基于密度的 *k*NN 文本分类器训练样本裁剪方法

李荣陆 胡运发
(复旦大学计算机与信息技术系 上海 200433)
(lironglu@163.net)

摘 要 随着 WWW 的迅猛发展,文本分类成为处理和组织大量文档数据的关键技术. *k*NN 方法作为一种简单、有效、非参数的分类方法,在文本分类中得到广泛的应用. 但是这种方法计算量大,而且训练样本的分布不均匀会造成分类准确率的下降. 针对 *k*NN 方法存在的这两个问题,提出了一种基于密度的 *k*NN 分类器训练样本裁剪方法,这种方法不仅降低了 *k*NN 方法的计算量,而且使训练样本的分布密度趋于均匀,减少了边界点处测试样本的误判. 实验结果显示,这种方法具有很好的性能.

关键词 文本分类; *k*NN; 快速分类

中图法分类号 TP391; TP18

A Density-Based Method for Reducing the Amount of Training Data in *k*NN Text Classification

LI Rong-Lu and HU Yun-Fa
(*Department of Computing and Information Technology, Fudan University, Shanghai* 200433)

Abstract With the rapid development of World Wide Web, text classification has become the key technology in organizing and processing large amount of document data. As a simple, effective and nonparametric classification method, *k*NN method is widely used in document classification. But *k*NN classifier not only has large computational demands, but also may decrease the precision of classification because of the uneven density of training data. In this paper, a density-based method for reducing the amount of training data is presented, which solves two problems mentioned above. It not only reduces the computational demands of *k*NN classifier, but also makes the density of training data even and decreases the wrong classification between the edge of classes. The experiment also shows that it has good performance.

Key words text classification; *k*nearest neighbor; fast classification

1 引 言

随着 World Wide Web 的迅猛发展,在线文档信息的迅速增加,文本分类成为处理和组织大量文档数据的关键技术. 现有的分类方法主要是基于统计理论和机器学习方法,比较著名的文档文类方法有 Bayes[1], *k*NN[2], LLSF[3], Nnet[4], Boosting[5] 及 SVM[6] 等. 在这些方法中,*k*NN(*k*nearest neighbor)作为一种简单、有效、非参数的方法,在文本分类中得到广泛使用,并且取得了很好的效果. 其基本思想是在训练样本中找到测试样本的 *k* 个最近邻,然后根据这 *k* 个最近邻的类别来决定测试样本的类别. *k*NN 分类是一种基于要求的或懒惰的学习方法,它存放所有的训练样本,直到测试样本需要分类时才建立分类,这样与测试样本比较的可能近邻数量(即训

收稿日期: 2002-10-25; 修回日期: 2003-06-09
基金项目: 国家自然科学基金项目(60173027)

练样本)较大时, 会招致很高的计算开销. 而且, 训练文档分布的不均匀性也会造成分类准确率的下降.

目前主要通过两种途径来减小 kNN 方法的计算量: 一种途径是通过快速搜索算法, 在尽量短的时间内找到测试样本的最近邻[7~8]; 另一种途径是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, 或删除原来的训练样本集中的某些样本, 并将剩下的样本作为新的训练样本, 从而达到减小训练样本集的目的. 本文主要讨论后一种途径. 对于这种途径最主要的方法是 Hart 的 Condensing 算法[9]、Wilson 的 Editing 算法[10] 和 Devijver 的 MultiEdit 算法[11], Kuncheva 使用遗传算法在这方面也进行了一些研究[12,13]. 但是这些方法在训练样本集中每增加或删除一个样本时, 都要对样本进行一次测试, 反复迭代直到样本集不再变化, 这对于有成百上千的训练文档来说, 其工作量是非常巨大的. 并且, 这些方法都没有考虑训练样本分布的不均匀性对分类准确率的影响.

本文提出了一种基于密度的 kNN 文本分类器训练样本裁剪方法, 根据训练样本的分布密度对其进行裁剪, 使训练样本的分布尽量均匀, 而且只需要两次迭代就可完成样本的裁剪. 实验结果显示, 这种方法不仅减少了训练样本的数量, 使 kNN 方法的计算量降低, 而且削弱了训练样本分布的不均匀性对分类性能的影响, 提高了分类的准确率和召回率.

