

doi:10.3772/j.issn.1000-0135.2009.05.010

文本粗分类数据中噪声的快速修正算法¹⁾

宣照国 党延忠

(大连理工大学系统工程研究所, 大连 116023)

摘要 训练数据中的噪声数据对文本分类结果的精度会造成不良影响,本文提出了一种对噪声数据进行修正的快速算法。针对以前的算法,每次迭代只对一个文档进行修正,迭代次数与噪声数据数量相当,算法运行效率较低的问题,本文通过分析调整文档所属类别对评价指标的影响,提出依据模块度变化量判断噪声数据,一次迭代过程中可以对多个文档进行修正处理,从而提高算法效率。实验结果表明,本文所提算法能够更快地修正粗分类数据中的噪声,算法复杂度从以前算法的 $O(Tnm^2)$ 降低为 $O(Tnm)$ 。该算法可以用于对大数据量数据进行处理,实用价值更高。

关键词 文本分类 集团结构 模块度优化 噪声数据

A Fast Noise Revision Algorithm in Text Categorization

Xuan Zhaoguo and Dang Yanzhong

(Institute of Systems Engineering, Dalian University of Technology, Dalian 116023)

Abstract The noisy texts in training data will influence the performance of the categorization system. This paper proposes a fast revision algorithm for revising the noisy texts. The previous algorithm, NNRA, revise only one document at one iteration step. The iteration times equal to the number of noisy texts in document set approximately. This paper aims to improve the efficiency of the algorithm, especially the convergent speed. By analyzing the influence of the modularity changing the categorization of the documents, we put forward an improved algorithm that using the changes of modularity to identify the noisy texts. The improved algorithm can revise several noisy texts at one iteration step. The experimental results indicate that the proposed algorithm can increase the convergent speed obviously. The computing complexity is $O(Tnm)$ instead of $O(Tnm^2)$. This algorithm can be used in the applications with large amount of documents.

Keywords text categorization, community structure, modularity optimization, noisy text

1 引言

文本分类是文本处理的一项基本工作,提高文本分类精度已成为一项重要而迫切的研究课题。文本分类技术可以分析被分类文档的特征,并与各类别中文档所具有的共同特征进行比较,将被分类文

档归为特征最接近的一类中并赋予相应的类别。文本分类是有指导的自动处理技术,分两个阶段进行。首先通过带有类别标签的训练文档集来构造分类器;然后使用构造出来的分类器对新文档进行分类。第一阶段中构造的分类器的质量,对第二阶段的分类精度具有直接的影响。但在实际应用中,无法预先提供类别准确的、全面的精分类训练文档数据,往

收稿日期: 2008年4月30日

作者简介: 宣照国,男,1971年生,博士,主要研究方向:中文信息处理、文本知识发现。E-mail: xzg@dl.cn。党延忠,男,1954年生,教授,博士生导师,主要研究方向:知识科学、知识管理、系统开发与集成。

1) 国家自然科学基金资助项目(70431001,70620140115,70771019)。

往直接使用一些粗分类数据作为训练文档,比如按照预先定义的栏目类别整理的网页数据、期刊上划分为各个栏目的论文、以及科研资助机构中标记着学科代码的立项建议书等。这些数据中具备了类别特征,即每个类别下的数据具有一定的共性特征,但是各个类别下的成员文档集合中,经常会存在一定数量的类别标记错误的文档,即文档内容与标记的类别不符。本文把这种类别标记错误的文档称为噪声数据。文献[1]的实验结果表明,噪声数据会对分类结果产生不良影响。

