# kNN算法综述

王宇航13120476

(北京交通大学计算机与信息技术学院，北京，100044)

**摘要**：kNN算法是著名的模式识别统计学方法，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。本文对kNN算法及相关文献做一份总结，详细介绍kNN算法的思想、原理、实现步骤以及具体实现代码，并分析了算法的优缺点及其各种改进方案。本文还介绍了kNN算法的发展历程、重要的发表的论文。本文在最后介绍了kNN算法的应用领域，并重点说明其在文本分类中的实现。

**关键字**：kNN算法；k近邻算法；机器学习；文本分类

**Abstract:** KNN algorithm, a famous statistical method of pattern recognition, which is one of the best algorithms for dealing with text categorization, is playing an important role in machine learning classification algorithm, and it is one of the simplest algorithms in machine learning. This paper mainly summaries the kNN algorithm and its related literature, and detailed introduces its main idea, principle, implementation steps and specific implementation code, as well as analyzes the advantages and disadvantages of the algorithm and its various improvement schemes. This paper also introduces the development course of kNN algorithm, its important published paper. In the final, this paper introduces the application field of kNN algorithm, and especially in text categorization.

**Keywords:** KNN algorithm, K neighbor algorithm, Machine learning, Text classification

## 引言

分类是数据挖掘中的核心和基础技术，在经营、决策、管理、科学研究等多个领域都有着广泛的应用。目前主要的分类技术包括决策树、贝叶斯分类、kNN分类、人工神经网络等。在这些方法中，kNN分类是一种简单、有效、非参数的方法，现已经广泛应用于文本分类、模式识别、图像及空间分类等领域。本文从各个角度对kNN算法进行较为全面的总结。

本文的结构如下：

在第二部分，主要介绍kNN算法的基本原理、思想、实现步骤、Java实现代码以及发展历程和经典论文。

第三部分是对kNN算法的诸多不足之处进行的讨论，并给出一些改进的方案。

第四部分介绍的是kNN算法如何处理多标签数据。

第五部分介绍了kNN算法目前的主要应用领域，并着重说明了其在文本分类中的出色表现。

## kNN算法简介

### 算法引入

KNN算法是机器学习里面比较简单的一个分类算法，整体思想比较简单：计算一个点A与其他所有点之间的距离，取出与该点最近的k个点，然后统计这k个点里面所属分类比例最大的，则点A属于该分类。下面用一个例子来说明一下：

|  |  |  |  |
| --- | --- | --- | --- |
| 电影名称 | 打斗次数 | 接吻次数 | 电影类型 |
| California Man | 3 | 104 | Romance |
| He’s Not Really into Dudes | 2 | 100 | Romance |
| Beautiful Woman | 1 | 81 | Romance |
| Kevin Longblade | 101 | 10 | Action |
| Robo Slayer 3000 | 99 | 5 | Action |
| Amped II | 98 | 2 | Action |

简单说一下这个数据的意思：这里用打斗次数和接吻次数来界定电影类型，如上，接吻多的是Romance类型的，而打斗多的是动作电影。还有一部名字未知（这里名字未知是为了防止能从名字中猜出电影类型），打斗次数为18次，接吻次数为90次的电影，它到底属于哪种类型的电影呢？

KNN算法要做的，就是先用打斗次数和接吻次数作为电影的坐标，然后计算其他六部电影与未知电影之间的距离，取得前K个距离最近的电影，然后统计这k个距离最近的电影里，属于哪种类型的电影最多，比如Action最多，则说明未知的这部电影属于动作片类型。

在实际使用中，有几个问题是值得注意的：K值的选取，选多大合适呢？计算两者间距离，用哪种距离会更好呢？计算量太大怎么办？假设样本中，类型分布非常不均，比如Action的电影有200部，但是Romance的电影只有20部，这样计算起来，即使不是Action的电影，也会因为Action的样本太多，导致k个最近邻居里有不少Action的电影，这样该怎么办呢？

没有万能的算法，只有在一定使用环境中最优的算法。

### 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。

先计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。

### 算法计算步骤

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。

### 相似性度量

用空间内两个点的距离来度量。距离越大，表示两个点越不相似。距离的选择有很多[13]，通常用比较简单的欧式距离。

**欧式距离**：

**马氏距离**：马氏距离能够缓解由于属性的线性组合带来的距离失真，是数据的协方差矩阵。

**曼哈顿距离**：

**切比雪夫距离**：

**闵氏距离**：r取值为2时：曼哈顿距离；r取值为1时：欧式距离。

**平均距离**：

**弦距离**：表示2-范数，即

**测地距离**：

**Mean** **character** **difference**：

**Index** **of** **association**：

**Canberra** **metric**：

**Czekanowski** **coefficient**：

**Coefficient** **of** **divergence**：

### 类别的判定

投票决定：少数服从多数，近邻中哪个类别的点最多就分为该类。

加权投票法：根据距离的远近，对近邻的投票进行加权，距离越近则权重越大（权重为距离平方的倒数）

### 优缺点

#### 优点

1. 简单，易于理解，易于实现，无需估计参数，无需训练；
2. 适合对稀有事件进行分类；
3. 特别适合于多分类问题(multi-modal,对象具有多个类别标签)， kNN比SVM的表现要好。

