上海大学

SHANGHAI UNIVERSITY

课程论文

COURSEPAPER

基于 K-means 的图像分割

学　　院　理学院

课 程 名　机器学习

专　　业　数学与应用数学

学　　号　20123175

姓　　名　俞梦泽

基于 K-means 的图像分割

摘要：图像分割是近年来比较火热的一个话题，其作为图像分析的基础步骤，在日常的生产、生活中都发挥着重要的作用。为了更好地处理这一问题，学者们提出了聚类方法，将图像的像素点进行聚类，再将“同类”的点赋上相同的特征，即可做到分割的效果。本文使用的 K-means 方法正是诸多聚类方法的其中之一。K-means 算法是一种随机挑选 K 个聚点中心，再将点按照与聚点中心的距离归类，并重新计算聚点中心，以此循环迭代的方法。收到论文《An Efficient k-Means Clustering Algorithm: Analysis and Implementation》的启发，本文在使用 K-means 聚类法的过程中，对其进行了一定程度上的改进。首先，本文引入了 kd 树算法帮助在每次迭代中寻找各个点最近的聚类中心，加快了运行时间。其次，本文选取了 Lloyd 方法，对算法迭代，以得到更准确的分类结果。

关键词：K-means 聚类、kd 树寻找近邻、Lloyd 迭代

一、引言

1.1 本文主要内容

本文主要参考了《An Efficient k-Means Clustering Algorithm: Analysis and Implementation》论文中对 K-means 聚类方法的应用，利用该算法进行图像分割。其中主要使用如下方法实现聚类：对于图像信息的提取与处理、kd 树构建与使用帮助查找近邻、基于 Lloyd 算法的迭代方式以及 K-means 方法构建聚类并归类。

1.2 本文章节安排

本文共分为四章，其中各个章节的主要内容如下。

第一章为引言，介绍了本文主要的研究内容和行文组织结构。

第二章将介绍程序对于图像信息的提取与处理。其中主要涉及如何将图像的 RGB 信息转化为可被聚类的数组信息

第三章主要介绍了本文所用的实验方法的主要内容。本文主要使用了 K-means 聚类方法将颜色相近的像素点归为同类，再转化输出。而在使用 K-means 聚类的过程中，又使用了 kd 树帮助计算最近聚点，和 Lloyd 方法帮助迭代。

第四章讲述了在本文完成对过程中以及本学期对《机器学习》这门课的一些学习和体会。

二、图像处理过程

本文主要对彩色图像进行处理。处理过程主要包括输入与输出两个部分。输入时主要使用 openCV 的 imread 方法读取。读取成彩色图像的 RGB 数组样式。同时，因为读取出来的数组是一个三维数组，以下图为例，各维度分别为：(800(宽)×1200(长)×3(RGB 参数))，为

了方便后续的聚类，reshape 数组将前两项展开，得到一个二维数组(96000(像素点个数)×3(RGB 参数))，数组中的每个向量表示按顺序排列的每个像素点的 RGB 参数。而在聚类完成后，需要将聚类回归的结果进行输出，由于聚类输出的是一个一维数组(96000×1)首先我们要将这个数组还原成图像大小的样式，即(800×1200)。最后将聚类得到的三个簇，对应三种颜色，利用 imshow 和 cmap 函数输出成分割后的图片。

实验样本图

三、实验方法概述

3.1 K-means 聚类基础介绍

聚类，简单来说，就是将一个庞杂数据集中具有相似特征的数据自动归类到一起，称为一个簇，簇内的对象越相似，聚类的效果越好。它是一种无监督的学习方法,不需要预先标注好的训练集。因此，聚类方法经常被用于图像分割。