A. Vinciarelli 分析了对含噪声数据的分类问题,分析结果同样显示了噪声数据会对分类结果产生不良影响,因此他提出可以通过降低召回率来保证分类结果准确率的策略^[2]。这种方法实际上只得到了待分类数据中部分数据的分类结果,并没有提出剩余的待分类数据的处理方法。文献[1]提出了一种针对粗分类文档中噪声数据的修正算法(记为:NNRA),把类别标记错误的文档放回正确的类别中,获得精分类的训练数据,从而提高分类器的性能。其基本思路是:通过计算文档之间的相关度,构建文档关联网络模型,进而借鉴复杂网络理论中社区集团发现方法及其结果评价算法,对类别划分质量进行评价,并通过优化类别划分质量来修正文档集合中的噪声数据,从而获得分类准确的训练数据。NNRA 算法根据对文档类别进行调整之后的模块度选择数据进行修正,并且每个时间步只能对一篇文档进行修正,效率不高,其计算复杂度为 $O(Tnm^2)$,其中 T 为迭代次数, n 为文本数量, m 为类别数量。本文提出一种改进算法,通过分析调整文档的所属类别对模块度的影响,依据调整文档类别后模块度变化量识别和修正噪声数据,较大程度降低了算法的计算复杂度,提高算法效率。

2 文档关联网络及 NNRA 算法

2.1 网络的构建及集团结构

根据传统的向量空间模型(Vector Space Model, VSM)^[3],文档的内容被表示为特征空间中的加权特征向量 $d_i = (tf - idf(t_1), tf - idf(t_2), \dots; tf - idf(t_n))$,其中, t_k 表示特征词条, $tf - idf(t_k)$ 表示使用 TFIDF 算法计算的词条 t_k 在文档 d_i 的特征向量中权值。把文档表示为特征向量之后,使用夹角余弦公式来计算两篇文档之间的相似性。对于文档 d_i 和 d_j ,它们之间的相似度 $s_{ij} = \cos(d_i, d_j)$ 。

由此构建出文档关联网络(Document-Relation Network, DRN)。在 DRN 中共包含了 n 个节点(d_1, d_2, \dots, d_n),每个节点代表一个文档,节点之间的边代表两个文档是相似的,并把文档之间的相似度 s_{ij} 作为边的权。DRN 可以用文档关联矩阵 $S = [s_{ij}]_{nn}$ 来描述。

按照文档的类别,DRN 上的 n 个文档分为 m 个集团(c_1, c_2, \dots, c_m),两个集团之间的所有边上的权值之和反映了两个集团(即两个类)之间的关联状况。以集团作为节点,集团之间的关联作为边,关联强度作为边的权值,这样就构成一个新的网络“集团网络”(Group-Relation Network, GRN)。GRN 可以用集团关联矩阵 $E = [e_{ik}]_{mm}$ 来描述。DRN 是基础网络,GRN 是建立在 DRN 上的集团网络,根据文档关联矩阵 S 定义集团矩阵 E ,其元素 e_{ik} 表示集团 c_i 和 c_j 之间的关联强度, e_{ik} 定义如下:

$$e_{ik} = e_{kl} = \frac{1}{2 * tot} \sum_{d_l \in c_j, d_i \in c_k} s_{ij} \quad (1)$$

其中, tot 为网络 DRN 中所有边的权值总和,即: $tot = \sum_{i,j} s_{ij}$ 。矩阵 E 中对角线上的元素 e_{kk} 为集团 c_k 内部所有边的权值之和在网络中所占的比例。根据矩阵 E 的构建过程可以得出, $\sum_{i,k} e_{ik} = 1$ 。

模块度(Modularity,记为 Q)^[4]被用来评价拓补网络中集团结构划分的质量。在加权网络中,模块度 Q^w 的计算方法如下:

$$Q^w = \sum_{l=1}^m (e_{ll} - a_l^2) \quad (2)$$

其中, a_l 为矩阵 E 中第 l 行元素的和,即: $a_l = \sum_{k=1}^m e_{lk}$ 。

模块度(Q^w)这一指标综合评估了集团内和集团间节点的连接强度,从网络结构角度衡量了集团结构划分的质量。如果权值大的边连接的节点被划分到同一类别中,那么模块度就大;反之,如果权值大的边连接的节点被划分到不同类别中,模块度就小。文献[1]考察了噪声数据对模块度的影响,通过计算不同比例噪声数据的模块度发现,模块度随着噪声数据含量的增加而下降,反之模块度 Q^w 越大,说明噪声数据的含量越少。由此,文献[1]构造了 NNRA 算法,把模块度作为优化指标使其不断增大,从而减少粗分类文档中的噪声数据。