#### 缺点

1. 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；
2. 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个邻居中大容量类的样本占多数；
3. 可解释性较差，无法给出决策树那样的规则。

### 常见问题

#### k值的设定

k值选择过小，得到的近邻数过少，会降低分类精度，同时也会放大噪声数据的干扰；而如果k值选择过大，并且待分类样本属于训练集中包含数据数较少的类，那么在选择k个近邻的时候，实际上并不相似的数据亦被包含进来，造成噪声增加而导致分类效果的降低。

如何选取恰当的K值也成为KNN的研究热点。k值通常是采用交叉检验来确定（以k=1为基准）。

经验规则：k一般低于训练样本数的平方根。

#### 类别的判定方式

投票法没有考虑近邻的距离的远近，距离更近的近邻也许更应该决定最终的分类，所以加权投票法更恰当一些。

#### 距离度量方式的选择

高维度对距离衡量的影响：众所周知当变量数越多，欧式距离的区分能力就越差。

变量值域对距离的影响：值域越大的变量常常会在距离计算中占据主导作用，因此应先对变量进行标准化。

#### 训练样本的参考原则

学者们对于训练样本的选择进行研究，以达到减少计算的目的，这些算法大致可分为两类。第一类,减少训练集的大小。KNN算法存储的样本数据,这些样本数据包含了大量冗余数据,这些冗余的数据增了存储的开销和计算代价。缩小训练样本的方法有:在原有的样本中删掉一部分与分类相关不大的样本样本,将剩下的样本作为新的训练样本;或在原来的训练样本集中选取一些代表样本作为新的训练样本；或通过聚类,将聚类所产生的中心点作为新的训练样本。

在训练集中，有些样本可能是更值得依赖的。可以给不同的样本施加不同的权重，加强依赖样本的权重，降低不可信赖样本的影响。

#### 性能问题

kNN是一种懒惰算法，而懒惰的后果：构造模型很简单，但在对测试样本分类地的系统开销大，因为要扫描全部训练样本并计算距离。

已经有一些方法提高计算的效率，例如压缩训练样本量等。

### 算法流程

1. 准备数据，对数据进行预处理
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，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
7. 组，将当前训练元组存入优先级队列。
8. 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。
9. 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。

### kNN算法的Java实现代码

|  |
| --- |
| **public** **class** KNN {  /\*\*  \* 设置优先级队列的比较函数，距离越大，优先级越高  \*/  **private** Comparator<KNNNode> comparator =**new** Comparator<KNNNode>(){  **public** **int** compare(KNNNode o1, KNNNode o2) {  **if** (o1.getDistance() >= o2.getDistance())  **return** -1;  **else**  **return** 1;  }  };  /\*\*  \* 获取K个不同的随机数  \* **@param** k 随机数的个数  \* **@param** max 随机数最大的范围  \* **@return** 生成的随机数数组  \*/  **public** List<Integer> getRandKNum(**int** k, **int** max) {  List<Integer> rand = **new** ArrayList<Integer>(k);  **for** (**int** i = 0; i < k; i++) {  **int** temp = (**int**) (Math.*random*() \* max);  **if** (!rand.contains(temp))  rand.add(temp);  **else**  i--;  }  **return** rand;  }    /\*\*  \* 计算测试元组与训练元组之前的距离  \* **@param** d1 测试元组  \* **@param** d2 训练元组  \* **@return** 距离值  \*/  **public** **double** calDistance(List<Double> d1, List<Double> d2) {  **double** distance = 0.00;  **for** (**int** i = 0; i < d1.size(); i++)  distance += (d1.get(i) - d2.get(i)) \*(d1.get(i)-d2.get(i));  **return** distance;  }    /\*\*  \* 执行KNN算法，获取测试元组的类别  \* **@param** datas 训练数据集  \* **@param** testData 测试元组  \* **@param** k 设定的K值  \* **@return** 测试元组的类别  \*/  **public** String knn(List<List<Double>> datas, List<Double> testData, **int** k) {  PriorityQueue<KNNNode> pq = **new** PriorityQueue<KNNNode> (k,comparator);  List<Integer> randNum = getRandKNum(k, datas.size());  **for** (**int** i = 0; i < k; i++) {  **int** index = randNum.get(i);  List<Double> currData = datas.get(index);  String c = currData.get(currData.size() - 1).toString();  KNNNode node = **new** KNNNode(index, calDistance(testData, currData), c);  pq.add(node);  }  **for** (**int** i = 0; i < datas.size(); i++) {  List<Double> t = datas.get(i);  **double** distance = calDistance(testData, t);  KNNNode top = pq.peek();  **if** (top.getDistance() > distance) {  pq.remove();  pq.add(**new** KNNNode(i, distance, t.get(t.size() - 1). toString()));  }  }  **return** getMostClass(pq);  }    /\*\*  \* 获取所得到的k个最近邻元组的多数类  \* **@param** pq 存储k个最近近邻元组的优先级队列  \* **@return** 多数类的名称  \*/  **private** String getMostClass(PriorityQueue<KNNNode> pq) {  Map<String, Integer> classCount=**new** HashMap<String,Integer>();  **int** pqsize = pq.size();  **for** (**int** i = 0; i < pqsize; i++) {  KNNNode node = pq.remove();  String c = node.getC();  **if** (classCount.containsKey(c))  classCount.put(c, classCount.get(c) + 1);  **else**  classCount.put(c, 1);  }  **int** maxIndex = -1;  **int** maxCount = 0;  Object[] classes = classCount.keySet().toArray();  **for** (**int** i = 0; i < classes.length; i++) {  **if** (classCount.get(classes[i]) > maxCount)  maxIndex = i; maxCount = classCount.get(classes[i]);  }  **return** classes[maxIndex].toString();  }  } |

