

基于改进 K 最近邻分类算法的不良网页并行识别

徐雅斌^{1,2*}, 李卓^{1,2}, 陈俊伊¹

(1. 北京信息科技大学 计算机学院, 北京 100101; 2. 网络文化与数字传播北京市重点实验室(北京信息科技大学), 北京 100101)

(* 通信作者电子邮箱 xyb@bistu.edu.cn)

摘要:互联网中,黄色、暴力、赌博、反动等不良网页大量存在。如果不进行有效过滤,将给搜索服务带来不良的影响。采用改进的 K 最近邻分类算法来提高识别的准确率,并在虚拟化平台上通过开源的 Hadoop 软件所提供的 MapReduce 模型进行分布式并行处理。对比实验结果表明,所采用的识别方法的识别准确率和识别效率都有较大的提高。

关键词:不良网页;文本分类;K 最近邻分类算法;Hadoop;MapReduce

中图分类号: TP393 **文献标志码:** A

Parallel recognition of illegal Web pages based on improved KNN classification algorithm

XU Yabin^{1,2*}, LI Zhuo^{1,2}, CHEN Junyi¹

(1. Computer School, Beijing Information Science and Technology University, Beijing 100101, China;

2. Beijing Key Laboratory of Internet Culture and Digital Dissemination Research
(Beijing Information Science and Technology University), Beijing 100101, China)

Abstract: There are many illegal Web pages on the Internet, which may have pornographic, violent, gambling or reactionary content. Without being filtered effectively, they will exercise a malign influence on the searching services. An improved K-Nearest Neighbors (KNN) classification algorithm to promote the recognition accuracy was proposed and implemented on a virtualized platform following the MapReduce model provided by the open source software Hadoop, which made it distributed and parallel. Through experiments and comparison with the existing work, it is proved that the proposed recognition method improves the accuracy and efficiency greatly.

Key words: illegal Web page; text classification; K-Nearest Neighbors (KNN) classification algorithm; Hadoop; MapReduce

0 引言

随着互联网覆盖范围和应用人群的不断扩大,以及带宽的逐渐增加,给人们的工作、生活和学习带来了很大的便利。但随之而来的,也涌现了包含大量黄色、暴力、赌博甚至是反动内容的网页,而且这类不良网页信息正在全球范围内呈现蔓延和泛滥之势。有些不良网页为了避免被禁止访问,往往在某些健康网页上添加链接。这样,搜索服务在利用爬虫软件爬取健康网页的过程中,就不可避免地会爬取到一些不良网页。一方面,由于相关教育知识的缺失,不良信息网页往往有着强大的受众群体,因而会在一定程度上危害青少年的身心健康;另一方面,经常有大量木马病毒植入该类网页,这样,所爬取的网页还可能对网络搜索系统和用户系统的安全构成很大的威胁。因此,识别并过滤掉这类网页具有非常重要的意义。

不良网页的识别比较困难,一个最大的问题是不同的人对不良网页有着不同的定义^[1]。如美国法律规定:“整体上以裸体、性为内容的信息”为黄色信息,我国刑法第 367 条规定“具体描写性行为或者露骨地宣扬色情等具有淫秽性”的信息为色情信息,本文采用后者作为黄色网页的定义和识别依据。

不良网页过滤(不良信息过滤)目前主要采用的方法^[2]有:1)分级法,根据网页的内容特征,采用分级策略,逐步过滤;2)URL 地址过滤,根据保存的网页 URL 信息决定其代表的网页是否被过滤;3)动态文本解析法,主要涉及文本表示和匹配技术。其中,前两种方法属于被动的方法,过滤效果有限;而第三种方法潜力较大,成为研究的热点。

吴慧玲等^[3]采用规则匹配方法,查找词典替换拼音,去除特殊符号,还原网页本来信息以检测网页内容;苏贵洋等^[4]通过增加邻近类别分类器,实现更高的过滤准确度;崔虹燕等^[5]以及杨晓懿等^[6]都指出使用特征词典来构造网页的文本特征信息;Lee 等^[7]根据文本中的词频信息进行识别,Du 等^[8]从不良网页和正常网页分别抽取特征向量用于识别,都取得了很好的效果;Wai 等^[9]训练贝叶斯分类器检测出了不良网页。

