K-MEANS聚类 实验报告

算法原理

步骤

对于N个数据点$\{\mathbf{x}_i\}$，K-MEANS算法采用以下步骤进行聚类：

- 给定类别个数K
- 初始化K个中心点$\{\mathbf{c}_i\}$
- 迭代直至中心点不再变化：
  - 对于每个k，簇$C_k = \{i|\mathbf{x}_i\text{距离}K\text{个中心点中的}\mathbf{c}_k\text{最近}\}$
  - 对于每个k，更新

$$\mathbf{c}_k = \frac{1}{|C_k|}\sum_{j\in C_k}\mathbf{x}_j$$

优化目标与收敛性

假设使用向量之差的范数$||*||$作为距离的度量，则上述步骤的优化目标为

$$\min_{\{C_k\}_{k=1}^K}\sum_{k=1}^K\sum_{i\in C_k}||\mathbf{x}_i-\mathbf{c}_k||^2 \quad s.t.\ \mathbf{c}_k=\frac{1}{|C_k|}\sum_{j\in C_k}\mathbf{x}_j$$

上述步骤中，每次迭代都会使优化目标非增，由于优化目标有界，且数据点有限，一定会收敛至某个局部最优点。

复杂度

每次迭代的时间复杂度均为$O(NK)$，总复杂度还取决于迭代次数和计算距离时的时间开销

实现细节

簇数K的选择

由于MNIST数据集共有10类，理想的$K=10$，也就是每个簇恰好对应一类。但在实现时，由于特征选择、距离度量的局限性，依靠距离往往很难将各类完美分开，这样若选择恰好10簇，误分类会很普遍。在$K=10$的基础上，我还进一步尝试了更大的K。

初始中心点的选择

我尝试了两种初始化方法：

- random：随机选取初始中心点
- representation：在每一类中，随机选取一个代表样本点作为初始中心点

特征向量的提取

我尝试了两种特征：

- raw：将28*28的图像直接展平为长度为784的一维向量作为特征向量
- hog：使用长度为36的HOG（方向梯度直方图）特征，通过库`skimage.feature`进行提取

距离的度量

经过尝试，∞范数效果较差，1范数收敛速度则不如2范数，因此我最终选择了2范数作为距离的度量

何时停止迭代

在我尝试的K值范围内（小于100），算法大多在200步以内可以收敛，因此我迭代直至收敛。若取更大的K，如1000，则最好设定一个100左右的最大迭代步数，此时已经收敛得很好了。

确定每个簇的标签

在算法收敛后，可以得到每个簇C_k涵盖的样本点。这时需要给每个簇加上0~9的标签。我采取的方法是majority voting，即取簇中样本数最多的标签作为整个簇的标签。在此基础上，可以计算分类准确率。

可视化方法

我使用了`sklearn.manifold`中的tsne方法将样本特征嵌入到二维平面，并通过matplotlib将各类和中心点绘出。由于样本共60000个，我在每簇中平均采样，共采样1000个展示在图中。