2.2 NNRA 算法

文献[1]提出的 NNRA 算法的基本思想是调整

网络的划分方案,使模块度指标最大化。NNRA 通过把文档从其所在的当前类别临时改成其他的类别来检验该文档是否为噪声数据。并定义了文档 d_i 对于类别 c_j 的隶属度为 d_i 与 c_j 的所有成员之间边的权值在网络中所占的比例,记为 D_{ij} ,即:

$$D_{ij} = \frac{1}{\text{tot}} \sum_{d_k \in c_j} s_{ik} \quad (3)$$

由此构建了文档-集团隶属矩阵 $A = (D_{ij})_{mn}$, m, n 分别表示集团(类)的个数和文档个数。

把文档 d_i 从标记类别 x 调整到测试类别 y 中,调整后集团关联矩阵 E 的元素改变为:

$$\begin{aligned} e'_{xx} &= e_{xx} - D_{ix}, e'_{yy} = e_{yy} + D_{iy}, e'_{xy} = e'_{yx} \\ &= e_{xy} + (D_{ix} - D_{iy})/2 \end{aligned} \quad (4)$$

而对于其他类别 $z (z \neq x, z \neq y)$,有:

$$e'_{xz} = e'_{zx} = e_{xz} - D_{ix}/2, e'_{yz} = e'_{zy} = e_{yz} + D_{iy}/2 \quad (5)$$

$$e'_{zz} = e_{zz} \quad (6)$$

进而使用公式(2)计算出调整后的模块度,如果模块度增大,说明调整是合理的,反之则不调整。

NNRA 算法每个时间步对所有文档进行考察时,计算各个文档调整后的模块度,选择一个调整后模块度最大的文档并对其进行修正。实验结果表明该算法的收敛速度与噪声数量基本一致。算法的计算复杂度为 $O(Tnm^2)$ 。

3 改进的噪声数据修正的改进算法

对文档进行考察时,只要判断调整后模块度的变化量,如果模块度变化量为正,说明调整后模块度增大,如果变化量为负,说明调整后模块度会减小。模块度变化量越大,调整后的模块度越大。所以,本文在文献[1]的基础上,进一步分析调整文档类别对模块度的影响。

根据公式(4)和公式(5),可得出调整后的 a_i 的值:

$$\begin{aligned} a'_x &= a_x - D_{ix} + (D_{ix} - D_{iy})/2 - \text{sum}(D_{ix})/2 \\ &= a_x - (D_{ix})/2 \end{aligned} \quad (7)$$

$$\begin{aligned} a'_y &= a_y + D_{iy} + (D_{ix} - D_{iy})/2 + \text{sum}(D_{ix})/2 \\ &= a_y + (D_{iy})/2 \end{aligned} \quad (8)$$

$$a'_z = a_z \quad (9)$$

其中, D_i 是文档 d_i 对于所有类别的隶属度的和,即: $D_i = \sum_{j=1}^m D_{ij}$ 。对于特定的文档 d_i , D_i 是一

个恒定量,不随着 d_i 所属类别的变化而变化。

根据公式(2)、公式(4)~(5)、公式(7)~(8),可以计算出文档 d_i 从类别 x 调整到类别 y 后的模块度的变化量:

$$\begin{aligned} \Delta Q^w &= \sum_{i=1}^m [(e'_{ii} - a'^2_i) - (e_{ii} - a^2_i)] \\ &= (D_{iy} - D_{ix}) - [D_i(a_y - a_x) + \frac{D_i^2}{2}] \end{aligned} \quad (10)$$

