k 近邻算法

我们在网上购买水果的时候经常会看到同一种水果会标有几种规格对应不同价格进行售卖，水果分级售卖已经是电商中常见的做法，那么水果分级具体是怎么操作的呢？一种简单的做法是根据水果果径的大小进行划分。今年老李家苹果丰收了，为了能卖个好价钱，老王打算按照果径对苹果进行分级。想法是很好的，但是面对成千上万的苹果这可愁坏了老李。老李的儿子小李是计算机系毕业的，他知道这件事后设计了一个算法，按照老李的要求根据果径大小定义了 5 个等级

70mm 左右（<72.5mm）
75mm 左右(>=72.5mm&&<77.5mm)
80mm 左右(>=77.5mm&&<82.5mm)
85mm 左右(>=82.5mm&&<87.5mm)
90mm 左右(>=87.5mm)
如下图：

当一个未分级的苹果拿到后可以首先将这个苹果的果径测量出来，然后再和这 5 个等级的苹果进行对照，假如未分级苹果的果径是 82mm 则划分为第三个等级，如果是 83mm 则划分为第二个等级，以此类推。基于这个原则小李发明了一个分级装置，见下图，大大提高了工作效率，很快将老李的问题解决了。

老李的问题是一个经典的最近邻模板匹配，根据一个已知类别参考模板对未分类的数据进行划分，小李选择的每个类的模板数是一，现实生活中的问题往往会复杂很多，可能需要多个参考模板进行综合决策，当选定的模板数为 k 的时候就是 k 近邻算法的思想了，最近邻算法是 k 近邻算法 k=1 时的一种特殊情况。

k 近邻算法简称 kNN 算法，由 Thomas 等人在 1967 年提出[1]。它基于以下思想：要确定一个样本的类别，可以计算它与所有训练样本的距离，然后找出和该样本最接近的 k 个样本，统计这些样本的类别进行投票，票数最多的那个类就是分类结果。因为直接比较样本和训练样本的距离，kNN 算法也被称为基于实例的算法。

基本概念

确定一个样本所属类别的一种最简单的方法是直接比较它和所有训练样本的相似度，然后将其归类的最相似的样本所属的那个类，这是一种模板匹配的思想。下图 6.1 是使用 k 近邻思想进行分类的一个例子：