通过分析发现,以上文献虽然给出了不同的识别方法,但是并没有考虑到不良网页识别所面对的是爬虫软件所爬取的海量网页数据,也没有深入研究识别算法的效率问题。因此,本文主要针对海量网页数据环境下,研究不良网页的有效识别方法,以及如何通过分布式并行计算方法提高不良网页识别的效率。

收稿日期: 2013-07-30; **修回日期:** 2013-08-17。 **基金项目:** 国家社会科学基金重大项目(12&ZD234);国家自然科学基金资助项目(60973107);网络文化与数字传播北京市重点实验室资助项目(ICDD201106, ICDD201207)。

作者简介: 徐雅斌(1962-),男,辽宁锦州人,教授,CCF 会员,主要研究方向:云计算、物联网、下一代互联网;李卓(1983-),男,河南南阳人,讲师,CCF 会员,主要研究方向:无线网络、移动计算;陈俊伊(1984-),女,山东威海人,硕士研究生,主要研究方向:云计算、下一代互联网。

1 网页数据预处理

爬取到的网页在分类器处理之前,首先经过黑名单过滤,如果待分类网页的网址已经处于黑名单中,则直接将其判断为不良网页并剔除;否则,再进行后续的识别处理,如果被判定为不良网页,则同样将其加入到黑名单中。

但是,从网络上爬取到的网页数据属于半结构化数据,很难被机器所识别;而且网页源码中存在很多对于理解内容无帮助的标签和 JavaScript 脚本信息,需预先进行过滤。因此,首先需要对网页内容进行预处理,预处理的过程如下:

1) 统一编码:为了便于网页处理,对所有网页统一采用 GB2312 编码格式。

2) 网页清洗:清洗网页中含有的大量与研究无关的数据,只提取需要的正文、标题、网址等信息。

3) 中文分词:本文主要针对中文网页进行不良及垃圾网页识别,采用中国科学院计算技术研究所开发的分词系统 ICTCLAS 对网页中的数据进行分词处理。

4) 去停用词:对分词处理后得到的结果与停用词库中的词进行匹配,得到有效的词语集合。

经过预处理的网页需要表示成计算机能够识别的形式,即文本表示。本文的文本表示采用向量空间模型。

向量空间模型是将网页或文本的处理简化成向量空间中的向量运算,把每个文本用特征向量的形式表示。例如,一个文本 D ,将其表示为 $D(t_1, t_2, \dots, t_n)$ 的形式,其中, $t_i (1 \leq i \leq n)$ 表示特征项,按照每个特征项所包含信息不同,用 w_i 表示特征项 t_i 的权重,则整个文本可以表示为 $D(t_1, w_1; t_2, w_2; \dots; t_n, w_n)$,简写为 $D(w_1, w_2, \dots, w_n)$ 。

2 不良网页的特征识别与提取

2.1 不良网页的识别

本文对不良网页的识别暂不考虑图片,即不作图像识别,只对不良文本进行识别。不良文本主要通过其中所包含的特征词体现,因此特征词提取成为不良网页识别的关键。不良网页的特征词即为突出体现黄色、暴力、反动等内容的各种敏感词汇。如体现黄色内容的部分特征词有:黄色、色情、情色、激情、性交、交媾等;体现暴力内容的部分特征词有:炸、毁、毁灭、炸掉、毁掉、杀、杀人、杀死、杀掉、杀戮、刺杀、凶杀、杀手、砍、砍死、砍伤、砍断、砍掉等;体现反动内容的部分特征词有:法轮、法轮功、法轮大法、真善忍等。

事实上,不良网页唯恐被搜索服务机构或管理机构识别出来而被过滤或被追查,大量使用以下三种手法:

- 1) 使用拼音、同音字代替想要隐藏的关键词;
- 2) 使用近义词代替敏感词;
- 3) 使用“!、#、%、&”等符号将敏感词隔离开,实现隐藏信息的目的。

例如,部分黄色网页大量采用的同义词和近义词如下:

黄色:黄 se、黄色、皇淫;

色情:色青、涩情、se 情、色轻;

情色:青涩、清色、情 se;

