基于KNN算法模型的研究

熊舒鹏

（昆明理工大学 机器人工程 昆明市）

1. **KNN算法（K Nearest Neighbors）**

## KNN算法定义

**(Gemin 2.5)**

KNN 算法本身并没有复杂的数学“推导”过程，它更多是基于距离度量和邻居投票/平均的规则。其数学表示主要体现在以下几个方面：

1. 距离度量 (Distance Metric)
2. 寻找 K 个最近邻 (Finding K Nearest Neighbors)
3. 预测 (Prediction)
   1. 分类 (Classification)
   2. 回归 (Regression)

下面我们逐一进行数学公式的阐述。

假设我们有一个训练数据，其中是一个具有*m*个特征的样本向量，*yi*是对应的标签（分类问题）或数值（回归问题）。现在给定一个待预测的新样本。

### **距离度量 (Distance Metric)**

计算待预测样本xtest与训练集中每个样本xi之间的距离是KNN的第一步。选择合适的距离度量对于KNN的性能至关重要。常用的距离度量包括：

1. **欧几里得距离 (Euclidean Distance)**

这是最常用的距离度量，衡量两点在多维空间中的直线距离。对于两个 m 维向量 *v*1=(*v*1,1,*v*1,2,…,*v*1,m) 和 *v2*=(*v*2,1,*v*2,2,…,*v*2,*m*)，欧几里得距离定义为：

在KNN中，计算*xtest*与训练样本*xi*的欧几里得距离为：

1. **曼哈顿距离 (Manhattan Distance)**

也称为城市街区距离或*L*1范数距离，衡量两点在坐标轴上的绝对差之和。

在KNN中，计算*xtest*与训练样本*xi*的曼哈顿距离为：

1. **闵可夫斯基距离 (Minkowski Distance)**

这是欧几里得距离和曼哈顿距离的推广，是*Lp*范数距离。

其中*p*是一个参数。

当*p*=1时，是曼哈顿距离。

当*p*=2时，是欧几里得距离。

当*p*→∞时，是切比雪夫距离 (Chebyshev Distance)，即各维度差的最大值：



在KNN中，计算*xtest*与训练样本*xi*的闵可夫斯基距离为：

**寻找K个最近邻 (Finding K Nearest Neighbors)**

计算出待预测样本*xtest*与所有训练样本的距离后，将这些距离进行排序，选出距离最小的K个训练样本。设这K个最近邻的集合为，对应的标签/数值集合为。

**预测 (Prediction)**

根据找到的K个最近邻，对新样本*xtest*进行预测。预测方式取决于任务是分类还是回归。

1. 分类 (Classification)

在分类问题中，KNN 使用多数投票 (Majority Voting) 的方式来决定*xtest*的类别。新样本被赋予其K个最近邻中出现次数最多的类别。

设*C*={*c*1,*c*2,…,*cL*}是所有可能的类别集合。新样本*xtest*的预测类别可以表示为：

其中是指示函数，当括号中的条件为真时取值为1，否则取值为0。这个公式的含义是：遍历所有可能的类别*c*，统计在K个最近邻中属于类别*c*的样本数量，选择数量最多的那个类别作为预测结果。

有时也会使用距离加权的投票，即距离越近的邻居权重越大。一种常用的加权方式是使用距离的倒数作为权重：

或者使用其他形式的权重函数，例如高斯核函数等。

1. 回归 (Regression)

在回归问题中，KNN 通常使用 K 个最近邻的标签值的平均值作为预测结果。

同样，回归问题也可以使用距离加权的平均，距离越近的邻居对预测值的贡献越大：

这个公式表示使用距离倒数作为权重的加权平均。