2 训练样本分布密度对分类结果的影响

设 n 维向量 $\boldsymbol{X}=(x_1, x_2, \cdots, x_n)$ 为一个文档, $C_i=(\boldsymbol{X}_1^i, \boldsymbol{X}_2^i, \cdots, \boldsymbol{X}_{n_i}^i)$ 为包含 n_i 个文档的、具有类标识 C_i 的文档集. 具有 m 个类 $C_1, C_2, \cdots, C_m$ 的文本分类问题, 其 kNN 分类决策过程如下: 对于一个给定的测试文档 $\boldsymbol{X}$, 分别计算它与训练样本集中每一个文档的距离(或相似度), 找到与之最近的 k ($k\geqslant 1$)个训练文档, 其中属于 C_i 类的文档数有 k_i 个, 则我们定义判别函数为

$$g_i(\boldsymbol{X})=k_i,\ i=1,2,\cdots,m,$$

那么分类的决策规则为

若 $g(\boldsymbol{X})=\arg\max_j(k_j)$, $j=1,2,\cdots,m$, 则决策 $\boldsymbol{X}\in C_j$.

kNN 方法实际上是一种基于类比的学习方法, 这就要求训练样本中样本必须具有代表性, 这种代表性不仅应该体现在样本间的距离(或相似度)上, 还应该体现在样本分布的均匀性上. 为了描述方便, 下面我们以二维空间两种分类为例, 看一下训练样本的分布密度对 kNN 分类器分类结果的影响.

从图 1 我们可以看到 kNN 方法存在如下两个问题:

问题 **1**. 在类边界区域, 训练样本分布的不均匀性可能会造成测试样本类别的误判. 在图 1 中, 我们可以直观地看到测试样本应该属于类 2, 但是由于类 1 比类 2 的分布密度要大, 这样当我们选测试样本的 10 个最近邻来判别它的类别时, 分类器就出现了误判. 如果 k 值更大些, 则误判更为明显. 而在实际设计分类器的时候, 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分布的不均匀; 而且, 即使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, 也会造成训练样本分布的不均匀.

问题 **2**. 靠近类中心区域的大多数训练样本对分类决策没有什么用处.

根据近邻规则, 如果测试样本不在类边界区域, 即使类中心区域没有训练样本, 只要在类边界区域有适当的训练样本, 那么根据这些边界区域的训练样本, 也可以实现对测试样本的正确分类; 如果测试样本在类边界区域, 那么它的最近邻也主要集中那些处于边界区域的样本中, 类中心区域的大多数训练样本在这里不会起到什么作用. 但是, 如果将类中心区域的样本全部裁剪掉, 也会产生一些问题. 因为在实际的训练样本中, 类与类之间的边界并不是非常明显, 而且会有一些交叉. 所以, 较为合适的解决方法是降低类中心区域的密度, 适当保留一些处于类中心区域的训练样本.

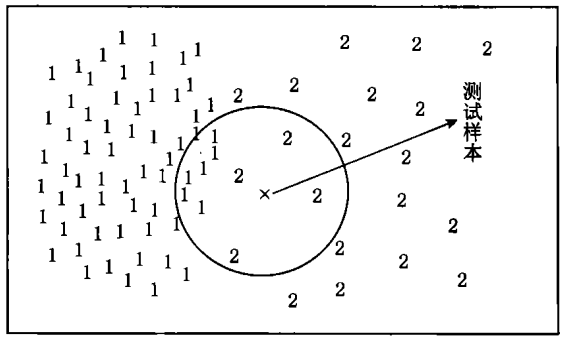

图 1 训练样本分布密度对分类结果影响示例图($k=10$)

我们提出了一种基于密度的 kNN 分类器训练样本裁剪方法, 根据训练样本的分布密度对其进行裁剪, 较好地解决了以上两个问题.

3　基于密度的 kNN 分类器训练样本裁剪方法

3.1　基本概念

为了对训练样本的分布密度进行衡量, 从而实现训练样本的裁剪, 我们引入以下一些概念. 给定一个样本集 $D=\{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_l\}$, 其中 $\boldsymbol{X}_i \in \mathrm{R}^n$, $i=1, \cdots, l$, 我们定义:

定义 **1**. 设 $Dist(\boldsymbol{X}, \boldsymbol{Y})$代表样本集 D 中两个样本X 和Y 间的距离(或相似度), 则对于任意的 $\boldsymbol{X} \in D$, 我们定义 $\boldsymbol{X}$ 的 ε 邻域为

$$N_\varepsilon(X)=\{Y \mid Dist(\boldsymbol{X}, \boldsymbol{Y}) \leqslant \varepsilon, Y \in D\}.$$