效果与分析

不同特征向量和不同初始化方法

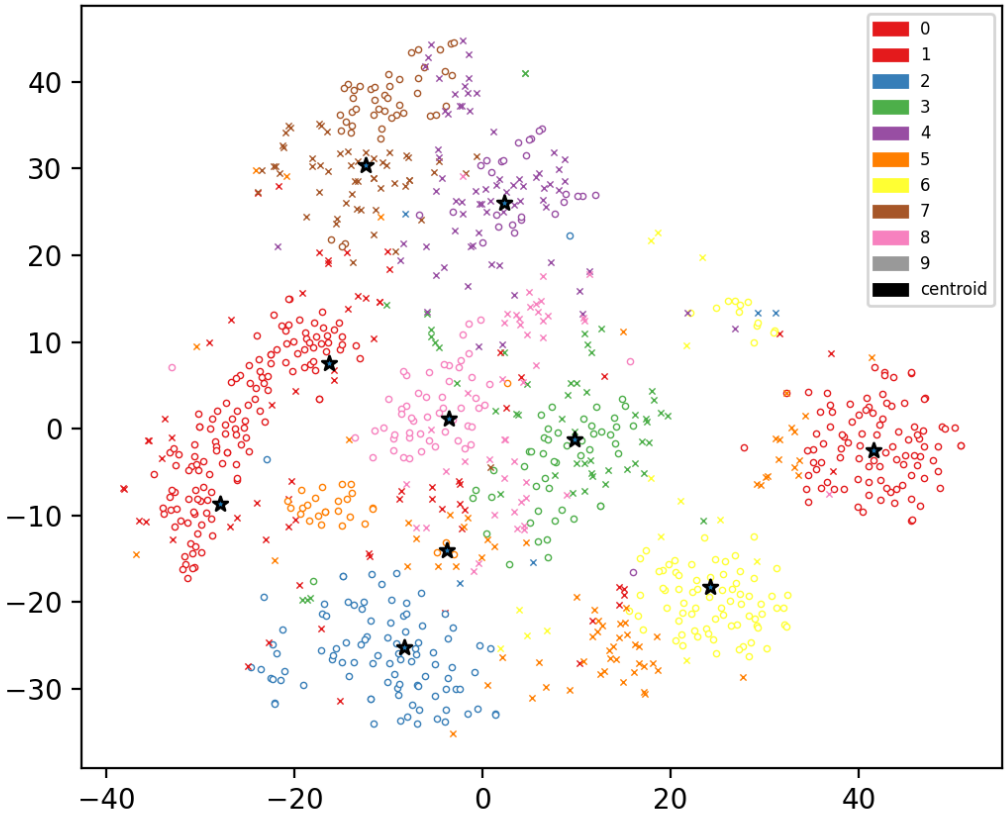
在$K = 10$时，采取不同特征向量和不同初始化方法，准确率结果如下表

特征/初始化	准确率	收敛步数
raw/random	59.07%	47
raw/representation	58.31%	126
hog/random	-	-
hog/representation	60.51%	79

其中hog/random出现了聚类失败的情况：某一初始中心点没有被分到任何样本。从上表可以看出：

- 使用hog特征相比于raw特征，在K较小时，可以提升收敛速度和准确率。由于hog特征向量长度为36，而raw特征向量长度为784，hog可以大大节省训练的时间、空间开销
- 使用random初始化有可能提高训练效果，但不如representation初始化稳定