通过公式(10)可以看出,模块度的变化量与文档的连接属性 D_i 、 D_{ix} 、 D_{iy} 相关,同时也和变量 a_x 、 a_y 相关。这说明模块度变化量只与文档的标记类别和测试类别相关,而与其他类别无关,即把文档 d_i 从类别 x 调整到类别 y 中,只和类别 x 和 y 相关。假如同时还对文档 d_j 从类别 p 调整到类别 q 进行了调整测试,如果 $\{x, y\} \cap \{p, q\} = \emptyset$,则对文档 d_i 的调整,不会影响到对 d_j 的测试结果 $\Delta Q^w(d_j: p \rightarrow q)$ 。因此,对所有文档进行一次调整测试的结果,可以用于对多个文档进行类别调整。

本文与文献[1]一样,根据文档对于类别的隶属度来判断是否需要对其修正。NNRA 计算调整后的模块度,而本文是根据公式(10)计算调整后模块度变化量。设 $c_i^* \uparrow$ 是文档 d_i 的隶属度最大的类别,即 $c_i^* = \arg(\max_j D_{ij})$,如果 c_i^* 就是 d_i 当前的所属类别,则不对其进行调整。否则,将 d_i 作为需调整的候选文档,并通过公式(10)计算把 d_i 从当前类别 x 调整到类别 c_i^* 后的模块度变化量 $\Delta Q^w(i)$ 。如果变化量 $\Delta Q^w(i) > 0$,则把 $d_i: x \rightarrow c_i^*$ 作为一个候选方案保存到集合 S 中。每个时间步对所有文档进行调整测试,所有使模块度增大的调整方案都保存到集合 S 中。

实施调整时,首先从集合 S 中选择一个模块度变化量最大的调整方案,进行调整处理,然后从集合 S 中删除与本次调整涉及的类别有关的候选方案。再从集合 S 的剩余方案中选择模块度变化量最大的调整方案,进行调整处理。如此反复,直到集合 S 中没有候选的调整方案。

为防止对模块度增加较小的调整方案获得优先处理,把集合 S 的所有候选方案中使模块度增大的最大增量 ΔQ^{\max} 作为基准增量,设置参数 α , $\alpha \in (0, 1]$ 。在每一轮调整中,只有使模块度增量不小于 $\alpha * \Delta Q^{\max}$ 的调整方案才作为有效方案,改变 α 值可以在收敛速度和调整精度之间进行调整。 α 增大,算

法优先处理模块度增量较大的调整方案,结果的精度提高,当 $\alpha = 1$ 时,算法成为每次调整都寻求模块度增量最大的贪婪算法;当 $\alpha = 0$ 时,每次都尽量寻求调整更多的文档,收敛速度加快。

算法流程如下:

输入: DRN 的节点集合 $D = \{d_i\}$, 边权集合 $\{s_{ij}\}$, 各个节点的原始类别 $\{Tag_i\}$;

输出: 各个节点的调整后的类别 $\{T_i\}$;

(1) 初始化: 建立矩阵 E 和矩阵 A ; 计算初始模块度 Q^w ; 设置 $T_i = Tag_i$; 设置参数 α ;

(2) 初始化: 测试结果集合 $S = \emptyset$;

(3) DO WHILE (true)

(4) $S := \text{getCandidate}()$; //调整测试, 所有候选调整方案保存在集合 S 中;

(5) IF $S = \emptyset$ THEN 算法终止, 输出修正结果 $\{T_i\}$;

(6) $\{T_i\} := \text{setCategory}(S)$; //根据集合 S 中的调整方案实施调整;