定义 **2**. 给定最少样本数 $MinPts$ ($MinPts>0$), 对于任意的 $\boldsymbol{X} \in D$, 如果 $\boldsymbol{X}$ 的 ε 邻域包含了 $MinPts$ 个样本, 即

$$|N_\varepsilon(\boldsymbol{X})| \models MinPts,$$

则称 $\boldsymbol{X}$ 处于均匀密度区域; 如果包含了 $MinPts$ 个以上的样本, 即

$$|N_\varepsilon(\boldsymbol{X})| \rhd MinPts,$$

则称 $\boldsymbol{X}$ 处于高密度区域; 如果包含了 $MinPts$ 个以下的样本, 即

$$|N_\varepsilon(\boldsymbol{X})| \lhd MinPts,$$

则称 $\boldsymbol{X}$ 处于低密度区域

定义 2 为我们衡量样本的分布密度定义了一个标准, 这样我们就可以根据样本的分布密度, 对高密度区域的样本进行裁剪, 对低密度区域的样本进行补充, 使样本的分布变得比较均匀.

定义 **3**. 设样本集 $D=C_1 \cup C_2 \cup \cdots \cup C_m$, C_i ($i=1, \cdots, m$)代表一个样本类别, 且 $C_i \cap C_j=\varnothing$ ($i, j=1, \cdots, m$), 任意的 $\boldsymbol{X} \in C_i$, 如果 $\boldsymbol{X}$ 的 ε 邻域的所有样本都属于 C_i, 即

$$\forall \boldsymbol{Y} \in N_\varepsilon(\boldsymbol{X}), \text{都有 } \boldsymbol{Y} \in C_i,$$

则称 $\boldsymbol{X}$ 处于类内区域; 否则, 称 $\boldsymbol{X}$ 处于类边界区域

根据样本的分布密度, 对训练样本进行裁剪时, 对于处于类内区域和类边界区域的样本需要分开处理, 定义 3 使我们能够轻易地区分开这两种不同的样本.

定义 **4**. 给定邻域半径 ε 和最少样本数 $MinPts$, 对于样本集 D 中任意两个样本$\boldsymbol{X}$ 和$\boldsymbol{Y}$, 且 $\boldsymbol{Y}$ 位于$\boldsymbol{X}$ 的 ε 邻域内, 即 $\boldsymbol{Y} \in N_\varepsilon(\boldsymbol{X})$. 如果 $\boldsymbol{Y}$ 处于高密度区域, 即

$$|N_\varepsilon(\boldsymbol{Y})| \rhd MinPts,$$

则称样本 $\boldsymbol{Y}$ 从样本$\boldsymbol{X}$ 是高密度可达的, 记所有从样本 $\boldsymbol{X}$ 是高密度可达的样本为 $RH_{\varepsilon, MinPts}(\boldsymbol{X})$; 如果 $\boldsymbol{Y}$ 处于均匀密度区域, 即

$$|N_\varepsilon(\boldsymbol{Y})| \models MinPts,$$

则称样本 $\boldsymbol{Y}$ 从样本$\boldsymbol{X}$ 是均匀密度可达的, 记所有从样本 $\boldsymbol{X}$ 是均匀密度可达的样本为$RL_{\varepsilon, MinPts}(\boldsymbol{X})$; 如果 $\boldsymbol{Y}$ 处于低密度区域, 即

$$|N_\varepsilon(;\boldsymbol{Y})| \lhd MinPts,$$

则称样本 $\boldsymbol{Y}$ 从样本X 是低密度可达的, 记所有从样本 $\boldsymbol{X}$ 是低密度可达的样本为$RL_{\varepsilon, MinPts}(\boldsymbol{X})$.

对处于高密度区域样本 $\boldsymbol{X}$, 定义 4 为我们提供了一个裁剪依据.

3.2　样本裁剪方法

下面我们来看如何使用基于密度的样本裁剪方法, 解决第 2 节中所提出的两个问题

问题 1 的解决方法: 对于任意处于类边界区域的样本 $\boldsymbol{X}$, 如果 $\boldsymbol{X}$ 处于高密度区域, 则裁剪掉从样本 $\boldsymbol{X}$ 是高密度可达的样本, 直到 $\boldsymbol{X}$ 所处的区域变为均匀密度区域; 如果 $\boldsymbol{X}$ 处于均匀密度区域或低密度区域, 则保留样本 $\boldsymbol{X}$ 的 ε 邻域的所有样本.

