通过确定邻近区域改进 KNN 文本分类①

Improving KNN for Text Classification by Adjacent Domain Determination

汪成亮 [1,2]　张硕果 [1]

(1.重庆大学 计算机学院 重庆 400030；2.重庆大学 电气工程学院 重庆 400030)

摘　要：文本分类技术已经成为处理和组织文本信息的关键技术之一。KNN 算法是文本分类中一种实用的方法。它在每次分类的过程中都要计算测试集中未标记文本与训练集合中所有样本的相似度(距离)，然后通过排序来找到 K 个最近邻样本，耗时较长，不利于 Web 上实时在线分类等应用。提出了一种确定邻近区域来加快搜寻 K 个最近邻的方法。试验证明，改进后的 KNN 算法较经典 KNN 算法在分类过程中速度有所提升，并且当训练文本数量增加时，在分类时间上表现相对更稳定。

关键词：文本分类 k-最近邻 邻近区域 相似度 kNN 算法

1　引言

文本分类是指根据文本内容将文本归入预先定义的类别。随着可用电子文档的增长和在线信息的快速膨胀，文本分类技术已经成为处理和组织文本信息的关键技术之一[1]。

KNN 文本分类算法首先确定 k 值，然后依据文本相似度找出 k 个最相似的训练文本，把测试文本指派给其中相似样本最多的一类[2]。

经典 KNN 算法在每次分类的过程中都要计算待分类文本与样本集中所有样本的相似度(距离)，然后排序找出 K 个最近邻，速度较慢，并且当训练集中文档样本的数量增加时，计算量也随之上升，这不太适合于在线 Web 文档分类等技术的应用[3]。

本文针对此提出通过确定测试文档的邻近区域来加速 K 个最近邻查找过程的方法。试验证明改进后的 KNN 算法在速率上有所提升，并且当测试文档数量增加时分类时间能够保持相对的稳定。

2　中文文本分类方法

2.1 文本预处理

文本预处理包括去除停用词等。由于中文文本词与词间没有天然间隔，还需要进行分词处理。

2.2 文本的表示

常用的文本表示方法是向量空间模型法(VSM)，即将文本表示为特征词组成的特征向量，这里称为文本的特征词向量。本文中，特征权重的计算采用 TF-IDF 方法[4]：

$$w_k=\frac{tf(t_k,D_i)\times\log\left(N/n_{t_k}+0.01\right)}{\sqrt{\sum\left[tf(t_k,D_i)\times\log\left(N/n_{t_k}+0.01\right)\right]^2}} \quad (1)$$

其中，tf (t_k，D_i)为特征词 t_k 在文本 D_i 中的频数，N 为文本的总数，n_t 为训练文本集中出现 t_k 的文本数。

2.3 特征选择

由于向量空间的高维性会导致分类时运算量过大，因此需要有效的特征选择方法来进行特征选择以降低向量空间的维度。常用的方法有信息增益、CHI 统计、互信息和期望交叉熵等。

本文采用信息增益方法进行特征项提取，公式[5]如下：

$$\begin{aligned}IG(t)=&-\sum_{i=1}^{m}P(C_i)\log P(C_i)\\&+P(t)\sum_{i=1}^{m}P(C_i\mid t)\log P(C_i\mid t)\\&+P(\bar{t})\sum_{i=1}^{m}P(C_i\mid \bar{t})\log P(C_i\mid \bar{t})\end{aligned} \quad (2)$$

① 基金项目:重庆市自然科学基金(CSTC)(2007BB6118)
收稿时间:2009-03-05

其中，P(C_i)表示 C_i 文档在语料中出现的概率；P(t) 表示语料中包含词条 t 的文档的概率；P(C_i | t)表示文档包含词条 t 且属于 C_i 类的条件概率；$P(\bar{t})$表示语料中不包含词条 t 的文档的概率；P(C_i | $\bar{t}$) 表示文档不包含词条 t 时属于 C_i 的条件概率；m 表示类别数。