(7) END WHILE

算法每一个时间步对每一个文档 d_i 进行调整测试。对于文档 d_i , 找出具有最大隶属度的类别, 判断把 d_i 从当前所属类别调整到最大隶属度类别后的模块度变化量 ΔQ^w , 如果 $\Delta Q^w > 0$, 则把对 d_i 的调整作为一个候选方案, 保存在集合 S 中。所有文档测试完毕后, 输出候选方案集合 S 。调整测试的算法流程如下:

FUNCTION $\text{getCandidate}()$

(1) FOR EACH $d_i \in D$ DO

(2) $c_i^* := \arg(\max_{j=1 \dots m} D_{ij})$; //获取 d_i 的隶属度最大的类别

(3) IF $c_i^* \neq T_i$ THEN

(4) 使用公式(10)计算 $\Delta Q^w(d_i: T_i \rightarrow c_i^*)$;

(5) IF $\Delta Q^w(d_i: T_i \rightarrow c_i^*) > 0$ THEN

(6) $S = S \cup (\Delta Q^w(d_i: T_i \rightarrow c_i^*))$

(7) END IF

(8) END FOR

(9) RETURN S ;

END FUNCTION

根据候选调整方案集合 S , 对文档类别进行调整。每次循环从集合 S 中选择一个使模块度增量最大的候选方案, 如果这个方案的模块度增量超过设

定的阈值, 则进行修正处理。本次修正处理涉及了两个类别, 即文档原来标记的类别和修正后的新类别。由于集合 S 中与这两个类别相关的候选方案的模块度增量已经不再正确, 因此要从集合 S 中把这些受到影响的候选方案删除。如此循环处理, 直到集合 S 为空, 没有候选调整方案。实施调整的算法流程为:

FUNCTION $\text{setCategory}(\text{Set } S)$

(1) $\Delta Q^{\max} := (\text{集合 } S \text{ 中所有调整方案中, 产生的模块度最大增量})$;

(2) DO WHILE $S \neq \emptyset$

(3) $t := (\text{集合 } S \text{ 中对文档 } t \text{ 调整后模块度增量最大})$

(4) IF $\Delta Q^w(d_i: T_i \rightarrow c_i^*) \leq \alpha * \Delta Q^{\max}$ THEN

(5) $T_i := c_i^*$;

(6) 使用公式(3)更新矩阵 A ;

(7) 使用公式(4)和(5)更新矩阵 E ;

(8) 删除 S 中与类别 T_i, c_i^* 相关的调整方案;

(9) ELSE

(10) $S := \emptyset$;

(11) END IF

(12) END WHILE

(13) RETURN $\{T_i\}$;

END FUNCTION

算法复杂度分析: 算法每个时间步分别进行调整测试和实施调整两个步骤。在步骤调整测试中, 要对所有文档进行测试。当对文档 d_i 进行调整测试时, 首先在矩阵 A 的第 i 列查找 d_i 的隶属度最大的类别 c_i^* , 复杂度为 $O(m)$; 然后使用公式(10)计算把 d_i 从当前类别 T_i 调整到类别 c_i^* 的模块度增量, 计算复杂度为 $O(1)$ 。每次调整要测试所有文档, 复杂度为 $O(nm)$ 。根据集合 S 中的候选方案实施调整, 每进行一次调整时要更新矩阵 A 和矩阵 E , 计算复杂度为 $O(m)$, 每个循环最多进行 $m/2$ 次调整, 因此每个时间步实施调整的计算复杂度为 $O(m^2)$ 。综合起来, 每个时间步的计算复杂度为 $O(nm + m^2)$ 。假设经过 T 次调整得到最大的模块度, 则计算复杂度为 $O[T(nm + m^2)]$, 即 $O(Tnm)$ 。改进后的算法根据每一组调整测试结果对多个文档进行调整, 减少了迭代次数 T , 因此, 改进后的文档调整效率高, 更加适合对大数据量文档集合进行调整。