从上面的方法可以看出, 对于位于边界且高密度区域的 $\boldsymbol{X}$, 我们尽量裁剪掉从样本 $\boldsymbol{X}$ 是高密度可达的样本, 这样不仅使 $\boldsymbol{X}$ 的密度变得较为均匀, 而且使 $\boldsymbol{X}$ 附近的样本也变得较为均匀. 如图 2 中, $MinPts=4$, $\boldsymbol{R}$ 是一个处于边界区域的、高密度区域的样本, 对 $\boldsymbol{R}$ 的 ε 邻域进行裁剪时, 样本 $\boldsymbol{Q}$ 所处的区域要比样本$\boldsymbol{P}$ 所处的区域密度高, 所以样本 $\boldsymbol{Q}$ 会被裁剪掉, 同理 $\boldsymbol{R}$ 的 ε 邻域还会有 3 个类 1 的样本会被裁剪掉, 从而使样本 $\boldsymbol{R}$ 所处区域的分布密度变得均匀, 避免了问题 1 的出现

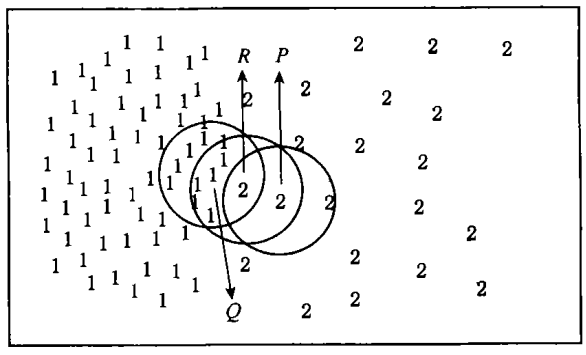

图 2　处于边界区域样本的裁剪示意图($MinPts=4$)

问题 2 的解决方法: 选取一整数 $LowPts$, 且 $0<LowPts \leqslant MinPts$, 对于任意处于类内区域的样本 $\boldsymbol{X}$, 如果$|N_\varepsilon(\boldsymbol{X})|>LowPts$, 则裁剪掉从样本 $\boldsymbol{X}$ 是

高密度可达的样本, 直到$|N_\varepsilon(\boldsymbol{X})|=LowPts$; 否则, 则保留样本 $\boldsymbol{X}$ 的 ε 邻域的所有样本

完全去掉处于类内区域的、且处于高密度区域的样本是不合适的, 所以在上面的方法中, 我们使用了一个小于 *MinPts* 的样本数 *LowPts* 来衡量处于类内区域的样本分布密度, 从而使裁剪后的、处于类内区域样本的分布密度低于处于类边界区域的样本的分布密度, 最终在不影响分类精度的情况下, 减少了分类的计算量.

对于问题 1 中的低密度样本和问题 2 中$|N_\varepsilon(\boldsymbol{X})|\leqslant LowPts$ 的训练样本, 我们都采取了保留其 ε 邻域所有样本的方法. 我们也可以先将这些样本记录下来, 然后在分类时通过隐式或显式的反馈, 给这些处于低密度区域样本的 ε 邻域内增加一些测试样本, 从而使这些样本的分布密度变得均匀起来

3.3 样本裁剪算法

输入: 训练样本集 D, 邻域半径 ε, 大于 0 的整数 *MinPts* 和 *LowPts*.

输出: 裁剪后的训练样本集 D'.

步骤:

$D'=\{\}$;

//保留处于低密度区域的训练样本

FOR EACH $\boldsymbol{X}\in D$ DO

BEGIN

IF X 处于类内区域 THEN

IF $|N_\varepsilon(\boldsymbol{X})|\leqslant LowPts$ THEN

$D'=D'+\{\boldsymbol{X}\}$;

ELSE

IF $|N_\varepsilon(\boldsymbol{X})|\leqslant MinPts$ THEN

$D'=D'+\{\boldsymbol{X}\}$;

END;

//裁剪处于高密度区域的训练样本

FOR EACH ($\boldsymbol{X}\in D$) And ($\boldsymbol{X}\notin D'$) DO

BEGIN

//$\boldsymbol{X}$ 的 ε 邻域内不能被裁剪掉的样本

$R(\boldsymbol{X})=\{\boldsymbol{Y}|\boldsymbol{Y}\in N_\varepsilon(\boldsymbol{X}),\boldsymbol{Y}\in D'\}$;