激情:基情、急情、鸡情。

但同义词或近义词的存在,将在一定程度上降低识别的准确率。对此,需要分别建立基础词库和同(近)义词库,每次进行关键词查询时,需要分别进行扫描和匹配。

在程序执行时,该过滤方法能有效过滤掉较大一部分不良网页,但是容易出现“一刀切”的问题,即有很多正规的教

育机构或者性知识教育网站,出于正常教学目的,其网页中存在一些专业名词,也被当作不良网页被过滤掉。例如:如果词库中将“阴茎”为关键词,则只要网页中包含有该词汇,就会被作为不良网页过滤掉,这是不良网页过滤中常见的一种“误判”现象。

通过研究发现,在正规网站中,对该类知识的描述一般采用正规性的词汇,而在不良网站中通常使用俗语来描述,故此将俗语类词汇收集在同(近)义词库中。为了尽量减少误判现象带来的影响,本文采用如下策略加以辅助判别:如果某个词存在于同(近)义词库中,则给该词赋予较高的权重;如果存在于基础词库中,则赋予相对较低的权重。扫描整个网页,总的权重 W 等于各权重之和,如果 W 值超过一定的阈值,则判定该网页属于不良网页。

在本文的实验中,存在于基础词库中的每个特征词的权重取值为 0.1,存在于同(近)义词库中的每个特征词的权重取值为 0.2。经过人工进行有效性验证,将 W 的阈值确定为 0.6。

2.2 不良网页的特征提取方法

经过预处理后,网页中含有大量的特征词汇,如果全部采用,对于后续工作而言,工作量极大。首先,将网页表示成空间向量时维数较大,即便采用并行处理方式,时间开销也会显著增多;其次,给存储空间造成了压力;再次,有很多特征对网页的分类作用不大。可以看出,特征过多会导致整体处理速度下降,因此,有必要对特征进行选择,降低空间维数。

本文以文档频率(Document Frequency, DF)作为特征选择方法。根据某个特征项在所有文本集中出现的文本数目 n_i 与所有文本集总数目 n 的比值来判断。通常,DF 太小的特征项不具有代表性,而 DF 太大的特征项又存在于太多的文本中,因此不具有区分度^[10]。为此,设定两个阈值,分别代表最大阈值和最小阈值,低于最小阈值和高于最大阈值的特征项都舍去,只保留符合中间值的特征项。

在本文的实验中,取 DF 的最小阈值为 0.001,最大阈值为 0.0763。针对我们建立的网页数据集进行验证,该阈值区间平均可以覆盖 97.41% 的不良网页。

3 文本分类算法

3.1 K 最近邻文本分类算法

进行不良网页识别的核心技术是文本分类,而文本分类的关键是文本分类算法。因此,文本分类算法的选择或设计至关重要。在各种文本分类算法中,K 最近邻(K-Nearest Neighbors, KNN)分类算法比较直观,容易理解,当向数据集中添加新的数据时,不需要进行新一轮的训练,可以有效缩短训练时间,并且易于实现并行化。为此,本文采用 KNN 分类算法。

KNN 分类方法主要根据待测网页与训练集中不同类别的网页进行相似度计算,将计算得到的结果进行排序,选取其中最相近的 K 个网页,将待测网页分类到 K 个网页中数量最多的类别中。

文本间的相似度可转化为向量空间模型中两个文本的相似度(距离)的运算。本文中相似度采用余弦距离,如果所得余弦值越小,说明两个向量的夹角比较小,两个文本属于同一个类别的概率就越大;反之,两个文本属于同一个类别的概率就较小。

3.2 KNN 分类算法的改进

尽管 KNN 分类算法原理简单,但是其计算量较大,其对

空间及时间的要求都比较高^[11],文献[12]提出压缩训练集的思想,但是未能很好地体现每一个文本作为近邻样本的贡献程度,为了进一步提高运行的效率,本文对 KNN 压缩算法进行了改进。

在 KNN 算法中,决策边界附近的文本对分类的贡献较大,如何找到并保留训练集中靠近决策边界的样本对于压缩样本空间而言至关重要。

压缩算法:

1) 将整个训练集人工分类为 $D = \{D_1^{N_1}, D_2^{N_2}, \dots, D_m^{N_m}\}$, $D_i^{N_i}$ 表示第 i 个类别中的文本数为 N_i 。

