Author：Liu Jian

Time：2020-02-28

机器学习7- k 近邻算法

1 k 近邻算法的原理

k 近邻 (k -Nearest Neighbor, KNN) 算法的原理：给定训练集，基于某种距离度量找出训练集中与待预测点最近的 k 个训练样本，然后根据这 k 个样本的输出对待预测点进行预测，比如，对于**分类任务**可采用投票法或基于距离远近的加权投票法，对于**回归任务**可采用平均法或基于距离远近的加权平均法。

可以看到， k 近邻算法作为一种监督学习算法，没有显示的训练过程，属于懒惰学习 (lazy learning) 的范畴。

> 懒惰学习在训练阶段仅仅是把样本保存起来，训练时间开销为零，待收到预测点后再进行处理；相应的，那些在训练阶段就对样本进行学习处理的方法被称为急切学习 (eager learning)。

接下来，我们只讨论分类问题中的 k 近邻算法。 k 近邻算法实际上是利用训练数据集对特征向量空间进行划分，并作为其分类的模型。 k 近邻算法的三要素：

1. k 值的设定： k 值越小，意味着整体模型变得复杂，容易发生过拟合； k 值越大意味着整体模型变得简单，容易发生欠拟合。实际应用中， k 值一般取一个比较小的数值，常采用交叉验证法来确定；
2. 距离度量：可参见《机器学习5-EM算法与聚类》；
3. 分类决策规则：多数表决法等价于损失函数为 $0-1$ 损失的经验风险最小化。

当 $k=1$ 时就为最近邻算法 (1NN)，当采样密度足够大即密集采样时，其泛化错误率不超过贝叶斯最优分类器错误率的两倍。

2 k 近邻算法的实现 - kd 树

为了提高对 k 个近邻样本点的搜索效率，可采用 kd 树存储训练数据。我们知道二叉树是对一维数据的不断划分，满足： $左树 < 叶节点 < 右树$ 。现在假设数据点有 k 维，即 $x_i=(x_i^{(1)},x_i^{(k)},\cdots,x_i^{(n)}),\ i=1,2,\cdots,N$ 。那么 kd 树就是对 k 维数据的不断划分，类似于二叉树，每次划分我们选择数据在某个维度上的值进行比较，若此时轮到比较第 j 维的值，那么对数据的划分会使得： $左树第j维的值 < 叶节点第j维的值 < 右树第j维的值$ ，我们轮流循环选择比较的维度直至处理完所有数据点，这样就生成了 kd 树。

- 用 kd 树实现最邻近算法，《统计学习方法》上写的并不清楚，参见博客 kd 数算法中第二个例子可能会更清楚一点，里面很关键的是存在一个用栈存储的搜索路径，当栈空时程序停止而不是如《统计学习方法》所说回溯到根节点时程序停止。
- 用 kd 树实现 k 邻近算法有待进一步学习。个人想法是在最邻近算法的基础上动态地维护 k 个最短距离直至算法结束即可。

kd 树的 k 是指数据的维度，而 k 近邻算法的 k 是我们人为给定的。