//$\boldsymbol{X}$ 的 ε 邻域内属于裁剪范围的样本

$R(\boldsymbol{X})=N_\varepsilon(\boldsymbol{X})-R(\boldsymbol{X})$;

//裁剪范围内从 X 高密度可达的样本

$H(\boldsymbol{X})=\{\boldsymbol{Y}|\boldsymbol{Y}\in R(\boldsymbol{X}),\boldsymbol{Y}\in RH_{\varepsilon,MinPts}(\boldsymbol{X})\}$;

IF $\boldsymbol{X}$ 处于类内区域 THEN

$Num=LowPts$;

ELSE

$Num=MinPts$;

WHILE $N_\varepsilon(\boldsymbol{X})>Num$ DO

BEGIN

$l=\arg\max_i |N_\varepsilon(t_i)|$, 其中 $t_i\in H(\boldsymbol{X})$;

$\boldsymbol{Y}=t_l$, 其中 $t_l\in H(\boldsymbol{X})$;

IF $N_\varepsilon(\boldsymbol{Y})>Hum$ THEN

$D=D-\{\boldsymbol{Y}\}$;

ELSE

Break;

END

$D'=D'+N_\varepsilon(\boldsymbol{X})$;

END;

算法讨论: 计算出所有训练样本的 k 个最近邻的时间复杂度是 $O(n^2)$, 裁剪过程的时间复杂度是 $O(2n)$. 相对于寻找所有训练样本的 k 个最近邻的时间, 裁剪过程的时间可以忽略不记, 那么整个算法时间复杂度为 $O(n^2)$.

3.4 参数估计

下面我们来讨论样本裁剪算法中的 *MinPts*、*LowPts* 和邻域半径 ε 这 3 个参数如何确定. 从前面的我们可以看到 *MinPts* 和邻域半径 ε 是相关的, 而 *LowPts* 和 *MinPts* 是相关的. 所以我们首先确定 *MinPts* 和邻域半径 ε, 然后根据确定后的 *MinPts* 来决定 *LowPts* 的值

(1) 确定 *MinPts* 和邻域半径 ε.

进行样本裁剪是为了使样本的分布密度变得均匀. 如何才算均匀呢? 显然, 如果某一区域的密度与整个样本集的平均密度相同或接近的话, 我们就可以认为这一区域的样本是分布均匀的. 那么, 我们就可以根据整个样本集的平均密度来确定 *MinPts* 和邻域半径 ε.

定义 **5**. 给定一个样本集 D, 我们定义样本集 D 在邻域半径为 ε 的情况下的平均样本数为

$$Density_\varepsilon(D)=\frac{1}{|D|}\sum_i^{|D|}|N_\varepsilon(\boldsymbol{X}_i)|,$$

其中 $\boldsymbol{X}_i\in D$.

根据定义 5, 给定邻域半径为 ε, 我们就可以将邻域半径为 ε 的情况下的平均样本数作为 *MinPts* 的值. 但在分类器设计时, 由于不同的特征加权方法、不同的特征数等一些因素, 会导致邻域半径 ε 的值差异很大, 所以很难选取一个合适的邻域半径 ε.

定义 **6**. 给定一个样本集 D, 设 $Dist_k(\boldsymbol{X})$代表样本集 D 中样本 $\boldsymbol{X}$ 的第 k 个最近邻到 $\boldsymbol{X}$ 的距离(或与 $\boldsymbol{X}$ 的相似度), 则我们定义样本集 D 在最少样本

数为 *MinPts* 的情况下的平均邻域半径为

$$Density_{MinPts}(D)=\frac{1}{|D|}\sum_{i}^{|D|}Dist_{MinPts}(\boldsymbol{X}_i),$$

其中 $\boldsymbol{X}_i \in D$.

根据定义 6, 给定最少样本数 *MinPts*, 我们就可以将最少样本数为 *MinPts* 的情况下的平均邻域半径作为 ε 的值. 而根据我们的实验, *MinPts* 的选值对分类器的影响很小, 取 *MinPts* 的值为类别的平均样本数的 5% ~8%都能取得很好的效果.

(2) 确定 *LowPts*.

通过实验, 我们发现在类别之间的界限比较明显、交叉情况不明显的情况下, 即处于边界区域的样本占训练样本总数的比例较小的情况下, 即使 *LowPts* 的取值很小也不会造成分类器的准确率和召回率的太大下降. 一般取 *LowPts* 为 0.7 ~0.8 倍的 *MinPts* 可以取得较好的效果.