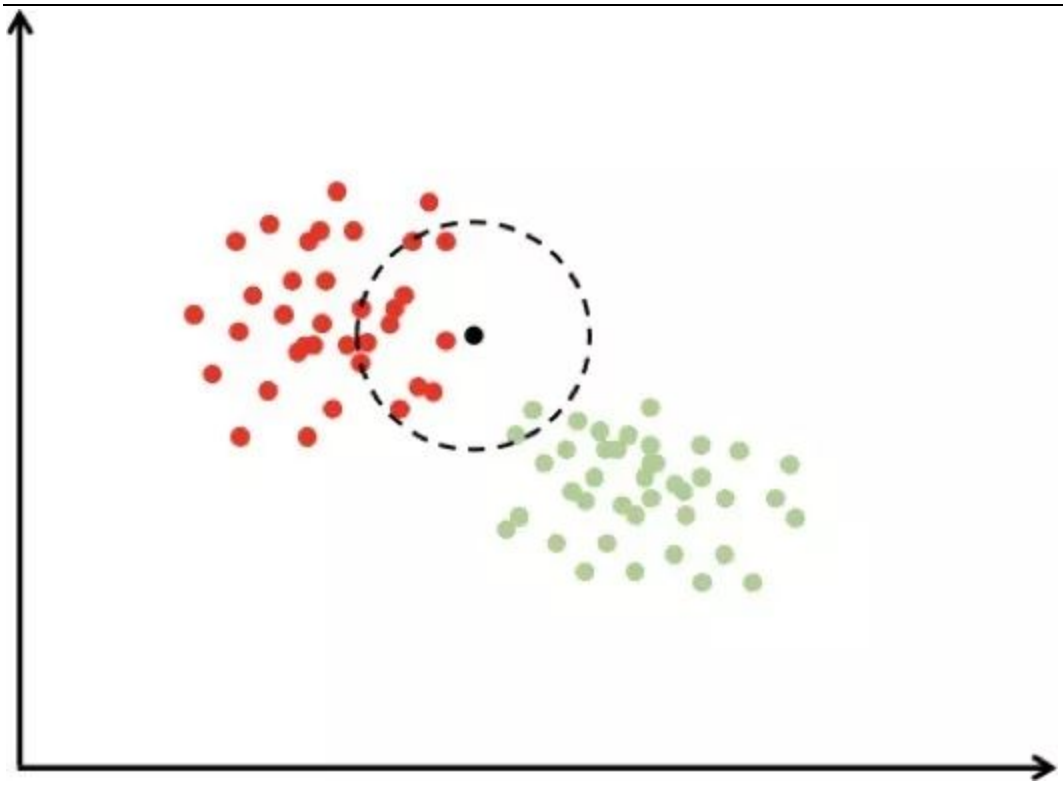

图 6.1 k 近邻分类示意图

在上图中有红色和绿色两类样本。对于待分类样本即图中的黑色点，我们寻找离该样本最近的一部分训练样本，在图中是以这个矩形样本为圆心的某一圆范围内的所有样本。然后统计这些样本所属的类别，在这里红色点有 12 个，圆形有 2 个，因此把这个样本判定为红色这一类。上面的例子是二分类的情况，我们可以推广到多类，k 近邻算法天然支持多类分类问题。

预测算法

k 近邻算法没有求解模型参数的训练过程，参数 k 由人工指定，它在预测时才会计算待预测样本与训练样本的距离。

对于分类问题，给定 l 个训练样本 (x_i, y_i)，其中 x_i 为维特征向量，y_i 为标签值，设定参数 k，假设类型数为 c，待分类样本的特征向量为 x。预测算法的流程为：1.在训练样本集中找出离 x 最近的 k 个样本，假设这些样本的集合 N。

2.统计集合 N 中每一类样本的个数 $C_i, i=1,...,c$。

3.最终的分类结果为 $\arg\max_i C_i$。

在这里 $\arg\max_i C_i$ 表示最大的 C_i 值对应的那个类 i。如果 $k=1$，k 近邻算法退化成最近邻算法。

k 近邻算法实现简单，缺点是当训练样本数大、特征向量维数很高时计算复杂度高。因为每次预测时要计算待预测样本和每一个训练样本的距离，而且要对距离进行排序找到最近的 k 个样本。我们可以使用高效的部分排序算法，只找出最小的 k 个数；另外一种加速手段是 k-d 树实现快速的近邻样本查找。

一个需要解决的问题是参数 k 的取值。这需要根据问题和数据的特点来确定。在实现时可以

考虑样本的权重，即每个样本有不同的投票权重，这称方法称为为带权重的 k 近邻算法。

另外还其他改进措施，如模糊 k 近邻算法[2]。

kNN 算法也可以用于回归问题。假设离测试样本最近的 k 个训练样本的标签值为则对样本的回归预测输出值为：

$$y = \left(\sum_{i=1}^{k} y_i \right) / k$$

即所有邻居的标签均值，在这里最近的 k 个邻居的贡献被认为是相等的。同样的也可以采用带权重的方案。带样本权重的回归预测函数为：

$$y = \left(\sum_{i=1}^{k} w_i y_i \right) / k$$

其中 w_i 为第 i 个样本的权重。权重值可以人工设定，或者用其他方法来确定，例如设置为与距离成反比。

距离定义

根据前面的介绍，kNN 算法的实现依赖于样本之间的距离值，因此需要定义距离的计算方式。接下来介绍常用的几种距离定义，它们适用于不同特点的数据。

假设两个向量之间的距离为这是一个将两个维数相同的向量映射为一个实数的函数。距离函数必须满足以下条件，第一个条件是三角不等式：

$$d\left(\mathrm{x}_i,\mathrm{x}_k\right)+d\left(\mathrm{x}_k,\mathrm{x}_j\right)\geq d\left(\mathrm{x}_i,\mathrm{x}_j\right)$$