### 经典文献

kNN算法是对NN（nearest neighbor）算法即近邻算法的改进，最初的近邻算法是由T.M. Cover在其文章“Rates of Convergence for Nearest Neighbor Procedures,”中提出的，是以全部训练样本作为带标点，计算测试样本与所有样本的距离并以最近邻者的类别作为决策，后学者们对近邻算法进行了各方面的改进。其中一个方向就是KNN算法，最初的KNN算法是由谁提出的我现在有两个怀疑，一个是Trevor Hastie提出KNN算法的人，我找到了他的那篇文献Discriminant Adaptive Nearest Neighbor Classification，但是在文章最后作者引用了R. Short &d K. Fukanaga的 “A’New Nearest Neighbor Distance Measure, 并且声称In our terminology they used our metric with W = I and E = 0, with B determined locally in a neighborhood of size K是否R. Short 就已经提出了KNN的概念，有待我进一步阅读相关文献。

|  |  |
| --- | --- |
| T.M. Cover | Nearest neighbor pattern classification, Proc. IEEE Trans. Information Theory, 1967  Rates of Convergence for Nearest Neighbor Procedures，Proc. Hawaii Int’l Conf. Systems Sciences，1968 |
| C. J. Stone | Consistent nonparametric regression, Ann. Stat., vol.3, no. 4, pp. 595-645, 1977. |
| W Cleveland | Robust Locally-Weighted Regression and Smoothing Scatterplots, J. Am. Statistical Soc., vol. 74, pp. 829-836, 1979. |
| T. A. Brown & J. Koplowitz, | The weighted nearest neighbor rule for class dependent sample sizes, IEEE Trans. Inform. Theory, vol. IT-25, pp. 617-619, Sept. 1979. |
| R. Short &K. Fukanaga, | A New Nearest Neighbor Distance Measure, Proc. Fifth IEEE Int’l Conf. Pattern Recognition, pp. 81-86, 1980.  The Optimal Distance Measure for Nearest Neighbor Classification,” IEEE Trans. Information Theory 1981 |
| J.P. Myles and D.J. Hand, | The Multi-Class Metric Problem in Nearest Neighbor Discrimination Rules, Pattern Recognition,1990 |
| N. S. Altman | An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression,1992 |
| Min-Ling Zhang & Zhi-Hua Zhou | Ml-kNN: A Lazy Learning Approach to Multi-Label Learning,2007 |
| Peter Hall, Byeong U. Park & Richard J. Samworth | Choice Of Neighbor Order In Nearest-Neighbor Classification, 2008 |
| Jia Pan & Dinesh Manocha | Bi-level Locality Sensitive Hashing for K-Nearest Neighbor Computation,2012 |

## 算法改进

kNN算法因其提出时间较早，随着其他技术的不断更新和完善，kNN算法的诸多不足之处也逐渐显露，因此许多kNN算法的改进算法也应运而生。

针对以上算法的不足，算法的改进方向主要分成了分类效率和分类效果两方面。

分类效率：事先对样本属性进行约简，删除对分类结果影响较小的属性，快速的得出待分类样本的类别。该算法比较适用于样本容量比较大的类域的自动分类，而那些样本容量较小的类域采用这种算法比较容易产生误分。

分类效果：采用权值的方法（和该样本距离小的邻居权值大）来改进，Han等人于2002年尝试利用贪心法，针对文件分类实做可调整权重的k最近邻居法WAkNN (weighted adjusted k nearest neighbor)，以促进分类效果；而Li等人于2004年提出由于不同分类的文件本身有数量上有差异，因此也应该依照训练集合中各种分类的文件数量，选取不同数目的最近邻居，来参与分类。