4 实验结果分析

使用中文文本分类语料^[5]作为实验数据。数据集中共包含10个类别,共2815篇文档。所有文档经特征提取后表示为VSM向量形式,并使用 $tf-idf$ 计算各个特征维度的权值^[3]。把语料库中每个类别的文档平均分成2组,分别作为正确数据和噪声数据。把噪声数据随机分配到正确数据的其他任意一个类别中,这样生成了含有50%噪声比例的粗分类数据。生成数据中有1409篇文档类别标记是正确的,另外1406篇文档的类别标记是错误的。使用本文提出的算法对这些粗分类数据进行噪声修正处理,分别使用模块度^[4]和准确率^[6,7]来评价修正结果。修正过程中的模块度和准确率变化曲线如图1所示。

图1显示,与NNRA算法相比,当 $\alpha=1$ 时,本文所提算法的迭代次数都为1446次,修正结果的模块度为0.674,准确率为0.951。降低参数 α 的值,本文所提算法的收敛速度明显加快。当 $\alpha=0.9$ 时,算法只迭代526次,修正后的模块度和准确率与 $\alpha=1$ 时相同。当 α 继续减小时,迭代次数也略有减少,但减少幅度不大,修正结果的模块度和准确率也变化很小。我们分析,这可能与我们的实验中采用的实验数据有关。实验数据只包含10个类别,并且各个类别是按照相同的比例生成噪声数据。因此噪声修正中涉及各个类别的调整方案的模块度增量相差无几,因此阈值 α 的取值到达一定值后,继续减小对结果不会产生明显的影响。如果类别数量较多,或各个类别的噪声数量不等,或不同类别文档的相似程度不同的情况下,阈值 α 的不同取值对修正结果将产生怎样的影响,还有待于进一步考察。

图2和图3显示了针对不同比例的噪声数据,

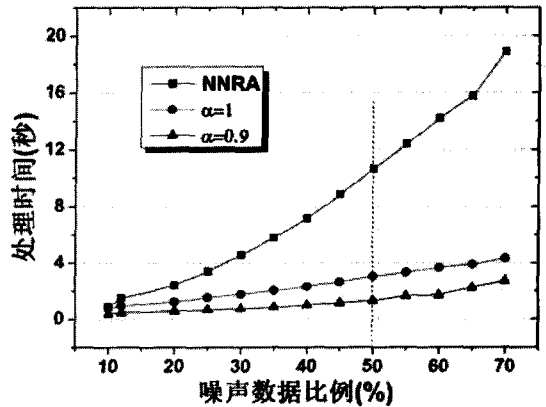


图2 对不同比例的噪声数据算法处理时间对比图

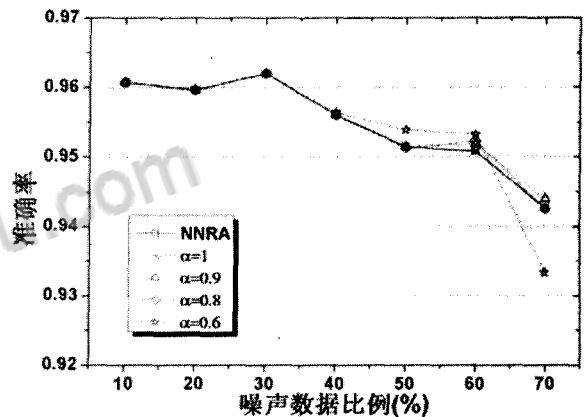


图3 对不同比例的噪声数据算法处理结果的准确率

本文所提算法与NNRA算法处理时间和处理结果准确率的对比。尽管本文所提算法在 $\alpha=1$ 时的迭代次数与NNRA算法相当,但在算法收敛时间方面本文所提算法具有明显的优势。从图中可以看出,在同样比例的噪声数据的情况下,本文所提算法的处理时间明显低于NNRA算法。比如,当噪声比例为50%时,NNRA算法的处理时间为10.6秒,本文所

