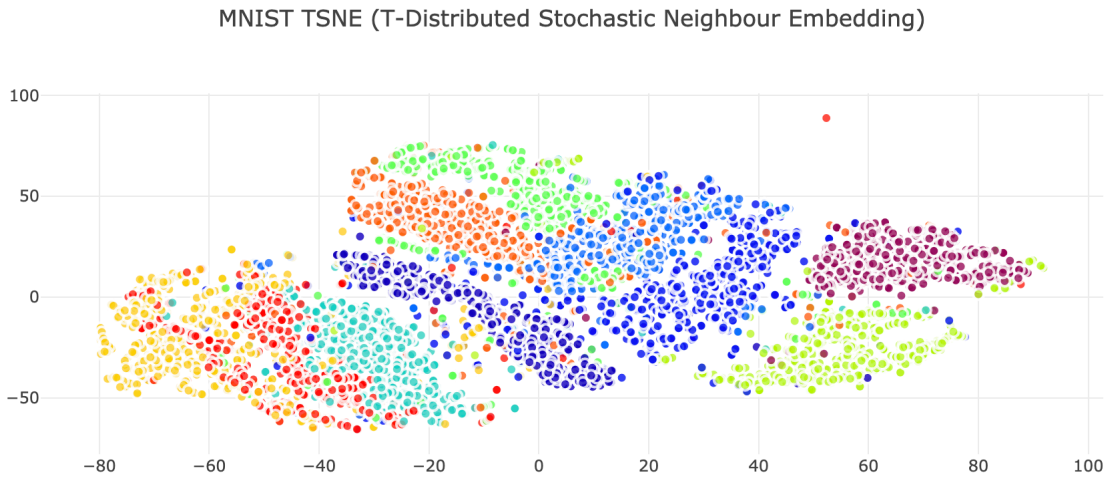

图 3.5: MNIST t-SNE Visualization

当我们想要对高维数据进行分类，又不清楚这个数据集的可分性如何时，可以通过 t-SNE 投影到 2 维或者 3 维的空间中先进行观察。如果在低维空间中具有可分性，则数据显然是可分的；如果在高维空间中不具有可分性，可能是有一部分高维数据不可分，需要将其降维（但可能会造成信息损失）。

3.4 K 近邻算法分类结果

3.4.1 主成分个数对分类算法结果的影响

主成分分析的重要思想是保留对信息贡献最大的特征，在主成分个数对分类精度影响的实验中，可以看到仅仅少量主成分就可以达到接近原始数据分类精度的效果。如图 3.6 所示，选取 15 个主成分后，再添加主成分就不再对算法精度有特别大的影响，可以看到分类的准确率始终保持在 96.5% 左右

值得注意的是，我尝试了对 MNIST 未经降维处理的原始数据进行 KNN 分类，随后发现降维后的数据集在分类的准确率上表现会偶尔略微超过原始数据集。这比较反直觉，因为一般而言降维会带来信息损失，导致分类问题的准确度下降。

我个人给出的解释是：MNIST 数据的原始分布在欧氏距离空间上的差异并非特别显著，在 PCA 降维的过程中，是以样本降维后的方差最大为目标进行降维的，也就是样本间降维后的平均距离变大了，这个过程把各个样本在欧式距离空间上的差异放大了，故导致了样本之间的差异变大，使得 KNN 获得了更好的效果。

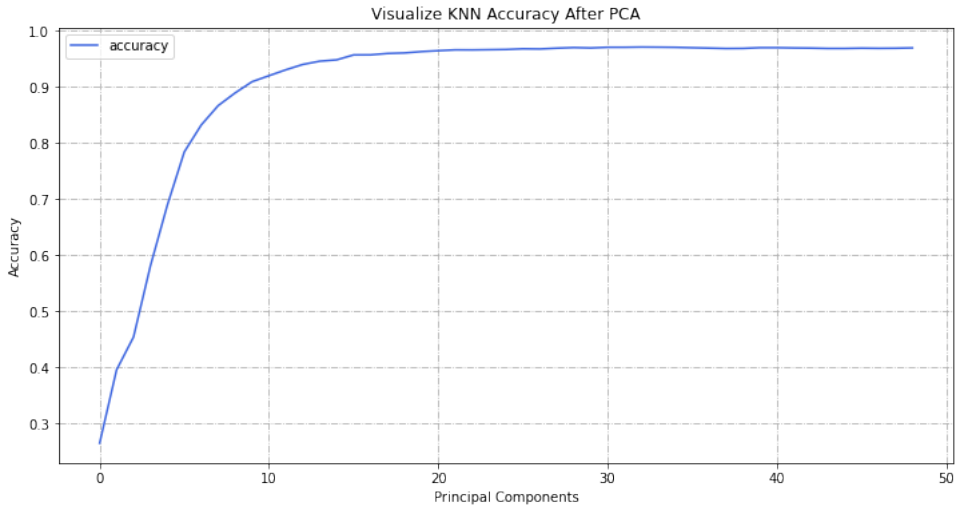

图 3.6: 主成分个数对分类算法结果的影响

3.4.2 邻居数对分类算法结果的影响

在实验开始时提到了对于 KNN 算法而言，邻居数 K 是对算法性能影响显著的一个超参数，如图 3.7 所示，我们可以看到从 1 个邻居（最近邻）算法到选取 12 个邻居时算法在测试集上的性能差异（由于对整个原始数据集进行交叉验证的时间开销比较大，此处使用 PCA 降维到 25 个特征进行预测）。