2.4 KNN

对于一个测试文本，计算它与训练样本集中每个文本的相似度，找出 k 个最相近邻文档样本，根据各类别的加权距离和判断测试文本所属类别。具体算法步骤如下：

①计算测试文本与训练集中每个文本的文本相似度，公式为：

$$Sim(d_i,d_j)=\frac{\sum_{k=1}^{n} w_{ik}\times w_{jk}}{\sqrt{\sum_{k=1}^{n} w_{ik}^2 \sum_{k=1}^{n} w_{jk}^2}} \quad (3)$$

其中，w_{ik} 是特征权值，n 是文本 d 特征向量的维度。

②按照文本相似度由大到小排序，选出与测试文本最相近邻的 k 个文档样例。

③在测试文本的 k 个近邻中，依次计算每类的权重，公式如下[6]：

$$\mu_j(X)=\sum_{i=1}^{k}\mu_j(X_i)sim(X,X_i) \quad (4)$$

其中，μ j(Xi)∈(0,1)表示文档 X_i 是否属于 C_j；sim(X,X_i)表示测试文档和训练文档的相似度。则决策为:如果 $\mu_j(X_i)=\max \mu_i(X)$，则决策 X∈C_j。即类别权重最大的便是测试文档的所属类别。

3 改进的KNN算法

3.1 类中心向量

类中心向量是类别 k 中所有训练文本向量的算数平均[7]，其表示为 C_k =< C_{k1} , C_{k2} ，…，C_{km} >其中: $C_k=\frac{1}{S_k}\sum_{k=1}^{S_k} W_{kij}$，m 为特征总数，Sk 为第 k 类训练样本总数，Wkij 为第 k 类的第 i 个样本中第 j 个特征项的特征权重。

3.2 确定邻近区域

定义 1. 文档样本 s 到各类别中心点的距离构成一个向量，称为文本的距离向量，表示为 D=<D_1，D_2，……，D_j，……，D_m>，其中：

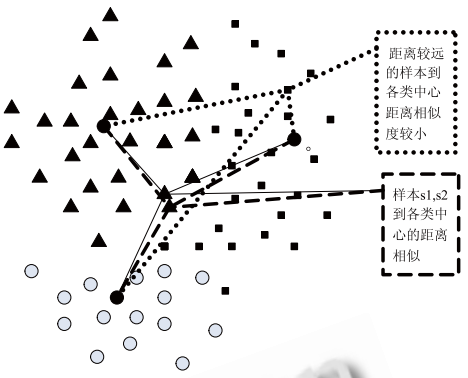

图 1　相邻样本与每个类别中心距离相近

$$D_j=sim(d_i,C_j)\Big/\sum_{j=1}^{m} sim(d_i,C_j) \quad (5)$$

m 表示文档类别总数，sim(d_i,C_j)用公式(3)进行计算。

定义 2. 文档样本 s1，s2 的距离向量相似程度称为距离相似度。

距离相近的两个文档样本，各自对应的距离相似度较高，而距离较远的两个文档样本，其距离相似度较低(见图 1)。

计算出测试集文档的距离向量<D_1，D_2，……，D_m>。测试集文档的 n 个近邻，便是与它距离相似度最高的 n 个训练文档样本。它们处于邻近区域 U(<[D_1- α，D_1+ α]，[D_2- α，D_2+ α]，……，[D_m- α，D_m+ α]>)内，满足：

$$\begin{aligned} &D_1-\alpha \le TD_1 \le D_1+\alpha \\ &D_2-\alpha \le TD_2 \le D_2+\alpha \\ &\quad\vdots \\ &D_m-\alpha \le TD_m \le D_m+\alpha \end{aligned} \quad (6)$$

其中：<TD_1，TD_2，……TD_m>是训练文档的距离向量。

设 K 值为 KNN 算法中要求的最近邻个数。

① 当 K<=n 时，邻近区域便确定下来；

② 当 K>n 时，增大 α 的取值，使区域 U 扩展到 U ’ (见图 2)，直到满足条件①为止。

最后使用如 2.4 节所示的 KNN 算法，计算邻近区域中测试集文档(注意：这里的计算不是用距离向量而是文档的特征词向量)与训练集文档的相似度，并排序，来搜寻 K 个近邻并作出分类决策。