4　实验结果

为了进一步考察算法的效果与效率, 我们用 VC++ 6.0 实现本文算法, 在 PIII 733, 128MB, Win2000 Professional 的环境下进行实验. 实验采用 N-gram 方法抽取文档属性向量, 使用信息增益(information gain)的方法来提取文档特征属性, 用 *k*NN 方法进行分类. 实验数据来自于政治、经济、文化、军事、体育、教育等 15 个领域的 12362 篇文档.

为了测试样本裁剪算法对分类器准确率和召回率的影响, 我们将 4685 篇文档作为训练样本, 将 7677 篇文档作为测试样本. 其中, 训练文档分为 *A*, *B* 两个文档集, 文档集 *A* 中的文档在各个类别的分布比较不均衡, 其中政治类的文档有 617 篇, 占到了训练文档集 *A* 的 25%, 而能源类的文档仅 63 篇, 才占到文档集的 2.5%; 文档集 *B* 中的文档在各个类别的分布情况则非常均匀, 每个类别中都有 150 篇文档. 这样可以使我们检验样本裁剪算法对文档在各个类别中分布均匀的情况和不均匀的情况是否都有效.

取 *MinPts*=10, 对文档集 *A* 取 ε=$Density_{MinPts}(A)$, 对文档集 *B* 取 ε=$Density_{MinPts}(B)$, 然后分别取 *LowPts* 的值为 1 ~10, 使用样本裁剪算法对其进行裁剪, 裁剪情况如表 1 所示. 表 1 中, "裁剪比例" = "裁剪数量" / "样本总数".

表 1　训练样本分布及裁剪情况表

LowPts	*A*		*B*	
	裁剪数量	裁剪比例/%	裁剪数量	裁剪比例/%
1	1329	54.6	1098	48.8
2	1084	44.5	926	41.2
3	935	38.4	815	36.2
4	821	33.7	711	31.6
5	750	30.8	635	28.2
6	682	28	577	25.6
7	640	26.3	534	23.7
8	595	24.4	485	21.6
9	551	22.6	450	20
10	510	20.9	425	18.9
样本总数	2435		2250	

从表 1 中我们可以看到, 即使 *LowPts*= *MinPts*, 文档集 *A* 和 *B* 的裁剪率也在 20%左右, 这就使 *k*NN 分类器的时间复杂度降低了 20%. 但是, 这并不是在降低分类准确率的代价下取得的. 图 3 ~图 10 为 *LowPts* 取 1 ~10 时, 使用未裁剪的样本和裁剪过的样本进行分类时的性能比较图.