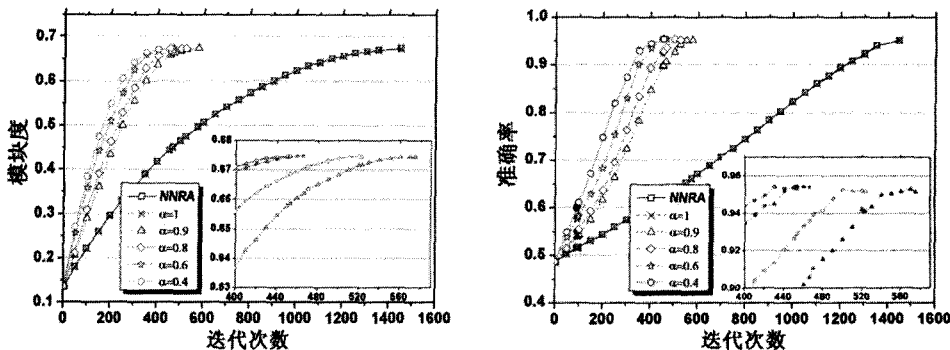


图1 对包含50%噪声比例的实验数据进行修正过程中模块度和准确率的变化曲线

提算法当 $\alpha = 1$ 时的处理时间为 3.02 秒, 当 $\alpha = 0.9$ 时的处理时间仅为 1.3 秒。说明本文所提算法的处理效率更高。从图 3 中可以看出, 当噪声数据比例不高于 40% 时, α 设置为 1 至 0.6 的处理结果的准确率没有变化。当噪声数据比例进一步增大后, 整体的处理结果的准确率有所降低, 不同的 α 取值对处理结果略有影响。

5 结 论

本文针对粗分类数据中的噪声修正算法, 提出了基于模块度增量判断噪声数据的改进算法, 较显著地提高了算法效率。同时本文通过分析发现, 数据类别修正对模块度的影响, 只与数据原来类别和修正后类别相关, 而与其他类别无关。并由此提出每次修正测试结果可用于对多个数据进行修正处理, 从而提高算法收敛速度。本文分别构造了不同噪声数据比例的实验数据, 并通过设置不同阈值对其进行修正处理, 算法能够快速收敛, 并获得准确的、稳定的结果。与原来算法相比, 本算法更适合大数据量的, 或实时性较强的实际应用。

本算法对静态数据进行修正处理, 但是在实际应用中, 很多数据是动态变化的, 比如期刊不断收录和刊登新论文, 网站不断增加新的网页等。对于增加了新样本的数据, 如果废弃以前的分析结果而完全重新进行修正处理是不切合实际的。下一步工作

将研究适合处理样本数据不断动态增加的改进算法。另外, 在样本数据不断增加过程中, 往往会产生出新的类别, 这在新闻网页的新主题、科研论文中的新的研究领域等方面都很常见。如何在数据分析过程中识别出这些新产生的类别也是下一步的研究方向。

参 考 文 献

- [1] 宣照国, 党延忠. 文本分类中粗分类数据噪声修正的网络算法[J]. 情报学报, 2008, 27(5): 670-676.
- [2] Alessandro Vinciarelli. Noisy text categorization[J]. IEEE transactions on pattern alalysis and machine intelligence, 2005, 27(12): 1882-1895.
- [3] Salton G, Wong A, Yang C S. A vector space model for automatic indexing[J]. Communications of the ACM, 1975, 18(11): 613-620.
- [4] Newman M E J, Girvan M. Finding and evaluating community structure in networks[J]. Physical Review E, 2004, 69, 026113, 1-15.
- [5] 李荣陆. 中文文本分类语料[OL]. [2007-02-10]. http://www.nlp.org.cn/docs/download.php?doc_id=281.
- [6] 程泽凯, 林士敏. 文本分类器准确性评估方法[J]. 情报学报, 2004, 23(5): 631-636.
- [7] Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval [M]. Addison Wesley, 1999.