这和我们学习的几何中的三角不等式吻合。第二个条件是非负性，即距离不能是一个负数：

$$d\left(\mathrm{x}_i,\mathrm{x}_j\right)\geq 0$$

第三个条件是对称性，即 A 到 B 的距离和 B 到 A 的距离必须相等：

$$d\left(\mathrm{x}_i,\mathrm{x}_j\right)=d\left(\mathrm{x}_j,\mathrm{x}_i\right)$$

第 5 个条件是区分性，如果两点间的距离为 0，则两个点必须相同：

$$d\left(\mathrm{x}_i,\mathrm{x}_j\right)=0\Rightarrow \mathrm{x}_j=\mathrm{x}_i$$

满足上面 4 个条件的函数都可以用作距离定义。

常用距离定义

常用的有欧氏距离，Mahalanobis 距离等。欧氏距离是最常见的距离定义，它就是维欧氏空间中两点之间的距离。对于 R^n 空间中有两个点 x 和 y，它们之间的距离定义为：

$$d\left(\mathrm{x},\mathrm{y}\right)=\sqrt{\sum_{i=1}^{n}\left(x_i-y_i\right)^2}$$

这是我们最熟知的距离定义。在使用欧氏距离时应该尽量将特征向量的每个分量归一化，以减少因为特征值的尺度范围不同所带来的干扰。否则数值小的特征分量会被数值大的特征分量淹没。例如，特征向量包含两个分量，分别为身高和肺活量，身高的范围是 150-200 厘米，肺活量为 2000-9000，如果不进行归一化，身高的差异对距离的贡献显然为被肺活量淹没。欧氏距离只是将特征向量看做空间中的点，并没有考虑这些样本特征向量的概率分布规律。

Mahalanobis 距离是一种概率意义上的距离，给定两个向量 x 和 y 以及矩阵 S，它定义为：

$$d\left(\mathrm{x},\mathrm{y}\right)=\sqrt{\left(\mathrm{x}-\mathrm{y}\right)^{\mathrm{T}}\mathrm{S}\left(\mathrm{x}-\mathrm{y}\right)}$$

要保证根号内的值非负，即矩阵 S 必须是半正定的。这种距离度量的是两个随机向量的相似度。当矩阵 S 为阶单位矩阵 I 时，Mahalanobis 距离退化为欧氏距离。矩阵可以通过计算训练样本集的协方差矩阵得到，也可以通过训练样本学习得到，优化某一目标函数。

对于矩阵如何确定的问题有不少的研究，代表性的有文献[9-12]，其中文献[9]提出的方法具有很强的指导意义和应用价值。文献[9]指出，kNN 算法的精度在很大程度上依赖于所使用的距离度量标准，为此他们提出了一种从带标签的样本集中学习得到距离度量矩阵的方法，称为距离度量学习（Distance Metric Learning），我们将在下一节中介绍。

Bhattacharyya 距离定义了两个离散型或连续型概率分布的相似性。对于离散型随机变量的分布，它的定义为：

$$d(\mathrm{x},\mathrm{y}) = -\ln\left(\sum_{i=1}^{n}\sqrt{x_i \cdot y_i}\right)$$

其中 x_i , y_i 为两个随机变量取某一值的概率，它们是向量 x 和 y 的分量，它们的值必须非负。两个向量越相似，这个距离值越小。

距离度量学习

Mahalanobis 距离中的矩阵 S 可以通过对样本的学习得到，这称为距离度量学习。距离度量学习通过样本集学习到一种线性变换，目前有多种实现。下面我们介绍文献[9]的方法，它使得变换后每个样本的 k 个最近邻居都和它是同一个类，而不同类型的样本通过一个大的间隔被分开，这和第 8 章将要介绍的线性判别分析的思想类似。如果原始的样本点为，变换之后的点为 y，在这里要寻找的是如下线性变换：