下面具体说明主要的改进方向，然后简单举一个算法改进的实例。

### 主要改进方向

#### 从降低计算复杂度提高算法的执行效率

KNN算法存储训练集的所有样本数据，这造成了极大的存储开销和计算代价。已有很多的文献提出减少计算的算法，这些算法大致可分为两类。第一类，减少训练集的大小。KNN算法存储的样本数据，这些样本数据包含了大量冗余数据，这些冗余的数据增了存储的开销和计算代价。缩小训练样本的方法有:在原有的样本中删掉一部分与分类相关不大的样本样本，将剩下的样本作为新的训练样本；或在原来的训练样本集中选取一些代表样本作为新的训练样本；或通过聚类，将聚类所产生的中心点作为新的训练样本。主要方法的文献[25]-[26]。这些方法筛选合适的新训练样本，对于大训练样本集，这个工作量是非常巨大的。第二类，采用快速算法，快速搜索到K个最近邻。KNN算法要找到K个最近邻的点，则要计算测试点到所有训练样本的距离，然后找出其中K个距离最小有数据点，当训练样本非常大时，KNN算法就不切实际了，为了加快KNN搜索过程，主要的方法，其中一个方法是部分距离计算，文献[27]中提出一种基于小波域部分距离计算的KNN搜索算法，文献[28]提出快速算法(KWENNS)。另外一种方法是，引入高效的索引方法，高效的索引方法可以大大降低K个最近邻的计算开销，特别是在高维空间中体现更为明显，文献[29]提出了一种新的索引结存模型，有的算法虽然能够有效降低K个最近邻的计算开销，提高了KNN的分类速度，但它们无法保证进行全局的最优搜索。

#### 优化相似度度量方法

基本的KNN算法基于欧基里德距离来计算相似度，这种计算距离的度量标准造成了KNN算法对噪声特征非常敏感。为了改变传统KNN算法中特征作用相同的缺陷，可在度量相似度的距离公式中给特征赋予不同权重，特征的权重一般根据各个特征在分类中的作用设定。可根据特征在整个训练样本库中的分类作用得到权重，也可根据其在训练样本的局部样本（靠近待测试样本的样本集合）中的分类作用得到权重。人们研究了各种学习调整权值的方法，从而提高了KNN分类器的性能。

#### 优化判决策略

传统KNN的决策规则一个明显的缺点是，当样本分布密度不均匀时，只按照前K个近邻顺序而不考虑它们的距离会造成误判，影响分类的性能。而且在实际设计分类器时，由于一些类别比另一些类别的训练样本更容易获得，往往会造成训练样本各类别之间目数不均衡，即是训练样本在各个类中的数目基本接近，由于其所占区域大小的不同，也会造成训练样本的分布不均匀。目前改进的方法有均匀化样本分布密度；文献[30]等对kNN的决策规则进行了改进，很好地解决了当各类数据分布不均匀时kNN分类器分类性能下降的问题，文献[31]利用大量近邻集来代替KNN中的单一集合，并通过累加近邻的数据集对不同类别的支持度，获得相对可信的支持值，从而改善了近邻判决规则。

#### 选取恰当的K值

由于KNN算法中几乎所有的计算都发生在分类阶段，而且分类效果很大程度上依赖于k值的选取，k值的选择很重要。k值选择过小，得到的近邻数过少，会降低分类精度，同时也会放大噪声数据的干扰；而如果k值选择过大，并且待分类样本属于训练集中包含数据数较少的类，那么在选择k个近邻的时候，实际上并不相似的数据亦被包含进来，造成噪声增加而导致分类效果的降低。如何选取恰当的K值也成为KNN的研究热点。

#### 多种算法集成

除了上述的各种方法外，也有研究者将KNN分类方法和其他算法进行集成从而提高KNN分类器的分类性能。有将SVM和KNN进行集成，将KNN、Grouping和LSA进行集成，将遗传算法和模糊KNN进行集成，将贝叶斯分类器和KNN分类器进行集成，将P-tree和KNN相结合，通过和其他算法集成从而提高了KNN分类器的分类性能。

### 类相关度差异优化距离的KNN改进算法

改进思想：将样本特征参数的熵值与样本分布概率的乘积作为特征参数针对分类的相关度，并根据相关度值衡量特征参数对分类影响程度的强弱以计算样本间的距离，解决KNN近邻选择时大类别、高密度样本占优的情况。

#### 特征参数类相关度差异优化距离机制

特征参数对分类的相关度的定义。设是训练样本集，它包含了个不同类别的样本，这些类别分别用表示，样本记为。设具有个特征属性，具有口个特征参数，记为具有特征参数样本的个数，记为特征参数在，类样本中出现的次数，则特征参数的类相关度(期望信息)为

|  |  |
| --- | --- |
|  | (1) |

其中，为具有特征参数样本的概率，为特征参数在，中出现的概率；熵值体现了特征参数对样本隶属的期望值，是特征参数对分类作用的不确定性的度量，越小，对分类的作用越大。当时，，对样本隶属类完全确定。以为主体，当时，值无意义，即特征参数对样本隶属类明确，与样本的分布无关联。