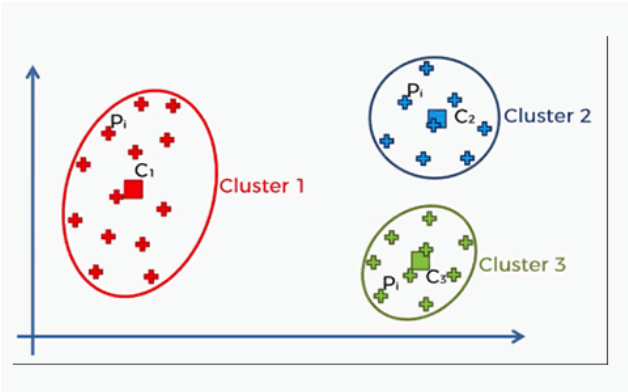
其中，K-Means 是发现给定数据集的 K 个簇的聚类算法, 之所以称之为 K-均值是因为它可以发现 K 个不同的簇, 且每个簇的中心采用簇中所含值的均值计算而成。簇个数 K 是用户指定的, 每一个簇通过其质心, 即簇中所有点的中心来描述。

K-means 的工作流程如下：

首先, 随机确定 K 个初始点作为质心。

然后将数据集中的每个点分配到一个簇中, 具体来讲, 就是为每个点找到距其最近的质心, 并将其分配该质心所对应的簇. 这一步完成之后, 每个簇的质心更新为该簇所有点的平均值.

重复上述过程直到数据集中的所有点都距离它所对应的质心最近时结束。



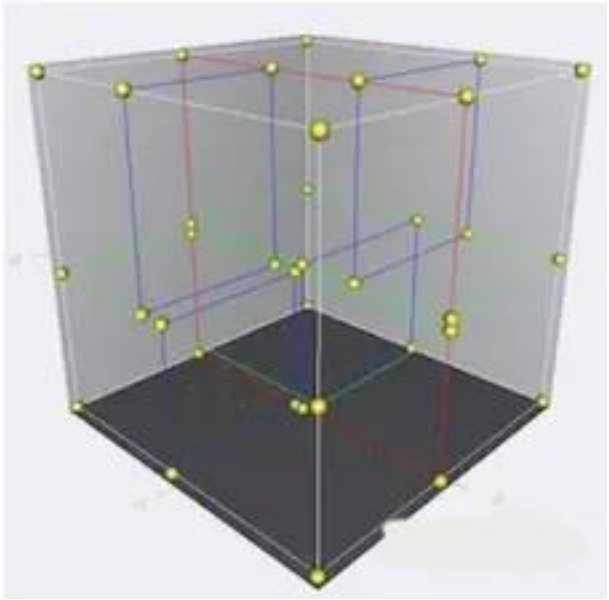
在这个过程中，有一些比较重要的概念，需要在此先进行定义：

在本文中，K，即簇的数量，经过实验，决定选择为 3。在本文中，选择最近聚点时，距离定义为正常的欧氏空间距离，由于是 RGB 表示，所以在三维空间中计算，寻找最近聚点时，采用 kd 树方法辅助。而在迭代时，使用 Lloyd 方法进行聚点选择与终止。下文将介绍 kd 树方法与 Lloyd 方法。

3.2 kd 树寻找近邻方法

最近邻搜索是指在一个确定的距离度量和一个搜索空间内寻找与给定查询项距离最小的元素。暴力搜索的解法时间复杂度是 O(n), 使用 KD-tree 能降低时间复杂度。由于维数灾难，我们很难在高维欧式空间中以较小的代价找到精确的最近邻。近似最近邻搜索则是一种通过牺牲精度来换取时间和空间的方式从大量样本中获取最近邻的方法。虽然，这种优化效果在本文中，并不明显，因为本文数据的维数只有 3。

更具体地，KD-tree 是 K-dimension tree 的缩写，是对数据点在 k 维空间中划分的一种数据结构。本质上，k-d 树是一种空间划分树，是一种平衡二叉树。将整个 k 维的向量空间不断的二分，从而划分为若干局部空间，然后搜索的时候，不断进行分支判断，选择其中的局部子空间，避免了全局空间搜索。举个例子，当 k=3 的时间，KD-tree 就会将三维空间分割成如下图的形式：