同时注意到，在$K = 10$时效果不甚理想，只能达到60%左右的准确率。我们可以从可视化图分析。



上图是$K = 10$，采取raw特征、representation初始化时，算法收敛后的可视化示意图。图中有三种标志：

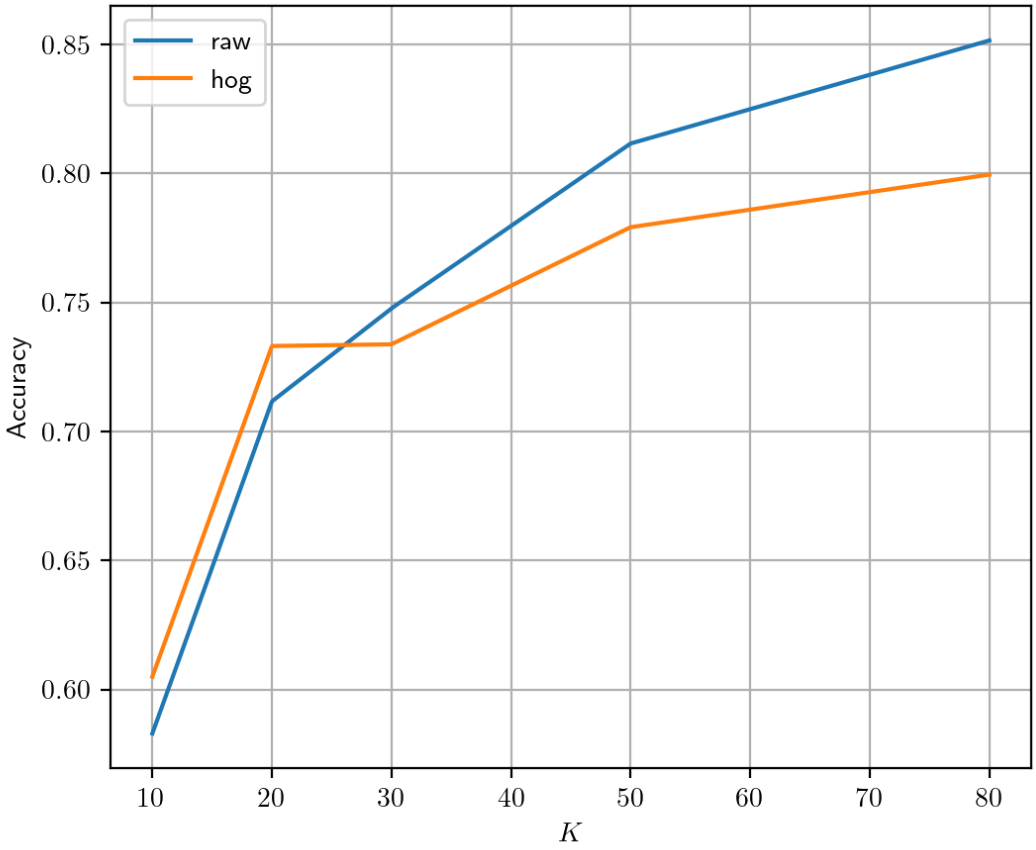
- o：分类正确
- x：分类错误
- *：簇中心点

颜色则代表每个样本被分到的类别。

从图中可以看到，有些类别分类的很好，而如棕色、紫色，中心点周围有很多x。用二维平面来类比，在这种特征、度量下，同一类的样本可能不分布在一个圆内，而是呈现狭长的长条状（如raw特征不存在shift invariance，不同位置的同一个数字在特征空间中可能距离很远）。图中左侧，两个中心点附近都是红色，是这种情况的例证。因此，10类数据很难被10个"圆"很好地分开。一种解决方法是增加K以增加聚类的精细度。

不同特征向量与不同K

我测试了在raw、hog两种特征下，$K = 10, 20, 30, 50, 80$时的分类准确率，结果如下图



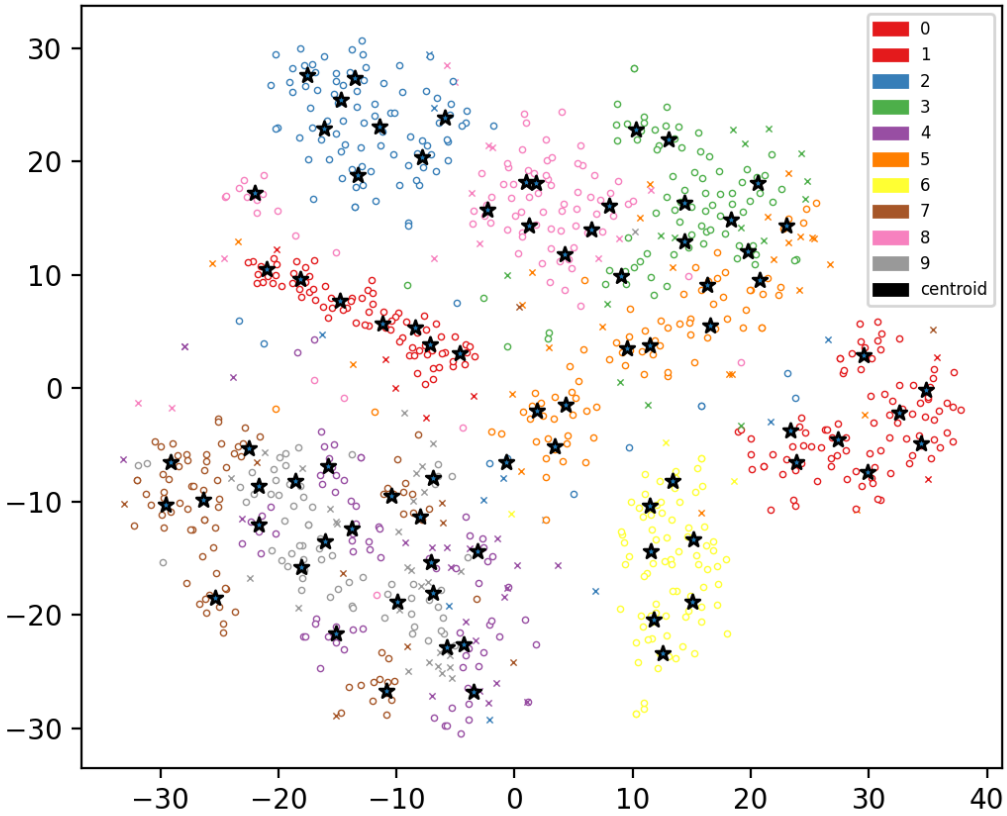
从中可以得出以下结论：

- K的增加可以大大增加分类准确率
- 在K较小时，hog特征具有速度和精度上的优势。而K较大时，hog特征反而不如raw特征，这可能是由于784->36的特征长度损失了一部分信息

总结与分析

在$K = 80$时，raw特征达到了85.16%的准确率。然而，随便训练一个神经网络，很容易达到98%以上的分类准确率。这可能是由于特征向量并不适合2范数的距离度量，我设想的一种方式是采用具有平移不变性的、高度凝练的特征，如将CNN去掉最后一层后的运算结果作为特征，这样有可能使用更短的特征向量达到更好的效果。

将$K = 80$时的可视化图拿出来，可以对其进行分析



图中距离较近的簇具有较高的相似度。如9,4,7都集中在左下角，3,5,8则位于右上。图中3/7，0/1，2/6位于图的两侧，距离很远，它们在形状上确实很不相近，这也与直觉是相符的。