样本间的距离的定义

|  |  |
| --- | --- |
|  | (2) |

其中，是样本特征参数相关性的转化，是样本特征参数相关性的转化，将特征参数类相关度值替代马氏距离机制中原数据的特征，并赋予相应的权重值以计算样本间的相似程度。由于特征参数相似性是分类训练器中潜在的关联关系，它的存在可以减少样本间的距离的计算误差。因而，类相关度能有效地度量样本间特征的相似程度，进而最大程度地提取与待分样本相似的近邻样本，从而得到正确的分类结果。

#### 特征参数类相关度优化距离的K最近邻改进算法

针对引言中KNN算法分类准确性和效率的弱点，采用一种新的数据预处理机制改进算法，将特征参数类相关度概念引入KNN分类。大量的研究表明，训练数据的类别如果不再细分的话，类别数不是很多，特征参数的重复率也较高，尤其对于一些常用的数据更是如此。如果使用降维方法对数据进行相似度计算，总会因为数据向量维数偏低或维数不适中而丢失过多的，甚至是重要的特征信息，从而影响分类效果。基于此因素考量，改进算法仍采用传统KNN算法中将训练集样本与待分样本的所有特征属性值均作为相似度计算参数的模式，主要通过优化距离机制，从根本上保证KNN分类的准确性及效率。

采用类相关度优化距离的KNN改进算法的实现：

输入：训练集与测试集合表示为,…,，，。

输出：测试集的类别标签。

(1)根据公式(1)计算训练集中每个样本及待分样本的所有特征参数对分类的相关度Diff，由Diff值向量化数据特征集，对样本参数进行相关度值计算预处理，实现基于类相关度的样本特征提取。

for(int i=1；i<m；i++)

//依次计算X；，石包含的q个特征参数t的相关度；

for(j=1；g；，++)

if 不存在

检索，收集参数集，由公式(1计算 ，并存储值；

存储新矩阵，其中，

(2)使用公式(2)计算待分样本与训练集各样本的距离。

for(int i=1；i<m；i++)

//依次计算戈与X；包含的距离

for(j=1；g；J++)

由公式(2累加求平均距离值。

(3)判断样本的类别归属。

按照和的距离远近排序，得到最近的k个样本，根据这k个样本的类别得到的类别。

### 基于聚类的KNN算法改进

改进思想：首先将训练集文本进行聚类，将训练集文本分为若干个簇，然后采用KNN 算法对测试文档进行测试，最后用距离最近的n 个簇中的若干训练集文本使用KNN算法对测试文本进行分类。因其降低了计算量所以改善了执行的效率。

#### 对训练集进行聚类

第一步：如果文本对象P未被归入某个簇或标记为噪声，就检查它的指定半径邻域r，如果指定半径邻域内包含的对象数目大于等于给定的值m，就建立新簇C，将p的指定半径领域r中所有点加入该簇C；

第二步：对C中所有尚未被处理(归入某个簇或标记为噪声)的对象q，检查它的指定半径邻域，如果该邻域内包含对象数目大于等于给定的值m，将该邻域中没有归入任何一个簇的对象加入C；

第三步：重复第二步，继续检查C中未被处理对象，直到没有新的对象加入当前簇C：

第四步：重复以上步骤，直到所有对象都被处理。

其中关键参数为作为密度计算的距离表示的半径，密集点所必需的在指定半径内拥有的最少的其他点的数目。通过这两个参数我们就可以计算在任何点周围的密度值。

这样，训练集中文本就聚为若干个类了。每个簇的类别由簇中多数文本类别而定。

#### 用kNN算法分类

结合KNN算法，计算测试集文本与训练集文本簇之间的距离，这样可以减少计算量和个别孤立点对测试集文本的影响。

第一步：对于任一个给定的测试集文本，计算与训练集中各个簇的距离，采用（2）式为测试集文本评分

其中x是一个测试集文本，c是训练集的类别，t是距离x最近的k个文本簇之一。是文本x与文本t簇的相似度，这里指的是距离。是表示t簇是否属于类c，如果属于类c则为1，否则为0。