3.3 改进的 KNN 算法描述与分析

3.3.1 算法描述

首先计算出各类别的中心，以及训练集文档的距离向量。查找 K 近邻过程如下：

输入：类中心，训练文档集 T，测试文档 d

输出：该测试文档的 K 个最近邻

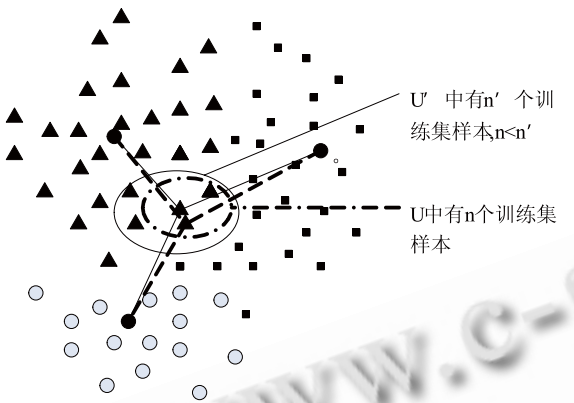

图 2　k 值不满足条件，U 扩大到 U'

Step1　λ =1，设定 K 值，α值，集合 S=∅，N=0

Step2　用公式(5)计算出测试文档的距离向量 <D_1，D_2，……，D_j，……，D_m>，确定邻近区域 U<[$D_1-\lambda\alpha_1$, $D_1+\lambda\alpha$],[$D_2-\lambda\alpha$, $D_2+\lambda\alpha$],... ...,[$D_m-\lambda\alpha$, $D_m+\lambda\alpha$]>

Step3　找出集合 T 中距离向量满足(6)式的训练样本，构成集合 t。

Step4　统计 t 中的样本个数 n，S=S∪t，N=N+n，T=T-t

Step5　如果 K>N，$\lambda=\lambda+1$，返回 Step3

否则，用 2.4 节描述的 KNN 方法在集合 S 中找出 K 个最近邻，作出分类决策。

3.3.2 算法分析

在 T 中寻找满足(6)式的样本时，先从类中心与测试样本 d 距离最近的类别开始，即由 Dj 值最大的类的文档集合，从近到远依次进行查找(这是因为 K 个最邻近样本有很大可能位于与 d 距离最小的前几个类中心所代表的类别中)，当满足 K≤N 时即可结束，无需扫描整个文档集合，缩短构造集合 S 的时间，提升整体速率。

算法最终得出的集合为 S，其中的文档数为 N≥K，假如是前 λ -1 次查找后得到的集合，其中文档数为 N'。因为 $N'<K$，可知 S' 是测试文档 K 个最近邻文档集合的子集，所以只需要对 $S-S'$ 集合中的 $N-N'$ 个文档进行排序，找出前 $K-N'$ 个文本即可，这使得改进后需要排序的文档数量远小于改进前。

4　实验与分析

4.1 实验设置

感谢中科院提供的 ICTCLAS 免费中文分词系统。

中文文档集合来自于——复旦大学语料库，搜狗实验室和中文自然语言处理开放平台。从中选取 15 个类别，5247 篇训练文档，1726 篇测试文档，进行分类测试。

试验将 k 值设定为 25。文档总数保持不变，先后将类别数设为 5、10、15，进行三组试验，测试类别数目对算法的影响。然后将类别数设定为 15，分步增加训练文档的数量，测试文档数目对算法的影响。