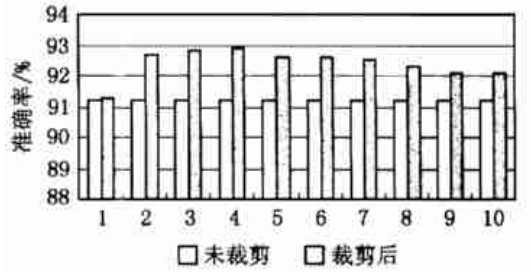

图 3　文档集 *A* 封闭测试准确率比较图

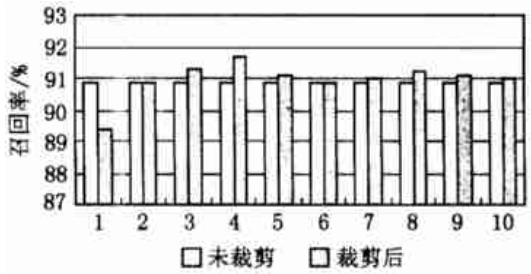

图 4　文档集 *A* 封闭测试召回率比较图

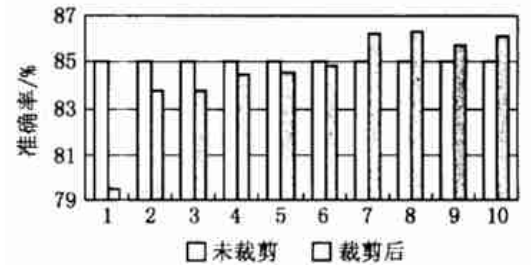

图 5　文档集 *A* 开放测试准确率比较图

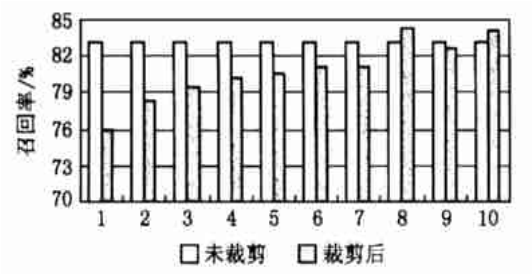

图 6 文档集 *A* 开放测试召回率比较图

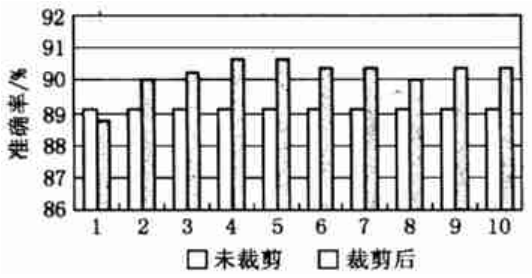

图 7 文档集 *B* 封闭测试准确率比较图

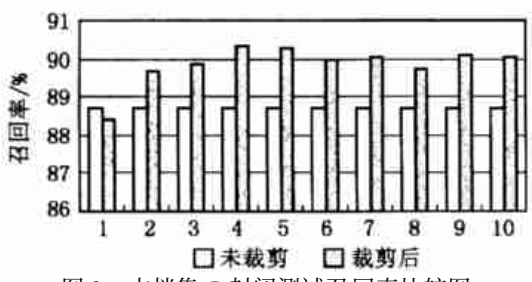

图 8 文档集 *B* 封闭测试召回率比较图

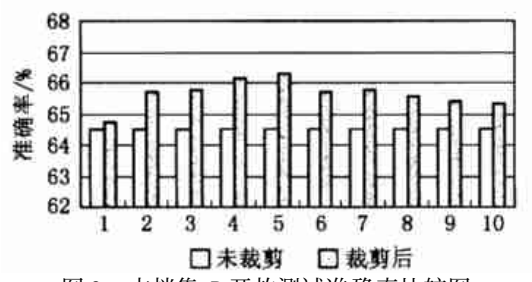

图 9 文档集 *B* 开放测试准确率比较图

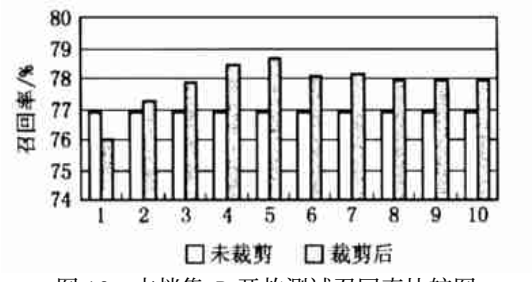

图 10 文档集 *B* 开放测试召回率比较图

从图 3 ~ 图 10 我们可以看到, 不管训练文档在各类别中分布是否均匀, 不管是在封闭测试还是开放测试中, *LowPts* 取 8 ~ 10 时, 使用裁剪后的训练样本进行分类, 取得了更高的分类准确率和召回率.

综上所述, 基于密度的样本裁剪方法不仅降低了 *k*NN 文本分类器的时间复杂度, 而且还能提高分类的准确率和召回率.

5 结 论

尽管基于密度的训练样本裁剪方法加快了 *k*NN 分类器的分类速度, 提高了分类的准确率, 但是仍有一些地方需要改进. 由于样本裁剪算法对处于低密度区域的样本采取了直接保留这些样本的方法, 那么这些样本所处的区域仍然是低密度区域, 这样训练样本中仍然会存在一些分布密度不均匀的区域. 今后, 我们在样本裁剪算法中将记录下这些处于低密度区域的样本, 然后采取反馈的方法, 补充一些样本到这些低密度区域, 从而使裁剪算法更加有效.

参 考 文 献

1 D D Lewis. Naive (Bayes) at forty: The independence assumption in information retrieval. In: The 10th European Conf on Machine Learning (ECML98), New York: Springer-Verlag, 1998. 4 ~ 15

2 Y Yang, X Lin. A re-examination of text categorization methods. In: The 22nd Annual Int'l ACM SIGIR Conf on Research and Development in Information Retrieval, New York: ACM Press, 1999

3 Y Yang, C G Chute. An example-based mapping method for text categorization and retrieval. ACM Trans on Information Systems, 1994, 12(3): 252 ~ 277

4 E Wiener. A neural network approach to topic spotting. The 4th Annual Symp on Document Analysis and Information Retrieval (SDAIR 95), Las Vegas, NV, 1995