第二步：根据评分结果进行排序，选取前k个簇。

第三步：从这些簇中选取n个与测试集文本最近的文本，按照（1）式评分，判定该测试集文本类别，回归到传统的KNN算法。

改进算法中有领域半径r，指定邻域内最小文本数m，选取簇类个数k，从k簇中选取距离最小的n个文本这几个参数。

## kNN算法分类多标签数据

数据分类是数据挖掘领域的重要分支，可表达为函数，其任务是赋予有序对一个布尔值，其中D表示数据实例的定义域，表示预定义的主题类集，亦称标签集(Label Set)。如果数据实例4拥有标签则，否则烈。在实际应用中数据分类存在两种情况：仅拥有一个标签的情况称为单标签分类(Single-Label Categorization)；拥有0至个标签的情况称为多标签分类(Multi-Label Categorization)，所谓标签是指数据实例所拥有的主题类。数据分类器有全自动和半自动之分。全自动分类器会做出关于为T或F的明确决策，称之为“硬”分类(“Hard”Categorization)；而半自动分类器则根据拥有的可能性大小，仅给出为T或F的分类信度排序(Confidence Ranking)，最终的明确决策则由专家根据信度排序结果做出，称之为排序分类(Ranking Categorization)。

互联网信息资源中存在着大量诸如文本数据、图像数据和音乐数据等的多标签数据分类问题。Sebastiani[15]和Tsoumakas[16]先后于2002年和2007年系统梳理了文本分类和多标签数据分类的研究进展。根据求解思路的不同，可将多标签分类算法分为问题转换和算法适应两类，前者将多标签分类问题转换为多个单标签分类问题，后者则通过对C4.5、AdaBoost等单标签分类算法的扩展而形成多标签分类算法。最近又出现了一些新的适应实际问题不同特点的多标签分类算法[17]-[21]。其中，Zhang和Zhou[20]的ML-KNN是一种KNN(K-Nearest Neighbor)与贝叶斯法则相结合的排序分类算法，具有思路简单、非参数化和性能优越等优点，其局限性在于计算量大、分类效率偏低，因此不适用于实时性要求较高的场合。

### 最近邻选择的相似度定义

两个基本的求解步骤组成：(1)从已分类数据实例集中选择待分类数据实例的个最近邻；(2)根据已分类的个最近邻对进行排序分类。

设预先定义的数据实例特征集为。根据向量空间模型，可表示为由其特征值构成的特征向量。因此，待分类实例可表示为特征向量，与的距离大小反映了两者之间的相似程度，可作为从中选择的个最近邻的依据。关于两个向量之间的距离，其他学者已提出了多种计算公式[22]-[24]，比如对于和而言，基于欧氏距离(Euclidean Distance)的相似度可定义为：

|  |  |
| --- | --- |
|  | (3) |

而基于曼哈顿距离(Manhattan Distance)的相似度则可定义为：

|  |  |
| --- | --- |
|  | (4) |

不同的相似度定义将产生不同的最近邻选择结果，从而对分类效果产生很大影响。

### 最近邻选择的相似度定义

任意已分类数据实例的标签集可表示为，其中

根据待分类数据实例的个最近邻进行该实例的排序分类是本文算法的基本思想。首先根据相似度从中选择个的最近邻形成集合；然后找出最近邻中拥有标签数的最大值，统计拥有中的最近邻数及其最大值；最后按公式(5)和(6)分别确定的及其排序值：

|  |  |
| --- | --- |
|  | (5) |
|  | (6) |

其中threshold和const为常数，s为拉普拉斯平滑因子。具体

### 算法的时问复杂度分析

在上述FKMC中，仅利用的个最近邻的局部信息进行的排序分类，省去了非常耗时的全局训练过程，这样可以极大地降低计算复杂度。其最近邻选择部分的时间复杂度为，其余部分的时间复杂度为，所以总体时间复杂度为。分析文献[20]以看出，ML-KNN利用已分类数据实例集的全局信息，先根据其中每个实例的个最近邻计算该实例拥有每一个标签的先验概率和最大后验概率，并据此进行分类器的训练，然后将训练后的分类器用于的排序分类过程，其运行时间主要花在了计算已分类数据实例集中的每个实例拥有标签集中每个标签的先验概率和后验概率上，其时间复杂度为，其余部分的时间复杂度为，所以总体时间复杂度为。很显然，在分类效率方面FKMC必然会高于ML-KNN。

## kNN算法的应用

kNN算法作为最经典的机器学习分类算法之一，必然有其十分广泛的应用。在这里仅仅列举一些常见的应用，并重点介绍以下kNN算法在文本分类中的应用。

### kNN算法的主要应用领域

1. 模式识别，特别是光学字符识别；
2. 统计分类；
3. 计算机视觉；
4. 数据库，如基于内容的图像检索；
5. 编码理论（最大似然编码）；
6. 数据压缩（MPEG-2标准）；
7. 向导系统；
8. 网络营销；
9. DNA测序；
10. 拼写检查，建议正确拼写；
11. 剽窃侦查；
12. 相似比分算法，用来推断运动员的职业表现。