$$\mathrm{y} = \mathrm{L}\mathrm{x}$$

其中 L 为线性变换矩阵。首先定义目标邻居的概念。一个样本的目标邻居是和该样本同类型的样本。我们希望通过学习得到的线性变换让样本最接近的邻居就是它的目标邻居：

$$j \rightsquigarrow i$$

表示训练样本 x_j 是样本 x_i 的目标邻居。这个概念不是对称，x j 是 x_i 的目标邻居

不等于 x 是 x_i 的目标邻居。

为了保证 kNN 算法能准确的分类，任意一个样本的目标邻居样本要比其他类别的样本更接近于该样本。对每个样本，我们可以将目标邻居想象成为这个样本建立起了一个边界，使得和本样本标签值不同的样本无法入侵进来。训练样本集中，侵入这个边界并且和该样本不同标签值的样本称为冒充者（impostors），这里的目标是最小化冒充者的数量。

为了增强 kNN 分类的泛化性能，要让冒充者离由目标邻居估计出的边界的距离尽可能的远。通过在 kNN 决策边界周围加上一个大的安全间隔（margin），可以有效的提高算法的鲁棒性。

接下来定义冒充者的概念。对于训练样本 xi，其标签值为 yi，目标邻居为 xj，冒充者是指那些和 xi 有不同的标签值并且满足如下不等式的样本 xl：

$$\left\|\mathrm{L}\left(\mathrm{x}_i-\mathrm{x}_l\right)\right\|^2 \leq\left\|\mathrm{L}\left(\mathrm{x}_i-\mathrm{x}_j\right)\right\|^2+1$$

其中 L 为线性变换矩阵，左乘这个矩阵相当于对向量进行线性变换。根据上面的定义，冒充者就是闯入了一个样本的分类间隔区域并且和该样本标签值不同的样本。

训练时优化的损失函数由推损失函数和拉损失函数两部分构成。拉损失函数的作用是让和样本标签相同的样本尽可能与它接近：

$$\varepsilon_{pull}(\mathrm{L})=\sum_{j \rightsquigarrow i}\left\|\mathrm{L}\left(\mathrm{x}_i-\mathrm{x}_j\right)\right\|^2$$

推损失函数的作用是把不同类型的样本推开：

$$\varepsilon_{push}(\mathrm{L})=\sum_{i, j \rightsquigarrow i} \sum_l\left(1-y_{il}\right)\left[1+\left\|\mathrm{L}\left(\mathrm{x}_i-\mathrm{x}_j\right)\right\|^2-\left\|\mathrm{L}\left(\mathrm{x}_i-\mathrm{x}_l\right)\right\|^2\right]_+$$

如果 yi =yj，则 yij=1 ，否则 yij=0。函数[z]+定义为：

$$[z]_+=\max(z, 0)$$

如果两个样本类型相同，则有：

$$1-y_{il}=0$$

因此推损失函数只对不同类型的样本起作用。总损失函数由这两部分的加权和构成：

$$\varepsilon(\mathrm{L})=(1-\mu)\varepsilon_{pull}(\mathrm{L})+\mu\varepsilon_{push}(\mathrm{L})$$

在这里 μ 是人工设定的参数。求解该最小化问题，就得到了线性变换矩阵。通过这个线性变换，同类样本尽量都成为最近的邻居节点；而不同类型的样本会拉开距离。这会有效的提高 kNN 算法的分类精度。

