KNN算法

1. **简介**

K最近邻(k-Nearest Neighbor，KNN)分类算法，是一个理论上比较成熟的方法，也是最简单的机器学习算法之一。该方法的思路是：如果一个样本在特征空间中的k个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别。KNN算法中，所选择的邻居都是已经正确分类的对象。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。 KNN方法虽然从原理上也依赖于极限定理，但在类别决策时，只与极少量的相邻样本有关。由于KNN方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因此对于类域的交叉或重叠较多的待分样本集来说，KNN方法较其他方法更为适合。KNN算法不仅可以用于分类，还可以用于回归。通过找出一个样本的k个最近邻居，将这些邻居的属性的平均值赋给该样本，就可以得到该样本的属性。更有用的方法是将不同距离的邻居对该样本产生的影响给予不同的[权值](http://baike.baidu.com/view/2109980.htm" \t "http://baike.baidu.com/_blank)(weight)，如权值与距离成反比。

1. **算法指导思想**

kNN算法的指导思想是“近朱者赤，近墨者黑”，由你的邻居来推断出你的类别。先算待分类样本与已知类别的训练样本之间的距离，找到距离与待分类样本数据最近的k个邻居；再根据这些邻居所属的类别来判断待分类样本数据的类别。

计算步骤如下：  
 1、算距离：给定测试对象，计算它与训练集中的每个对象的距离  
 2、找邻居：圈定距离最近的k个训练对象，作为测试对象的近邻  
 3、做分类：根据这k个近邻归属的主要类别，来对测试对象分类

对于k值的选择，没有一个固定的经验，一般根据样本的分布，选择一个较小的值，可以通过交叉验证选择一个合适的k值。选择较小的k值，就相当于用较小的领域中的训练实例进行预测，训练误差会减小，只有与输入实例较近或相似的训练实例才会对预测结果起作用，与此同时带来的问题是泛化误差会增大，换句话说，K值的减小就意味着整体模型变得复杂，容易发生过拟合；选择较大的k值，就相当于用较大领域中的训练实例进行预测，其优点是可以减少泛化误差，但缺点是训练误差会增大。这时候，与输入实例较远（不相似的）训练实例也会对预测器作用，使预测发生错误，且K值的增大就意味着整体的模型变得简单。一个极端是k等于样本数m，则完全没有分类，此时无论输入实例是什么，都只是简单的预测它属于在训练实例中最多的类，模型过于简单。

对于距离的度量，我们有很多的距离度量方式，但是最常用的是欧式距离，即对于两个n维向量x和y，两者的欧式距离定义为：QQ截图20170223142030

　　大多数情况下，欧式距离可以满足我们的需求，我们不需要再去操心距离的度量。当然我们也可以用他的距离度量方式。比如曼哈顿距离，定义为：1

1. **优缺点**

1、优点  
简单，易于理解，易于实现，无需估计参数，无需训练；  
适合对稀有事件进行分类（例如当流失率很低时，比如低于0.5%，构造流失预测模型）；  
特别适合于多分类问题(multi-modal,对象具有多个类别标签)，例如根据基因特征来判断其功能分类，kNN比SVM的表现要好。  
2、缺点  
懒惰算法，对测试样本分类时的计算量大，内存开销大，评分慢；

当样本不平衡时，如一个类的样本容量很大而其他类样本容量很小时，有可能导致当输入一个新样本时，该样本的K各邻居中大容量类的样本占多数；  
可解释性较差，无法给出决策树那样的规则。

1. **算法流程**

1. 准备数据，对数据进行[预处理](http://baike.baidu.com/view/1168039.htm" \t "http://baike.baidu.com/item/_blank)

2. 选用合适的数据结构存储训练数据和测试元组

3. 设定参数，如k

4.维护一个大小为k的的按距离由大到小的[优先级队列](http://baike.baidu.com/view/2539614.htm" \t "http://baike.baidu.com/item/_blank)，用于存储最近邻训练元组。随机从训练元组中选取k个元组作为初始的最近邻元组，分别计算测试元组到这k个元组的距离，将训练元组标号和距离存入优先级队列

5. 遍历训练元组集，计算当前训练元组与测试元组的距离，将所得距离L 与优先级队列中的最大距离Lmax

6. 进行比较。若L>=Lmax，则舍弃该元组，遍历下一个元组。若L < Lmax，删除优先级队列中最大距离的元组，将当前训练元组存入优先级队列。

7. 遍历完毕，计算优先级队列中k 个元组的多数类，并将其作为测试元组的类别。

8. 测试元组集测试完毕后计算误差率，继续设定不同的k值重新进行训练，最后取误差率最小的k值。

个人感想：

这次做MATLAB课程报告的过程中我收获了很多，过程很艰辛，甚至做的想要放弃，但是我们坚持下来了，我们小组成员通力合作，终于让努力结出了果实。

首先说一说艰辛之处。我们毕竟只学习了MATLAB的一点皮毛，用它来实现算法解决实际问题对我们而言还是力不能及的事情。当拿到Topic1的题目时，感觉很吃力，里面要用到的算法，诸如K-Means、KNN算法、Baggging算法对我来说都太陌生了。和小组成员讨论过后，我们决定先学习算法，了解算法的基本原理和作用，然后再设计算法来解决实际问题。为了学习这些算法，我从图书馆借来了数据挖掘和机器学习方面的书籍，认真研究了这些算法，对算法原理基本原理有所了解了。接下来是算法设计过程，如何用算法去解决问题。我们的问题就在于了解算法，却不知道如何去用算法。我们思考了很久，也看了很多算法实例，还是一无所获，最后去请教了学长才解决了问题。最后一步就是算法实现了，我们在编写代码的过程中也有很多障碍，我深感Matlab基础知识不够扎实，代码经常报错。我们找了很多代码示例，才勉强成功了。

回想起做的过程，真的很揪心，但是我们痛并快乐着。说一说收获吧，在做报告的过程中我收获很大。首先是我学到了几个有用的算法，其次是我看到了合作的力量，我们小组每个人都很努力，很团结，大家都积极贡献自己的智慧。最大的收获是让我认识到了Matlab的强大功能，这是一款很实用的科学计算语言，可以解决很多方面的问题。这次课程报告让我学会了用Matlab去解决实际问题，也巩固了Matlab基础知识。