### kNN算法处理文本分类问题

#### 文本分类介绍

文本自动分类最初是应信息检索（IR）系统的要求而出现的。随着全球互联网络的普及，文本自动分类对于信息处理的意义变得更加重要。在互联网上，电子文档信息每 天都在急剧增加，通过网络，人们可以很方便地共享巨大的信息资源。但是，网络信息 的快速膨胀，信息资源无法有效利用。面对 网上的海量信息，传统的做法是，对网上信息进行人工分类， 并加以组织和整理，为人们提供一种相对有效的信息获取手 段。但这种人工分类的做法存在着许多弊端：一是耗费大量 的人力、物力和精力；二是分类结果一致性不高。即使分类 人的语言素质较高，对于不同的人来分类，其分类结果仍然 不尽相同。甚至同一个人，在不同时间做分类也可能会有不 同的结果。网络信息的激增一方面增加了对于快速、自动文 本分类的迫切需求。另一方面又为基于机器学习的文本分类 方法准备了充分的资源。电子化信息的自动分类处理技术正 越发显示着其优越性，文本自动分类及其相关技术的研究也 正日益成为一项研究热点。

文本分类主要应用于信息检索，机器翻译，自动文摘，信息过滤，邮件分类等任务。文本分类在搜索引擎中也有着大量的使用，网页分类/分层技术是检索系统的一项关键技术，搜索引擎需要研究如何对网页进行分类、分层，对不同类别的网页采用差异化的存储和处理，以保证在有限的硬件资源下，提供给用户一个高效的检索系统，同时提供给用户相关、丰富的检索结果。在搜索引擎中，文本分类主要有这些用途：相关性排序会根据不同的网页类型做相应的排序规则；根据网页是索引页面还是信息页面，下载调度时会做不同的调度策略；在做页面信息抽取时，会根据页面分类的结果做不同的抽取策略；在做检索意图识别的时候，会根据用户所点击的url所属的类别来推断检索串的类别。

#### 文本分类过程

以Internet中的文本为例，待分类文本以HTML格式存储的半格式化的Web 页面、文档为主，也是当前 Internet 信息的主要组织形式。文本知识挖掘就是要发现其中隐含的规则，以便于实现 Internet数据挖掘的智能化，离开了文本知识挖掘，智能化是不能实现的。最常用的文本知识挖掘方法是基于文档特征向量空间模型（Characteristic Vector Space Model，CVSM）的。

1. **文档模型建立**
2. 预处理过程。一是要根据禁用词集去除文档中的语义虚泛的禁用词；二是要利用特征词典集（包括通用集和专业集）进行分词，如果出现词集中没有的词，则将它整体作为一词并记录以便人工分词。
3. 概念映射和概念消歧。有些词形式不同但概念相同，要求根据概念集将它们映射为同一概念；对于未登录词，则选择与之共现率最多的词作为其概念；对于一词具有多概念标注的，选择概念标注出现次数最多者为其标注。
4. 一般特征项提取和姓名日期数字等特征抽取，结果存入文档矢量库。
5. 特征集缩减。通过以上方法得到的特征集数目巨大，所以必须对其进行缩减。其算法一般是构造一个评价函数，对每个特征向量进行评估，然后根据评估值的大小选取一定数量或超过阈值的特征向量子集。特征集缩减的结果存入文档矢量库。
6. **知识发现**
7. 文本摘要。采用基于统计的自动生成方式较多，其基本思想是把文中与主题密切相关的句子挑选出来，这样的句子往往位于特殊的部分或含有较多的特征项，一般以句子权重函数为评价标准。
8. 文本分类。文本分类是文本知识挖掘的主要目的，基本思想是将训练集、矢量集与文档矢量集相比较，方法有朴素贝叶斯分类算法和K－最近邻居分类算法等。
9. **模型评价**

文本评价的模型比较多，一般是将数据集分为训练集与测试集。学习—测试循环反复执行，最后用一个指标来衡量模型质量。模型评价具体指标有分类正确率、查准率、查全率、查准率、查全率的平均和信息估值等。

#### kNN算法实现文本分类

文本自动分类的一个关键问题是如何构造分类函数（分类器），并利用此分类函数将待分类文本划分到相应的类别空间中。训练方法和分类算法是分类系统的核心，这里介绍采用KNN分类算法对文本知识进行类别学习。

kNN算法的基本思路是：在给定新文本后，考虑在训练文本集中与该新文本距离最近（最相似）的 K 篇文本，根据这 K 篇文本所属的类别判定新文本所属的类别，具体的算法步骤如下：

1. 根据特征项集合重新描述训练文本向量；
2. 在新文本到达后，根据特征词分词新文本，确定新文本的向量表示；
3. 在训练文本集中选出与新文本最相似的K个文本，计算公式为：  
   其中，K 值的确定目前还没有很好的方法，一般采用先定一个初始值，然后根据实验测试的结果调整 K 值，一般初始值定为几百到几千之间。
4. 在新文本的 K个邻居中，依次计算每类的权重，计算公式：  
   其中，为新文本的特征向量，为相似度计算公式，与上一步骤的计算公式相同，而为类别属性函数，即如果属于类，那么函数值为1，否则为0。
5. 比较类的权重，将文本分到权重最大的那个类别中。