2) 对于每个类别 D_i , 根据 KNN 算法求出其中每个文本 j 相对于非自身类别的文本的相似度,将相应最近邻的影响因子 (Impact Factor, IF) 值加 1。

3) 循环执行 2), 直到所有类别中的每个文本都计算完成。

4) 将各类别中的文本按影响因子 IF 的值降序排列,选取其中前 $n/2$ 个文本作为训练集样本, n 为压缩前文本的总数。

通过上述步骤,训练集只保留存在于分类边界附近的若干样本,极大地减轻了计算的工作量。通过算法描述可知,每个样本需要计算非自身类别的 m 个近邻值,其时间复杂度为 $O(m^2n^2)$ 。根据每个样本成为其他样本的近邻的次数,设定影响因子后,需要遍历所有的样本,求出压缩后的样本集,时间复杂度为 $O(n^2)$, 所以,总的时间复杂度为 $O(n^2)$ 。可见,随着样本集的增大,计算处理时间会成倍增长,采用并行方式处理提高效率显得尤为重要。

4 改进 KNN 分类算法的分布式并行处理

为提高大数据量的处理及运行速度,采用 MapReduce 架构进行并行化处理。整个过程分为分布式特征项 TFIDF 计算、分布式文本间余弦相似度的计算和分布式分类器实现三个部分。

4.1 TFIDF 的分布式计算

网页的特征提取采用 TFIDF 进行量化计算,以特征权值的形式表征每个特征项。

TFIDF 算法的核心思想是,如果某个文本特征(词或者短语)在一篇文本中经常出现 (tf 高),而在其他的文本中出现的概率 df 较小,那么该特征适合于分类。计算方法如式 (1) 所示:

$$w_{ij} = tf_{ij} * idf_j = \frac{|t_{ij}|}{|d_j|} * \lg\left(\frac{N}{n_i}\right) \quad (1)$$

其中: tf_{ij} 表示文本中特征项 t_i 在文本 d_j 中出现的频率, idf_j 是文档频率的倒数, $|t_{ij}|$ 表示特征词语 t_i 在文本 d_j 中出现的数量, $|d_j|$ 表示文档 d_j 中包含的所有词语的总数量, N 是指数据集中所有文档的总数目, n_i 是数据集中凡出现过特征项 t_i 的文档数据之和。

在求出 TFIDF 后,面对海量数据,其包含的特征词数目较多,为提高识别效率,需要特征降维,选出 TFIDF 值最大的若干个特征词。此处,利用 Hadoop 平台的 MapReduce 并行架构实现 TFIDF 计算的并行化,从而有效地提高运行效率。

在 Hadoop 平台中,数据经过 Map 函数处理之后,默认是按照键值升序排序,这与我们的要求不相符,而且采用二次排序也不适宜。因此,按照 $1 - TFIDF$ 值的方法取得升序排列的前若干位数值,即 TFIDF 按照降序排序的相应若干较大值。

4.2 余弦相似度的分布式计算

当计算两个网页的相似度时,可以采用向量相似度的计算方法。以余弦距离作为距离计算依据,向量的距离计算公式表示如下:

$$\cosDistance(D_1, D_2) = 1 - \frac{\sum_{k=1}^n w_{1k} * w_{2k}}{\sqrt{\left(\sum_{k=1}^n w_{1k}^2\right) \left(\sum_{k=1}^n w_{2k}^2\right)}}$$

其中: w_{1k} 表示文档 D_1 中第 k 个特征值, w_{2k} 表示文档 D_2 中第 k 个特征值,特征总数为 n 。由该公式可知,要求出两个文档的余弦距离,需要知道每个文档的所有特征值,并且进行点积和平方根乘积的运算。由于任意两个网页文本之间的相似度运算并没有依赖性,因此也可以并行计算,其并行计算过程略。

4.3 KNN 分类器的分布式计算

KNN 算法主要根据不同文本之间的余弦相似度求出最近邻的 K 个文本项。任意两个文本间相似度的计算及相似度排序都互不相关,因此可以并行进行。训练集的压缩及测试集的测试过程如下:

1) 求出每个类别中文本在非自身类别的 m 个最近邻文本,将其影响因子值设置为 1。

Map 阶段输入:

$(\text{textfile}, \text{fileName}_1 = \cosDistance_1;$

$\text{fileName}_2 = \cosDistance_2; \dots; \text{fileName}_i = 0; \dots;$

$\text{fileName}_n = \cosDistance_n)$

进行格式转换后,输出 $((\text{fileName}_i, \cosDistance_i), \text{fileName}_j; \text{type}_j)$ 。经过二次排序, Reduce 得到经过排序的 Map 阶段的计算结果,选择前 m 个样本,输出为 $(\text{filename}, \text{type})$ 。

所有的 Map 节点读取余弦相似度阶段得到的结果,并且根据每个文本的类别,选出与其自身类别不同的文本相似度,进行格式转换,输出结果为 $((\text{fileName}_i, \cosDistance_i), \text{fileName}_j; \text{type}_j)$ 。其中: \cosDistance_{ij} 表示文本 D_i 和 D_j 的余弦相似度, type_j 表示文本 D_j 的类别。

Map 阶段输出结果中的 key 值是一个自定义数据组形式 $KeyPair(\text{String}, \text{float})$, MapReduce 框架中 Map 过程和 Reduce 过程之间默认会根据 key 值进行排序,即按照 fileName 排序,但是我们还需要将同一个 fileName 相关的 \cosDistance 分配到同一个 Reduce 节点上,并且将相同 fileName 的 \cosDistance 值进行降序排列。

经过二次排序处理后, Reduce 阶段接收 Map 的输出作为输入,即 $((\text{fileName}_i, \cosDistance_i), \text{fileName}_j; \text{type}_j)$ 根据排序结果,选取每个文档的前 500 个最近邻的文档进行输出,输出结果为 $(\text{filename}, \text{type})$, 表示 key 中该 fileName 成为其他的文档的近邻 1 次。

2) 计算每个文本的总的影响因子值。

Map 阶段接收上一个 MapReduce 的输出,即 $(\text{filename}, \text{type})$, 该键值对表示文本 fileName 成为其他文本近邻 1 次,改变表示形式,输出 $(\text{fileName}@\text{type}, 1)$ 。Reduce 阶段接收 Map 的输出,统计相同 fileName_1 作为其他文本的近邻的总次数 n_1 , 以 $(\text{fileName}@\text{type}, IF)$ 的形式输出。

所有 Map 节点从第一个 MapReduce 节点获得作为其他文本近邻的文本名称和类别,将其影响因子 IF 记为 1, 输出 $(\text{fileName}@\text{type}, 1)$; 然后 Reduce 阶段对具有相同 key 的 $value$ 值求和,得出结果为文本 fileName 在所有的文本中作为其他近邻文本的总的次数并输出,为后来的训练集选择做准备。

3) 选取每一类别中影响因子较大的一些文本, 压缩训练集。

Map 阶段根据统计得出的影响因子值, 输出 $((type, IF), filename)$ 。经过二次排序之后, Reduce 函数将每一类别具有较大影响因子的文本输出, 作为训练集文本, 输出形式为 $(type, filename_1, filename_2, \dots, filename_n)$ 。

所有 Map 节点读入上一个 MapReduce 的输出结果, 将其转换为适于二次排序的格式, 即 $((type, IF), filename)$ 。这样, 在 Map 阶段之后, MapReduce 会将具有相同 $type$ 的数据分配到同一个 Reduce 中, 并且按照 IF 值的降序排序。Reduce 读取排好序的数据后, 选择前 n 个文本, 从而有效地将训练集压缩。

在采用 KNN 分类方法进行测试时, 如同训练集中的文本, 待测文本首先也需要进行向量化表示, 然后计算待测文本与训练集中每个文本的相似度, 从中找到最相近的 K 个文本。经过余弦相似度计算后, Map 函数输入为:

$(testFile, filename_1 = \cosDistance_1; filename_2 = \cosDistance_2; \dots; filename_i = \cosDistance_i; \dots; filename_n = \cosDistance_n)$

进行格式转换, 输出结果为 $((testFile, \cosDistance_i), filename_i, type_i)$; 经过二次排序后, Reduce 阶段进行类别判定, 输出 $(testFile, type)$ 。所有的 Map 节点接收经过余弦相似度计算的结果 $testFile$:

$filename_1 = \cosDistance_1; filename_2 = \cosDistance_2; \dots; filename_i = \cosDistance_i; \dots; filename_n = \cosDistance_n$

将其转换为 $((testFile, \cosDistance_i), filename_i, type_i)$ 的形式, 其中 $type_i$ 为 $value$ 值中对应的文本 $filename_i$ 的类别。该结果经过二次排序后, 相同的测试文本 $testFile$ 会被分配到同一个 Reduce 节点上, 并且按照 \cosDistance 的降序排列。Reduce 函数以该结果为输入, 选取每个 $testFile$ 的余弦相似度最大的 K 个文本, 读取文本的类别进行统计, 选择类别个数最大的一类作为该测试文本的类别, 并且输出结果。

5 实验结果与分析

5.1 实验环境与实验数据

实验环境共有 5 台服务器, 其中 1 台为双 CPU 服务器, 另外 4 台为单 CPU 服务器; 实际构成了 6 个物理节点的集群计算环境, 采用 VMWare 5 虚拟化平台软件构成一个集群计算环境, 其中 1 个作为管理节点, 其余 5 个为计算节点; 开源云计算平台 Hadoop 的版本为 0.20.2; 将其中 1 个节点作为 NameNode 和 JobTracker 服务节点, 其他 5 个节点作为 DataNode 和 TaskTracker, 每个节点配置为 CPU 2.3 GHz, 内存 8 GB, 硬盘配置为 600 GB, 千兆网络控制器, 共计 16 TB 存储。

网页数据的采集通过笔者设计的网络爬虫软件来实现, 从新浪、网易的各板块获得正常网页 6946 个, 通过搜索引擎和可疑链接等从互联网上搜集到不良网页 582 个。

5.2 单机与 Hadoop 平台识别时间比较

采用 Hadoop 平台与单机环境中运用 KNN 算法进行分类识别, 在速度方面的性能对比如图 1 所示。

从图 1 中可见, 采用 Hadoop 分布式并行处理环境后, KNN 算法的运行时间明显降低。

KNN 算法改进前与改进后的对比如图 2 所示。从图 2 中可以看出, 在只有 1 个节点的情况下, Hadoop 平台上的执行时间要高于单机的执行时间, 随着节点数从 2 开始逐渐增