实际操作中，我们要先构建 kd 树：

1. 选择维度：KD 树建树采用的是从 m 个样本的 n 维特征中，分别计算 n 个特征的取值的方差，用方差最大的第 k 维特征来作为根节点。

2.选择中位数：对于这个特征，我们选择特征的取值的中位数 v 对应的样本作为划分点。

3. 分割数据：对于所有第 k 维特征的取值小于 v 的样本，我们划入左子树，对于第 k 维特征的取值大于等于 v 的样本，我们划入右子树。

4. 递归迭代：对于左子树和右子树，我们采用和刚才同样的办法来找方差最大的特征来做更节点，将空间和数据集进一步细分，如此直到所有点都被划分。

在本文中，k 为 3，而我们用来构建 kd 树的样本为 K-means 中的聚点(即每个簇的中心点)。因此，我们需要比较三个聚点的 R、G、B 三个维度特征中方差最小的那个维度。然

后，寻找该特征下处在中间的那个点，以其为中位数划分剩下两个点。再用剩下两个点各自划分。

接着，我们要利用构建好的 kd 树，对引入的目标点(在本文中为每个像素点的 RGB 数据)寻找最近的聚点：

1. 在 KD 树中找出包含目标点 t 的叶结点。即从根结点出发，递归地向下搜索二叉树。若在当前划分维度, t 的坐标值小于切分点，则移动到左子结点，否则移动到右子结点，直到走到叶结点为止。

2. 以此叶结点为“当前最近点”，递归的向上回溯，在每个结点进行以下操作：

i) 如果该结点保存的实例点比当前最近点距离目标点更近，则更新“当前最近点”，也就是说以该实例点为“当前最近点”。

ii)当前最近点一定存在于该结点一个子结点对应的区域，检查子结点的父结点的另一子结点对应的区域是否有更近的点。具体做法是，检查另一子结点对应的区域是否以「目标点为球心，以目标点与“当前最近点” 间的距离为半径的超球体」相交。

a) 如果相交，可能在另一个子结点对应的区域内存在距目标点更近的点，移动到另一个子结点，接着，继续递归地进行最近邻搜索；

b) 如果不相交，向上回溯。

当回退到根结点时，搜索结束，最后的“当前最近点”即为 x 的最近邻点。

这种处理法的好处是，相比枚举法，搜索过程中，一般只需要计算与最近的几个点的距离，较为快速。如下图所示：

但由于本文中，kd 树只使用到了三个聚点，所以优化效果并不明显。

综上，在本文中，我们只需要每次迭代时，根据当前聚点建立 kd 树，再将每个像素点作为目标点，寻找最近的聚点，将其归类即可。

3.3 Lloyd 方法

Lloyd 算法是 K-means 的三种迭代方式之一，其步骤如下：

1. 随机选取 K 个点作为初始的中心点
2. 计算每个点与 K 个中心点的 K 个距离(假如有 N 个点，就有 N*K 个距离值)
3. 分配剩下的点到距离其最近的中心点的类中，并重新计算每个类的中心点
4. 重复步骤 2 和步骤 3
5. 直到达到收敛或者达到某个停止阈值(如最大计算时间)，在本文中，当聚类结果不再

