# KNN实验说明

KNN是机器学习的经典入门算法，是最好的文本分类算法之一，在机器学习分类算法中占有相当大的地位，是最简单的机器学习算法之一。

KNN的基本思想有点类似“物以类聚，人以群分”，打个通俗的比方就是“如果你要了解一个人，可以从他最亲近的几个朋友去推测他是什么样的人”。概括来说，就是已知一个样本空间里的部分样本分成几个类，然后，给定一个待分类的数据，通过计算找出与待分类的数据最接近的K个样本，由这K个样本投票决定待分类数据归为哪一类。

KNN可以应用在各个方面，例如预测分析(股票)、模式识别（0-9数字识别）、文本分类、图像处理（图片分类）、电影推荐、保险业精准营销等。

## 实验一：手写数字识别

使用K-NN算法识别数字0-9，数据集来自Machine Learning in Action 第二章。书中把来自UCI数据库的手写数据集简化成32像素x32像素的黑白图像，并且以01矩阵的方式存储在txt文件中。大约有训练样本2000个，测试样本900个。

我们在课堂上已经做了演示：<https://shichengcn.github.io/KNN>

注意：需要使用谷歌浏览器或者webkit内核浏览器，初始化时会阻塞一段时间，用来下载数据、创建缓存。

**编程语言：**不限，推荐常用语言如python等

**实验数据集：**<https://shichengcn.github.io/KNN/digits/digits.zip>

**数据集介绍：**

digits 目录下有两个文件夹，分别是:

trainingDigits：训练数据，1934个文件，每个数字大约200个文件。

testDigits：测试数据，946个文件，每个数字大约100个文件。

每个文件中存储一个手写的数字，文件的命名类似0\_7.txt，第一个数字0表示文件中的手写数字是0，后面的7是个序号。

我们使用目录trainingDigits中的数据训练分类器，使用目录testDigits中的数据测试分类器的效果。两组数据没有重叠，你可以检查一下这些文件夹的文件是否符合要求。根据这些数据我们开始实现KNN算法。

**实验内容：**

1. 实现最基本的KNN算法，使用trainingDigits文件夹下的数据，对testDigits中的数据进行预测。（K赋值为1，使用欧氏距离，多数投票决定分类结果）
2. 改变K的值，并观察对正确率的影响。
3. 更改距离度量方式，更改投票方式（距离加权），分析错误率。

**实验要求：**

1. 要求给出代码，以及运行窗口截图。
2. K对正确率的影响，最好用表格或作图说明，并做简要分析。
3. 实验内容3为选做，不做统一要求。

## 实验二 KNN的其他相关知识

1. 搜索KNN的常见应用场景，例如预测分析(股票)、文本分类、图像处理（图片分类）、保险业精准营销等。
2. 了解KNN最近邻查找和范围查找使用的数据结构KD树。
3. 了解KNN算法中找到离自己最近的K个点的”最大堆(Max Heap)”算法，这些内容都是很经典的算法面试题。

实验要求：

选择1-2项进行了解，给出你查询的资料的网页链接或者是参考文献引用列表，每一个都要给出主体思想或者内容的简短介绍。（最好每个不少于50字）

## 实验三 KNN其他方面的应用（例如：推荐系统）

**推荐系统入门资料：**

探索推荐引擎内部的秘密<https://www.ibm.com/developerworks/cn/web/1103_zhaoct_recommstudy1/index.html>

基于协同过滤的推荐方法：**<https://www.tuicool.com/articles/UZ7FFnm>**

ItemCF：ItemCollaborationFilter，基于物品的协同过滤：<http://blog.csdn.net/yeruby/article/details/44154009>

UserCF：UserCollaborativeFilter，基于用户的协同过滤：<https://www.tuicool.com/articles/AN7Rf2>

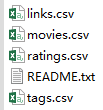
**编程语言：不限**

**实验数据集：**<http://files.grouplens.org/datasets/movielens/ml-latest-small.zip>

**数据集介绍：**<http://files.grouplens.org/datasets/movielens/ml-latest-small-README.html>

Movielens由美国 Minnesota（明尼苏达） 大学计算机科学与工程学院的 GroupLens 项目组创办，是一个非商业性质的、以研究为目的的实验性站点。MovieLens 主要使用 Collaborative Filtering（协同过滤） 和 Association Rules （关联规则）相结合的技术，向用户推荐他们感兴趣的电影。

MovieLens采集了一组从20世纪90年末到21世纪初由用户提供的电影评分数据。数据中包括电影评分、电影元数据（风格类型和年代），有不同的打包格式和文件大小，这里我们使用“ml-latest-small”这一版本，它是csv格式文件，总体大小为1M。包含大约700个用户对9000部电影1-5分的100,000个打分，另外有1300种电影标签 。每个用户至少对20部电影进行过评价。



其中links.csv中给出了每个movie-ID对应的imdb-ID，IMDB是美国最大的电影评论网站，类似于豆瓣。相当于可以知道每一个电影ID对应的具体信息。movies.csv给出了电影的名称和流派信息。tags.csv给出了用户对电影的打标签情况。ratings.csv给出了用户对电影的打分（1-5分）。Timestamp一项是时间戳，一种时间表示方式，定义为从格林威治时间1970年01月01日00时00分00秒起至现在的总秒数，代表了打分动作发生的时间。

这里为了简化问题，给出的文件中我们只使用ratings.csv，也就是只考虑打分数据，而且也不使用时间项。

**实验内容：**

1. 了解推荐系统的基础知识，最基本的基于近邻的方法。（ItemCF和UserCF）
2. 划分实验数据集为训练集和验证集。
3. 使用ItemCF或UserCF算法对每个用户进行推荐，统计推荐结果。（一些参数自行决定，包括近邻个数，具体推荐的电影个数，推荐的评价指标一般选RMSE）

**实验要求：**

1. 了解KNN在推荐系统中的应用，给出Collaborative Filtering（协同过滤）的思想介绍，简单叙述ItemCF和UserCF的思想。
2. 整个第二问的代码部分为选做，不做统一要求。建议有兴趣、有时间的同学上手尝试，网络上也有非常多的例子。

实验三编程参考资料

**Python读取csv文件：**

<http://www.360doc.com/content/14/0707/10/16740871_392565773.shtml>

**划分实验数据集为训练集和验证集：**

<https://github.com/shichengcn/recommender-system/blob/master/Create_training_data_and_test_data.py>

**UserCf和ItemCF协同过滤推荐算法的实现：**

<https://github.com/xingzhexiaozhu/MovieRecommendation>