5 R E Schapire, Y Singer. Improved boosting algorithms using confidence-rated predications. In: Proc of the 11th Annual Conf on Computational Learning Theory. Madison: ACM Press, 1998. 80 ~ 91

6 T Joachims. Text categorization with support vector machines: Learning with many relevant features. In: The 10th European Conf on Machine Learning (ECML-98). Berlin: Springer, 1998. 137 ~ 142

7 S O Belkasim, M Shridhar, M Ahmadi. Pattern classification using an efficient KNNR. Pattern Recognition Letter, 1992, 25(10): 1269 ~ 1273

8 V E Ruiz. An algorithm for finding nearest neighbors in (approximately) constant average time. Pattern Recognition Letter, 1986, 4(3): 145 ~ 147

9 P E Hart. The condensed nearest neighbor rule. IEEE Trans on Information Theory, 1968, IT-14(3): 515 ~ 516

10 D L Wilson. Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans on Systems, Man and Cybernetics, 1972, 2(3): 408~421
11 P Devijver, J Kittler. Pattern Recognition: A Statistical Approach. Englewood Cliffs: Prentice Hall, 1982
12 L I Kuncheva. Editing for the k-nearest neighbors rule by a genetic algorithms. Pattern Recognition Letters, 1995, 16(8): 809~814
13 L I Kuncheva. Fitness functions in editing *K*NN reference set by genetic algorithms. Pattern Rcognition, 1997, 30(6): 1041~1049

李荣陆　男, 1976 年生, 博士研究生, 主要研究方向为人工智能和自然语言处理

胡运发　男, 1940 年生, 教授, 博士生导师, 主要研究方向为数据工程与知识工程

第 13 届中国多媒体学术会议(NCMT2004)

征文通知

2004 年 10 月 14~17 日, 宁波

http://medialab.cs.tsinghua.edu.cn/~ncmt04

中国计算机学会

中国图像图形学会

由中国计算机学会多媒体专业委员会及中国图像图形学会多媒体专业委员会联合召开的第 13 届全国多媒体技术学术会议定于 2004 年 10 月 14~17 日在风景秀丽的浙江宁波市召开. 会议由宁波大学承办, 会议期间将组织著名学者就宽带和无线网络、新型计算模式、流媒体、数字版权管理等热点领域做大会特邀报告和广泛的学术讨论

欢迎各位同行踊跃投稿, 这次大会的内容将覆盖以下广泛的领域, 但并不局限于这些内容.

- 多媒体信息处理和编码: 多媒体信息处理和压缩、嵌入式多媒体处理、内容分析、基于内容的检索、数字版权管理(DRM)和信息安全
- 多媒体系统支持和网络技术: 网络协议、无线网络、操作系统、中间件、流媒体服务器、多媒体服务质量保证(QoS)、数据库、传感器和执行元件、客户终端、流媒体技术
- 多媒体工具、应用系统: 超媒体系统、用户接口、著作工具、多媒体教育系统、分布式多媒体系统和应用、虚拟空间的设计和应用、系统集成
- 计算机图形、虚拟现实、多媒体人机交互、多媒体与 CSCW

关于投稿的重要日期

投稿截至日期(以邮件寄出日期): 2004 年 7 月 20 日

录取通知: 2004 年 8 月 4 日以前

正式稿件发回: 2004 年 8 月 20 日以前

NCMT2004 上的优秀论文将推荐到国内著名学术刊物发表, 其中包括《电子学报》、《系统仿真学报》等

程序委员会主席: 钟玉琢 教授, 清华大学计算机系

组织委员会主席: 蒋刚毅 教授, 宁波大学信息学院

征文要求

① 反映在多媒体及相关技术领域中的技术和应用研究成果;
② 未在其它会议或刊物上公开发表;
③ 每篇来稿篇幅不超过 6 页, 按 A4 纸排版, 论文格式参见会议主页;
④ 每篇论文务请附上作者联系信息(电话、通信地址、电子邮件);
⑤ 来稿请寄: 北京清华大学计算机系人机交互与媒体集成研究所　孙立峰收, 邮编: 100084
同时将论文电子版以 word 或 pdf 格式用 E-mail 方式发至 ncmt04@media.cs.tsinghua.edu.cn

联系方式

联系人: 孙立峰, 田淑珍　清华大学计算机系

电话: 010-62786910, 010-62784141　　传真: 010-62771138

电子邮件: ncmt04@media.cs.tsinghua.edu.cn

有关会议的更详细的信息请访问 http://medialab.cs.tsinghua.edu.cn/~ncmt04