(责任编辑 芮国章)



论文写作，论文降重，
论文格式排版，论文发表，
专业硕博团队，十年论文服务经验



SCI期刊发表，论文润色，
英文翻译，提供全流程发表支持
全程美籍资深编辑顾问贴心服务

免费论文查重：<http://free.paperyy.com>

3亿免费文献下载：<http://www.ixueshu.com>

超值论文自动降重：http://www.paperyy.com/reduce_repetition

PPT免费模版下载：<http://ppt.ixueshu.com>

阅读此文的还阅读了：

- [1. 数据采集算法的研究](#)
- [2. 电阻应变式称重传感器噪声信号的分离](#)
- [3. 柴油机瞬态工况振动噪声的测量和分析](#)
- [4. 实现多字节数基本运算功能的算法研究](#)
- [5. 基于ARM的多轴伺服控制器设计及其在7自由度数据臂中的应用](#)
- [6. 公路工程检测试验数据的处理方法](#)
- [7. 一种链式存储方式下的排序算法——串链法](#)
- [8. 企业厂界噪声监测问题的探讨](#)
- [9. 柴油机瞬态工况振动噪声的测量和分析](#)
- [10. 数据处理的一种算法](#)
- [11. 一种关键词抽取方法研究](#)
- [12. 一种奇异混合信号盲分算法的稳定性分析](#)
- [13. 计算机中数据加密技术](#)
- [14. 我国农村城市化影响因素的计量经济分析](#)
- [15. 格网中p-cycles重配置的快速实现算法](#)
- [16. 求最短路径的新方法](#)

[17. An adaptive time delay estimation algorithm based on quadratic weighting of the frequency domain](#)

[18. 基于单片机的汽车内后视镜反射率测试台](#)

[19. 文本挖掘在Web中的技术分析](#)

[20. 多目标基因算法用于脉冲中子测井数据处理，增强神经网络体的选择](#)

[21. 安全层次系统及其密钥生成算法](#)

[22. Excel录入技巧](#)

[23. 多维数据插值算法及其仿真应用](#)

[24. 关于逐时平均光照强度的算法的修正](#)

[25. 新的厂界噪声标准实施后的几点认识](#)

[26. 基于分割策略的快速抽题组卷算法](#)

[27. Pro/ENGINEER软件在汽车覆盖件逆向工程中的应用](#)

[28. 开启RFID普及大门 成本、标准与识别率为关键之钥](#)

[29. 文本粗分类数据中噪声的快速修正算法](#)

[30. 模糊聚类分析在文本分类中的应用](#)

[31. Notes on the Algorithm for Calculating Betweenness](#)

[32. 数据结构教学方法的探讨](#)

[33. 大批量数据出入数据库方法研究](#)

[34. 一种基于VSM的中文网页分类方法](#)

[35. 一种改进的中值滤波算法](#)

[36. 卫生统计数据网络传输的安全性](#)

[37. 安全层次系统及其密钥生成算法](#)

[38. VB编程实现多种格式数据的集成存取](#)

[39. 快速网格去噪声算法](#)

[40. 移动存储技术的变迁](#)

[41. 地籍空间数据质量检查与修正](#)

[42. 基于区分能力大小的启发式约简算法的研究](#)

[43. 机器新闻发展的市场进路及反思——以A-utamatedInsights公司为例](#)

[44. A SVM Based Text Steganalysis Algorithm for Spacing Coding](#)

[45. 抑制成像数据噪声的模糊多目标算法](#)

[46. 美國對付快速機動目標的新舉措——“戰爭破壞者”計劃](#)

[47. 多维数据插值算法及其仿真应用](#)

[48. 基于中心偏置特性的快速运动估计算法](#)

[49. 信念修正的实现方法](#)

[50. 《数据结构》课程的体系结构和学习方法](#)