实验程序

下面用一个例子程序来演示 kNN 算法的使用，这里我们对 2 个类进行分类。

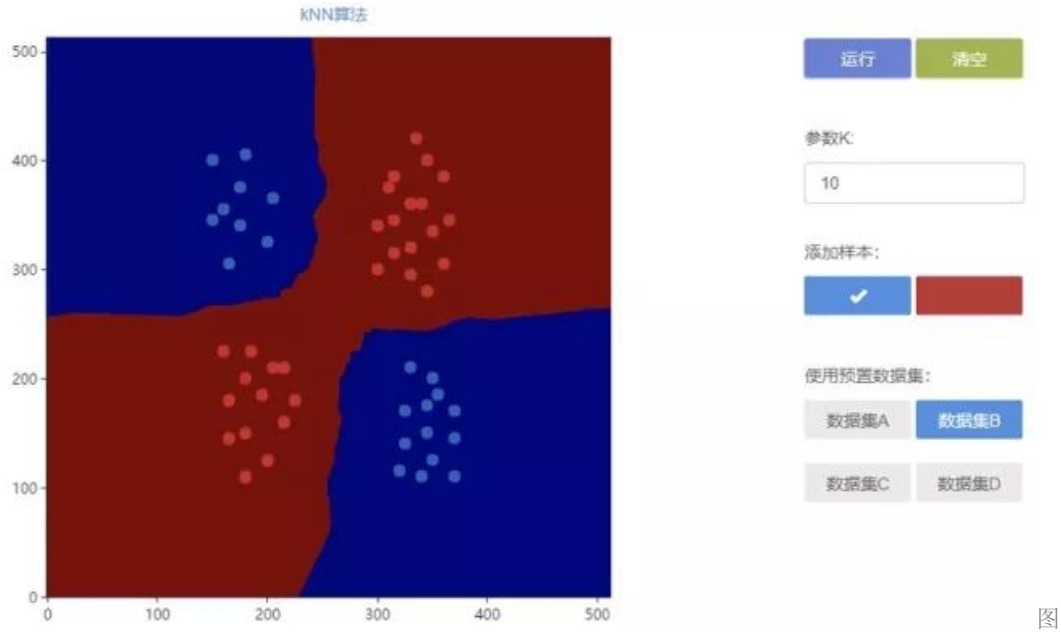

图 6.2 kNN 算法的分类效果

在这里分类边界是曲线，证明了 kNN 算法有非线性分类的能力。以上结果来自 SIGAI 云端实验室，如果你对此感兴趣，可以向 SIGAI 公众号发消息，申请使用。我们的实验室提供了强大的功能，可以帮助大家更容易，深刻的理解各种数学，机器学习，深度学习，以及应用领域的算法。

应用

kNN 算法简单但却有效，如果能够定义合适的距离度量，它可以取得很好的性能。kNN 算法被成功的用于文本分类[5-7]，图像分类[8-11]等模式识别问题。应用 kNN 算法的关键是构造出合适的特征向量以及确定合适的距离函数。

参 考 文 献

[1] Thomas M Cover, Peter E Hart. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 1967.
[2] James M Keller, Michael R Gray, James Givens. A fuzzy K-nearest neighbor algorithm. systems man and cybernetics, 1985.
[3] Thierry Denoeux. A k-nearest neighbor classification rule based on Dempster-Shafer theory. systems man and cybernetics, 1995
[4] Trevor Hastie, Rolbert Tibshirani. Discriminant adaptive nearest neighbor classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1996.
[5] Bruno Trstenjak, Sasa Mikac, Dzenana Donko. KNN with TF-IDF based Framework for Text Categorization. Procedia Engineering, 2014.
[6] J He, Ahhwee Tan, Chew Lim Tan. A Comparative Study on Chinese Text Categorization Methods. pacific rim international conference on artificial intelligence, 2000.
[7] Shengyi Jiang, Guansong Pang, Meiling Wu, Limin Kuang. An improved K-nearest-neighbor algorithm for text categorization. 2012, Expert Systems With Application.
[8] Oren Boiman, Eli Shechtman, Michal Irani. In defense of Nearest-Neighbor based image classification. 2008, computer vision and pattern recognition.
[9] Kilian Q Weinberger, Lawrence K Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. 2009, Journal of Machine Learning Research.
[10] S. Belongie, J. Malik, J. Puzicha. Shape matching and obejct recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(4):509-522, 2002.
[11] P. Y. Simard, Y. LeCun, I. Decker. Efficient pattern recognition using a new transformation distance. In S. Hanson, J. Cowan, and L. Giles, editors, Advances in Neural Information Processing Systems 6, pages 50-58, San Mateo, CA, 1993. Morgan Kaufman.
[12] S. Chopra, R. Hadsell, Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005), pages 349-356, San Diego, CA, 2005.