多时, 程序在 Hadoop 平台上的执行时间逐渐减少, 并低于单机运行时间, 当节点数目增加到 4 之后, 运行时间减少的幅度逐渐趋于平缓。

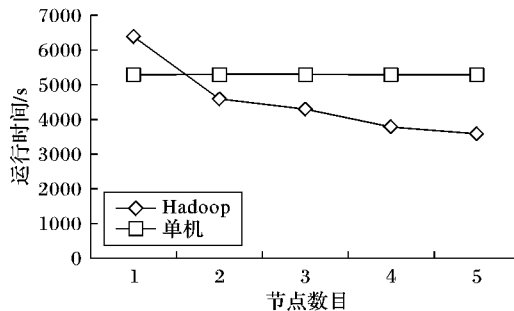


图1 单机与 Hadoop 运行时间比较

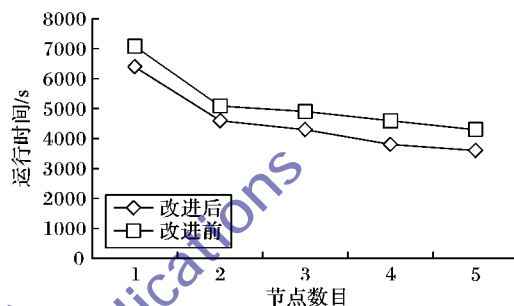


图2 KNN 算法改进前后运行时间对比

5.3 KNN 算法改进前后性能比较

算法改进前后在准确率与召回率方面的对比情况分别如图 3、4 所示。

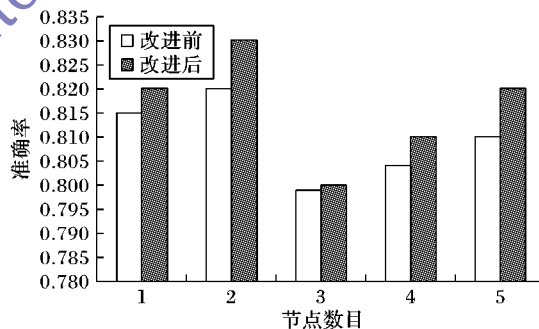


图3 KNN 算法改进前后准确率对比

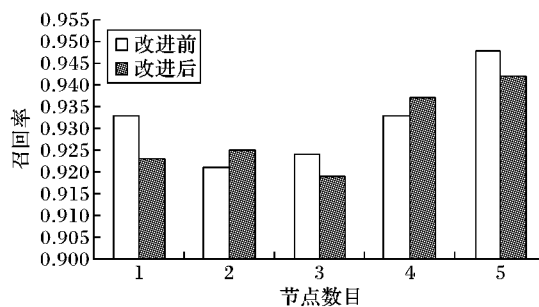


图4 KNN 算法改进前后召回率对比

由对比实验结果可以发现, 改进后算法的运行时间减少, 分类准确率有所提高, 但是召回率稍有降低。分析其原因, 主要是压缩算法中只保留分类边界周围的样本点, 而除此之外的样本点都被删除掉了, 这样使得训练数据更集中, 每个样本代表的信息更加精确, 因此提高了准确度。但是训练样本的减少, 不可避免地导致部分信息丢失, 在分类时与该部分信息同类的网页不能被查询到。但从整体来看, 算法的改进是有意义的。

(下转第 3379 页)

4 结语

本文主要研究了用动态多粒子群保持多目标粒子群优化算法如何保持粒子多样性的问题,首先从粒子群生成着手,根据粒子群的分布情况在决策变量空间动态生成粒子群,为避免粒子群收敛过快改变了粒子选择粒子最优解和群最优解的方法,并在粒子群搜索解的过程中动态增加或减少粒子群的个数。测试表明这个方法能有效改变粒子的分布情况,对粒子群多目标优化算法的改进有一定的意义。

参考文献:

- [1] 公茂果,焦成,杨咚咚,等. 进化多目标优化算法研究[J]. 软件学报, 2009, 20(2): 271-289.
- [2] REYES-SIERRA M, COELLO C A C. Multi-objective particle swarm optimizers: a survey of the state-of-the-art [J]. International Journal of Computational Intelligence Research, 2006, 2(3): 287-308.
- [3] MARLER R T, ARORA J S. Survey of multi-objective optimization methods for engineering [J]. Structural and Multidisciplinary Optimization, 2004, 26(6): 369-395.
- [4] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of the 1995 IEEE International Conference on Neural Networks. Piscataway: IEEE, 1995: 1942-1948.
- [5] KUMAR R, SHARMA D, SADU A. A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch [J]. International Journal of Electrical Power and Energy Systems, 2011, 33(1): 115-123.
- [6] RABBANI M, BAJESTANI M A, KHOSHKHOU G B. A multi-objective particle swarm optimization for project selection problem[J]. Expert Systems with Applications, 2010, 37(1): 315-321.
- [7] CHANG R I, LIN S Y, HUNG Y. Particle swarm optimization with query-based learning for multi-objective power contract problem[J]. Expert Systems with Applications, 2012, 39(3): 3116-3126.
- [8] BRIZA A C, NAVAL JR P C. Stock trading system based on the multi-objective particle swarm optimization of technical indicators on

end-of-day market data[J]. Applied Soft Computing, 2011, 11(1): 1191-1201.