参考文献

|  |  |
| --- | --- |
| [1] | Cover, Hart T, P. Nearest neighbor pattern classification [J]. IEEE, 1967(1):21 - 27. |
| [2] | Cover T. Rates of Convergence for Nearest Neighbor Procedures [J]. Systems Sciences, 1968. |
| [3] | Stone C J. Consistent Nonparametric Regression [J]. Institute of Mathematical Statistics, 1977(7), 5(4):595-620. |
| [4] | Cleveland W S. Robust Locally Weighted Regression and Smoothing Scatterplots [J]. Journal of the American Statistical Association, 1979, 74:829-836. |
| [5] | Brown, T., Koplowitz, Jack. The weighted nearest neighbor rule for class dependent sample sizes (Corresp.)[J]. IEEE, 1979(9).IT-25:617 - 619. |
| [6] | Short R, Fukunaga K. A new nearest neighbor distance measure [J]. IEEE, 1980:81-86. |
| [7] | Short, Robert D.; Fukunaga, K. The Optimal Distance Measure for Nearest Neighbor Classification [J]. IEEE, 1981(9), 27(5):622 - 627. |
| [8] | Myles J, Hand D. The Multi-Class Metric Problem in Nearest Neighbor Discrimination Rules [J]. Pattern Recognition, 1990, 23(11):1291–1297. |
| [9] | Altman N S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression [J]. , 1992, 46(3):175-185. |
| [10] | Zhang M, Zhou Z. ML-KNN: A lazy learning approach to multi-label learning [J]. Pattern Recognition, 2007(7), 40(7):2038–2048. |
| [11] | Hall P, Samworth B U P A R J. Choice Of Neighbor Order In Nearest-Neighbor Classification [J]. The Annals of Statistics, 2008(10), 36(5):2135-2152. |
| [12] | Pan J, Manocha, D. Bi-level Locality Sensitive Hashing for k-Nearest Neighbor Computation [J]. IEEE, 2012(4):378 - 389. |
| [13] | Michel M Deza, Elena Deza. Encyclopedia of Distances. Springer, 2009 |
| [14] | 周靖, 刘晋胜. 一种采用类相关度优化距离的KNN算法[J]. 微计算机应用, 2010(11), 31(11):7-12. |
| [15] | Sebastiani F．Machine learning in automated text categorization [J].ACM Computing Surveys, 2002, 34(1):1-47. |
| [16] | 赵继东，鲁坷，吴跃．一种基于谱图理论的web图像搜索方法[J].计  算机应用研究,2008(5):12-13. |
| [17] | 张华．www图像语义信息提取方法研究[D].济南：山东师范大学,2004. |
| [18] | 温小斌．Interact图像搜索引擎的研究与实现[D]．海口：海南大学,2006. |
| [19] | Cai Dang，He Xiaofei，Li Zhiwei，et a1．Hieraehical clustering of WWW image search results using visual．textual and link Information[C] //Proceedings of the ACM International Conference on Multimedia，New York，USA，2004：952—959． |
| [20] | Cheng En, Jing Feng, Zhang Chao, et a1.Search result clustering based relevance feedback for web image retrieval [C] //Interactional Conference on Acoustics, Speech, and Signal Processing, Hawaii, 2007:961-964. |
| [21] | 谢同．基于文本的Web图片搜索引擎的研究与实现[D].成都：电子科技大学,2007. |
| [22] | Cai D, Yu S, Wen J R, et a1.VIPS:a vision-based page segmentation algorithm,MSR-TR-2003-79[R].Microsoft Research,2003. |
| [23] | 亢世勇，刘艳．汉语动词谓语句的语义成分和语义句式[J].唐都学刊.1998, 14(1):89-93. |
| [24] | 徐斌基于PCFG-HDSM模型的语义句式识别[D].南京:南京航天航空大学,2008. |
| [25] | P E Har. The condensed nearest neighbor rule. IEEE Trans on Information Theory, 1968, IT-14(3):515-516. |
| [26] | 李荣陆,胡运发.基于密度的kNN文本分类器训练样本裁剪方法.计算机研究与发展, 2004, 41(4):539-546. |
| [27] | W J Hwang, K W Wen. Fast KNN classification algorithm based on partial distance search [J].Electron lett, 1998, 34(21):2062\_2063. |
| [28] | J S Pan, Y L Qiao, S H Sun. A fast K nearest neighbors classification algorithm [J].IEICE Trans Fundamentals, 2004, E87-A (4):961-961. |
| [29] | 侯士江,刘车华,余靖,褚兵义.空间网络数据库中的K个最近邻查询算法.计算机科学,2006Vol.33No.8. |
| [30] | 孙秋月.基于SNN相似度的KNN算法研究.云南大学硕士学位论文, 2008. |
| [31] | H. Wang. Nearest Neighbors without k: A Classification Formalism based on Probability, technical report, Faculty of Informatics, University of Ulster, N, Ireland, UK, 2002. |