改变时，便停止。

用公式解释如下：

1. 把所有的点分配到 K 个类的系数 r，属于第 k 个类的记为 1，否则为 0

$$r_{ik} \in \{0,1\} \ \ (1 \le i \le N, 1 \le k \le K)$$

2. 由上可知第 k 个类中的样本数量

$$N_k = \sum_{i=1}^{N} r_{ik}$$

3. 目标是最小化损失函数 J

$$J = argmin \sum_{k=1}^{K} \sum_{i=1}^{N} r_{ik} \times |x_i - \mu_k|$$

4. 计算每个点到中心点的距离，分配 r 的系数

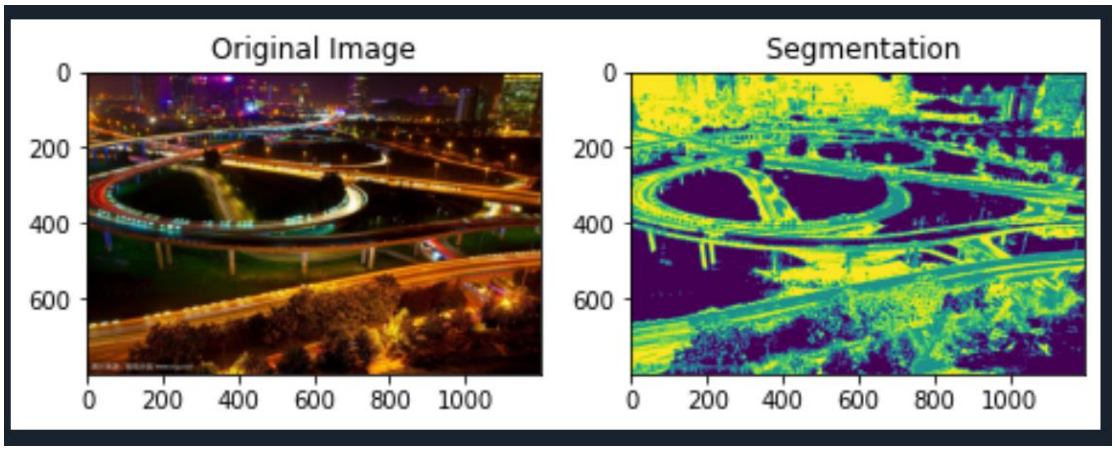
$$r_{ij} = \{k|argmin|x_i - \mu_k|\}$$

5. 重新计算每个类的中心点

$$\mu_k = \frac{\sum_{i=1}^{N} r_{ik} \times x_i}{\sum_{i=1}^{N} r_{ik}} = \frac{\sum_{i=1}^{N} r_{ik} \times x_i}{N_k}$$

3.4 实验结果

如上，进行操作，对图片的每个像素点进行聚类，寻找近邻，并迭代，最终得到的结果，如下：



缺陷：

由于本文选择的 K(簇数)较小，且像素点特征为 3 维，事实上并没有能很好地展现出 kd 树算法和 Lloyd 迭代法的优势，比较遗憾。

四、总结与体会

在本学期的《机器学习》课中，我系统性地学习到了多种建立模型与数据处理、预测的方法，也渐渐学会了使用一些常见的模型对数据进行处理和分析，然后采用设计好的模型进行训练，不管是前几周的课堂练习，还是期末的大作业，都得到了一个相对来说较好的效果。但是，在实际运用算法的过程中，我还是发现自己的基本功，不是很牢靠，尤其在代码实现

的过程中，对许多函数的功能，函数的参数设置，不太熟悉，导致程序往往达不到模型想要的效果。另外，在实际拿到问题时，自己有时候也不确定各种算法之间的优劣，无法做出良好的判断，希望后续可以改进。

参考文献：

[1] Tapas Kanungo, Senior Member, IEEE, David M. Mount, Member, IEEE,Nathan S. Netanyahu, Member, IEEE, Christine D. Piatko, Ruth Silverman, andAngela Y. Wu, Senior Member, IEEEMaosong Sun, Jingyang Li, Zhipeng Guo, Yu Zhao, Yabin Zheng, Xiance Si, Zhiyuan Liu. THUCTC: An Efficient k-Means Clustering Algorithm:Analysis and Implementation, An Efficient Chinese Text Classifier. 2016.

[2] 胡素黎, 黄丰喜, and 刘晓英,基于SVM-Kd-Tree的树型粗分类方法，软件导刊 19.4 (2020): 111114. Print.

[3] 高亮 谢健 曹天泽，基于 Kd 树改进的高效 K-Means 聚类算法，计算技术与自动化 34.4 (2015): 6974. Print.