- [9] 岳林,易本顺,肖进胜. 能量平衡与 QoS 保障的无线传感器机会路由[J]. 湖南大学学报: 自然科学版, 2011, 38(11): 82-87.
- [10] MOUSA A A, EL-SHORBAGY M A, ABD-EL-WAHED W F. Local search based hybrid particle swarm optimization algorithm for multiobjective optimization [J]. Swarm and Evolutionary Computation, 2012, 3(1): 1-14.
- [11] 任子晖,王坚. 动态拓扑结构的多目标粒子群优化算法[J]. 同济大学学报: 自然科学版, 2011, 39(8): 1222-1226.
- [12] HERNÁNDEZ-DÍAZ A G, SANTANA-QUINTERO L V, COELLO COELLO A C, *et al.* Improving the efficiency of ϵ -dominance based grids [J]. Information Sciences, 2011, 181(15): 3101-3129.
- [13] 聂瑞,章卫国,李广文,等. 一种自适应混合多目标粒子群优化算法[J]. 西北工业大学学报, 2011, 29(5): 695-701.
- [14] ZHANG Y, GONG D W, DING Z H. Handling multi-objective optimization problems with a multi-swarm cooperative particle swarm optimizer [J]. Expert Systems with Applications, 2011, 38(11): 13933-13941.
- [15] KNOWLES J D, CORNE D W. Approximating the nondominated front using the Pareto archived evolution strategy [J]. Evolutionary Computation, 2000, 8(2): 149-172.
- [16] ZHANG Q F, ZHOU A M, ZHAO S Z, *et al.* Multiobjective optimization test instances for the CEC2009 special session and competition[EB/OL]. (2009-04-20) [2013-01-19]. <http://decs.essex.ac.uk/staff/zhang/MOEAcompetition/cec09testprob/lem0904.pdf>.
- [17] COELLO COELLO C A, LECHUGA M S, LECHUGA M S. MOPSO: a proposal for multiple objective particle swarm optimization [C]// CEC'02: Proceedings of the 2002 Congress on Evolutionary Computation. Piscataway: IEEE, 2002: 1051-1056.
- [18] LI H, ZHANG Q F. Multiobjective optimization problems with complicated Pareto sets MOEA/D and NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 284-302.

(上接第 3371 页)

6 结语

不良网页的大量存在,不仅严重影响了搜索引擎的检索效率,而且降低了用户体验的满意度。因此,搜索服务非常有必要对不良网页进行识别和过滤。在设计和实验过程中,我们编写了爬虫软件,爬取了大量的网页,建立了实验数据集,并对实验数据集中的网页进行了清洗和过滤,分析了不良网页的特征,给出了行之有效的特征提取方法。鉴于 KNN 分类算法的计算量较大,需要将所有训练文本都存储起来,对空间及时间的要求都比较高,本文对 KNN 压缩算法进行了改进,并通过 Hadoop 平台上提供的 MapReduce 编程模型进行了分布式并行计算,不仅给出了实验结果,而且还对不同环境下的实验结果进行了分析和比较,结果证明了改进的 KNN 算法和基于 MapReduce 模型进行并行化计算的有效性。

参考文献:

- [1] HU W M, WO O, CHEN Z Y, *et al.* Maybank: recognition of pornographic Web pages by classifying texts and images [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(6): 1019-1034.
- [2] 施聪莺. 针对青春期少年的网络色情文本过滤技术研究[D]. 南京: 南京师范大学, 2010.
- [3] 吴慧玲,沈建京,贺广生. 基于不良文本信息过滤预处理方法的

研究[J]. 网络安全技术与应用, 2006(11): 61-63.

- [4] 苏贵洋,马颖华,李建华. 一种基于内容的信息过滤改进模型[J]. 上海交通大学学报, 2004, 38(12): 2030-2034.
- [5] 崔虹燕,蒋念平. 一种改进的多级信息安全过滤模型[J]. 情报理论与实践, 2006, 29(5): 615-617.
- [6] 杨晓懿,刘嘉勇. 基于内容的信息安全过滤技术[J]. 信息安全, 2004(4): 47-49.
- [7] LEE P Y, HUI S C, FONG A C M. An intelligent categorization engine for bilingual Web content filtering [J]. IEEE Transactions on Multimedia, 2005, 7(6): 1183-1190.
- [8] DU R, SAFAVI-NAINI R, SUSILO W. Web filtering using text classification [C]// ICON 2003: Proceedings of the 2003 11th IEEE International Conference on Networks. Piscataway: IEEE, 2003: 325-330.
- [9] WAI H H, PAUL A W. Statistical and structural approached to filtering Internet pornography [C]// Proceedings of the 2004 IEEE International Conference on System, Man and Cybernetics. Piscataway: IEEE, 2004, 5: 4792-4798.
- [10] 潘文锋. 基于内容的垃圾邮件过滤研究[D]. 北京: 中国科学院计算技术研究所, 2004.
- [11] 刘慧. 基于 KNN 的中文文本分类算法研究[D]. 成都: 西南交通大学, 2010.
- [12] 文思. 基于 Hadoop 的 K 近邻分类算法的研究[D]. 广州: 华南理工大学, 2011.