可以看到，在邻居数上升的初期，算法性能有一定的提升，但在 K 大于 4 后，算法的性能趋于稳定。

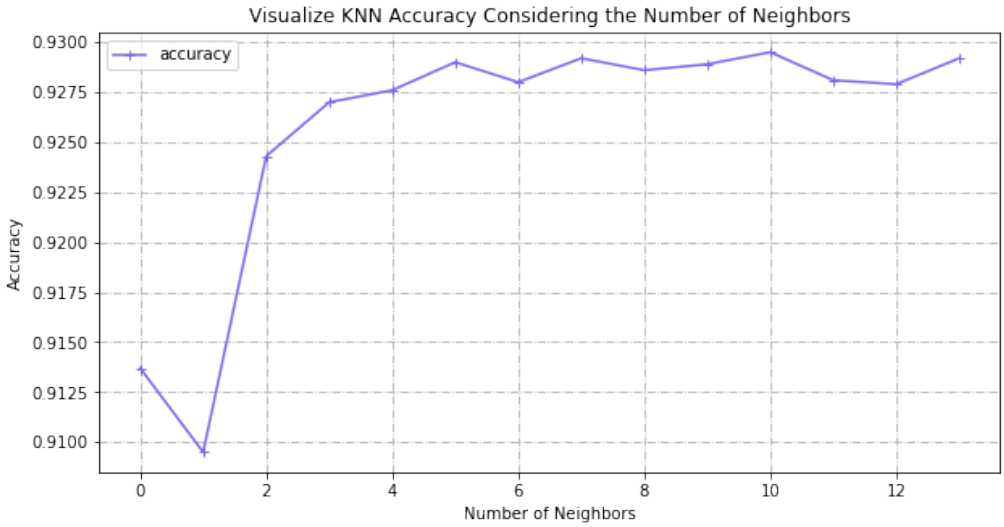

图 3.7: 超参数 K 对分类算法性能的影响

3.5 降维对算法时间开销的影响

实验开始时提到，KNN 是惰性分类器，没有显式的学习过程，其单纯记忆训练数据，而非提取特征后建立一个模型。这就导致了预测复杂度为 O(n)，在大样本情况下的预测速度非常慢。

而主成分降维后，算法的时间开销被明显缩短了，我们依次选择 1-50 个成分进行预测，如图 3.8 可见，完成对测试集所有样本的预测的时间，由不降维的原始数据开销约 5min 降低到了 30s 内，仅保留 10 个成分的情况下，只需要约 3s 就可以完成预测，这个提升是非常显著的，尤其是在 PCA 并不会很大程度上影响算法精度的情况下。

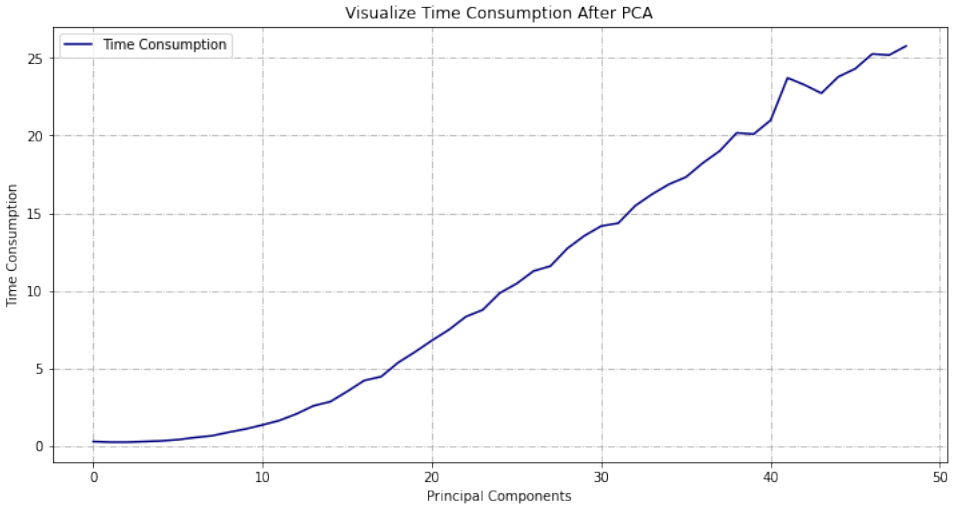

图 3.8: 特征数量对 KNN 算法时间开销的影响

第 4 章　参考资料

(1)《统计学习方法（第二版）》李航. 清华大学出版社

(2) Stanford cs229-Lecture11: Unsupervised Learning

(3) Stanford cs231n-Lecture2: Image Classification Pipeline

(4) Visualizing data using t-SNE, Laurens van der Maaten and Geoffrey Hinton