4.2 试验结果与分析

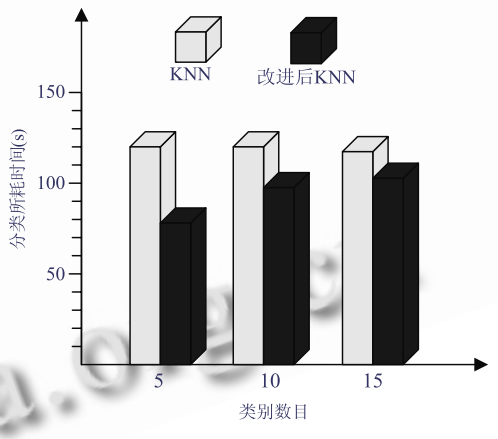

图 3　文档总数不变时改进前后分类时间

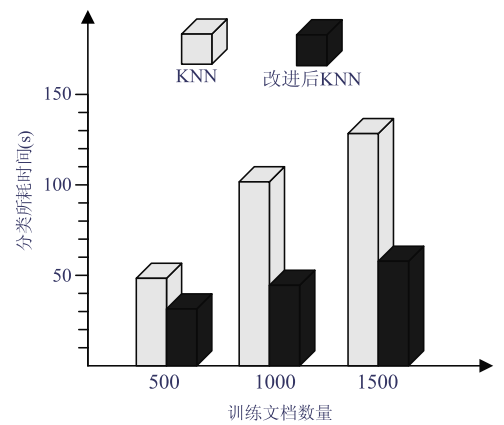

图 4　文档类别数目维持不变时改进前后分类时间

从图 3 中可以看出，当文档总数维持不变，类别增加时，经典 KNN 算法的时间保持不变，而改进后的 KNN 算法随着类别的增加，分类时间有所上升，这主要是因为随着类别的增多，测试文档与类别中心距离的计算次数增加，而距离向量的维数增多也增加了确定邻近区域所耗费的时间。

从图 4 可以看出，当文档类别数目维持不变。训练文档数量增加时，经典 KNN 算法分类时间随之增加，而改进后的 KNN 算法时间增长较缓，相对较为稳定。其原因第一可能是由于 K 近邻的查找过程在(与测试集文档 d)距离最近的前几个类中心所代表的类中即可完成，而无需扫描整个文档集合；第二，训练文档的增多会增加经典 KNN 算法的排序时间，而改进后的 KNN 算法需要排序的文档数量远小于前者。因此训练文档的增加对其影响不是很明显。

在以上两个实验中，改进后的 KNN 算法与经典 KNN 算法得出的 K 个最近邻是相同的，所以分类的精度也相同。改进后的 KNN 分类速度有所提升很重要的原因在于不用对整个文档集合中的文档进行排序。

5 结论

本文提出了一种加快 KNN 运算过程的方法。改进后的 KNN 算法与经典 KNN 算法相比多了一个计算文档距离向量的步骤，相当于增加了训练时间，而在分类中的速率高于经典 KNN 算法，并且当类别一定，训练文档数量增加时相对于经典 KNN 算法也更加稳定。下一步将在此基础上进行优化，加快确定邻近区域的过程。

参考文献

1 徐燕,李锦涛,王斌,孙春明.基于区分类别能力的高性能特征选择方法.软件学报, 2008,19(1):82 - 89.
2 张晓辉,李莹,王华勇,赵宏.应用特征聚合进行中文文本分类的改进 KNN 算法.东北大学学报(自然科学版), 2003,24(3):229 - 232.
3 牛强,王志晓,陈岱,夏士雄.基于 KNN 的 Web 文本分类方法的研究.计算机应用与软件, 2007,24(10):210 - 211.
4 罗欣,夏德麟,晏蒲柳.基于词频差异的特征选取及改进的 TF-IDF 公式.计算机应用, 2005,25(9):2031 - 2033.
5 刘健,张维明.基于互信息的文本特征选择方法研究与改进.计算机工程与应用, 2008,44(10):135 - 137.
6 吕震宇,赵爽,林永民.KNN 在中文文本分类中的应用研究.计算机与现代化, 2008,(11):69 - 72.
7 陈瑞芬.一种结合反馈方法的中文文本分类算法.计算机应用, 2005,25(12):2862 - 2863.