第 36 卷 第 1 期 计 算 机 工 程 2010 年 1 月
Vol.36 No.1 Computer Engineering January 2010
·软件技术与数据库· 文章编号：1000—3428(2010)01—0040—03 文献标识码：A 中图分类号：TP311.12

基于存储过程的海量邮件数据挖掘

郭绍忠，甄 涛，贾 琦
(解放军信息工程大学信息工程学院，郑州 450002)

摘 要：现有的邮件系统缺少对海量邮件数据的分析和挖掘功能，传统的对单封邮件的分类方式效率低下。针对该问题，研究文本挖掘特点，提出一种基于海量关系型数据库存储过程实现的高效的海量邮件内容数据挖掘算法，并对算法进行多个级别的性能优化。实验结果表明，该算法具有高效性、稳定性和普适性。

关键词：邮件分类器；数据挖掘；存储过程

Data Mining of Massive Mail Based on Storage Procedure

GUO Shao-zhong, ZHEN Tao, JIA Qi
(School of Information Engineering, PLA Information Engineering University, Zhengzhou 450002)

【Abstract】It is short of the functions to analysis and mining great capacity mail data on existing mail data engine. Aiming at this problem, this paper describes and optimizes an efficient great capacity mail data mining algorithm based on directly storage procedure of Relational Database Management System(RDBMS) on performance on many levels after the character of text mining characteristic is studied. Experimental results demonstrate that this algorithm is effective, stable and adaptable.

【Key words】mail classifier; data mining; storage procedure

1 概述

邮件信息系统能帮助用户管理分析海量邮件数据，具有多种用途。本文通过对主题和正文的数据挖掘查找属于某一种特定类别的邮件。由于用户查询常常不能够准确地表达用户的信息需求，因此获取的数据不一定能够满足用户的需求。

为了满足用户对文章准确分类的需求，本文采用通过对已知类别的文章进行分词学习来获得查询条件，在数据库中对邮件正文数据进行查询分析的方法。在数据库中采用存储过程对数据库进行复合式的查询。

存储过程作为数据库对象存储在数据库中，保存在服务器端，它减少了数据传递造成的时间开销，即“分析一次，执行多次”。存储过程的数据类型与 SQL 语言数据类型保持一致，无需进行数据转换，也不存在诸如打开或关闭查询等一系列步骤。同时，存储过程还可以降低整个应用程序的复杂性，具有可移植性、重用性、安全性和伸缩性等特性。因此，使用存储过程操作数据库可以有效地提高程序的实行效率[1]。

本文研究文本挖掘特点，提出一种基于海量关系型数据库存储过程实现的高效的海量邮件内容数据挖掘算法，并对算法进行多个级别的性能优化。在信息库中的文本一般是通过索引词条来表示。但是对海量数据建立全文索引代价非常高。由于邮件分类系统针对的是少数用户，目标是少数用户感兴趣的分类，因此本文邮件正文没有建立索引，无须分词，查找时进行全文匹配。

2 基于存储过程的海量邮件数据挖掘算法

本文提出的算法总体设计流程如图 1 所示。

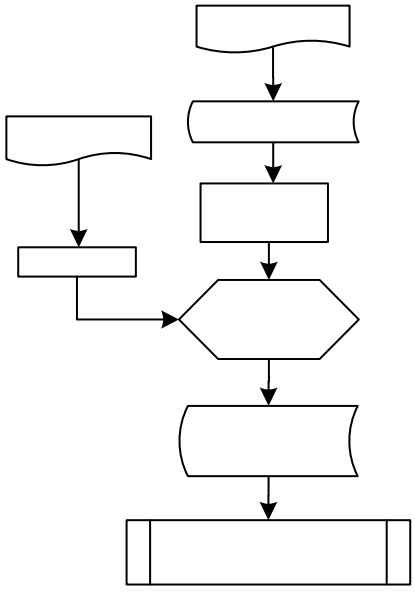

图 1 算法设计流程

2.1 数据准备

2.1.1 数据装入

采用VB.NET将邮件数据装入数据库。利用ChikatDotNet可以解析邮件，得到邮件的各个信息，如收发地址、收发姓名、主题、正文、附件等(具体使用规则参考 Chikat_NET Email Class Reference(VB_NET).htm)。在装入过程中需要对垃圾邮

基金项目：国家“863”计划基金资助项目(2007AA01Z146)
作者简介：郭绍忠(1964－)，女，副教授，主研方向：数据挖掘，海量信息处理；甄 涛、贾 琦，硕士研究生
收稿日期：2009-06-02 **E-mail：**12355200@qq.com

件进行初步的过滤。将内容相同并且收发人也相同的邮件去掉。然后给剩下的邮件信息加上 ID 号后存入数据库中。将邮件数据在用户机解析然后通过网络传输给服务器端再装入数据库，每执行 10 000 条所用时间是 4 min。在服务器端解析可减少网络传输消耗的时间。

2.1.2　文本预处理

文本预处理过程可以分如下 5 个步骤[2]：

(1)文本的词法分析，主要处理文本中的数字、连接符、标点符号和字符大小写；

(2)无用词汇的删除，主要过滤掉那些对于信息获取过程来说区分能力低的词汇；

(3)词干提取，主要去除词缀(前缀和后缀)，这样可以允许所获取的文档包含一些查询词条的变化形式；

(4)索引词条/词干的选择，在选择时通常按单词的习惯用法，实际上名词往往要比形容词、副词和动词包含更多语意；

(5)构造词条的分类结构。

2.1.3　利用中文分词技术获取特征词

通过对样本邮件进行学习得到某类样本邮件的特征词。首先要将样本邮件分词，目前只考虑对中文和英文的文章进行分词。英文比较容易，是通过空格分割开的，注意区分大小写即可。中文分词采用的方法是用字符串逐段查找中文字典哈希表。如果某段字符串存在于字典中，就认为它是一个词。这里采用逆向最大分词获得较高的准确率。

对分好的词进行词条频度和文章频度的统计。通过对某专业领域的了解，用特征选取算法得到该类文章的特征词集合。以及每个特征词的权重。太频繁出现的单词将不会成为具有良好区分能力的词汇。实际上，如果一个单词出现在信息库中 80%的文本中，该单词对于信息获取过程来说根本没用，在选择特征词的时候，这些词要被去掉。

邮件中可能包含某些特殊形式的字符串，例如网址、电话号码、保险号码、邮件地址等。这些格式相同内容不同的字符串，可以运用正则表达试作为特征来实现对其的匹配。如“%^([-a-zA-z0-9_]+)@(([-a-zA-z0-9_]+[.])+[a-zA-z]+)$%”表示的是字符串中的电子邮件地址。同样，正则表达式可以对文本结构不是很好的文章、含有容易出现错别字的单词的文章进行查询。并且可以大量的排除语义上存在着的某些歧义性。

2.2　算法实现

当特征词已经由事先对特征文章进行分词学习得到或通过经验人工指定。本文将同一类别的特征词保存在一个表中。利用游标对海量邮件的文本部分进行递归循环，统计特征词在每篇文章中出现的次数。特征词表如下：

keyword
数据库
数据挖掘
机器学习
分类
…

通过游标依次遍历特征词表的每一行，每次将该行的特征词提取出来。每次取出一个特征词，传递给变量。在两边加上通配符，作为一个循环查询条件。

对于大数据量的巨型数据表而言，全表扫描查询是一项极为耗时的查询操作。由于含有某些特征词的文章只是所有文章的一部分，因此把含有这些特征词的文章取出，在该范围内进行搜索大大提高了程序的效率。包含特征词的文章的表记录是全表记录的子集，假设表的记录数为 N，特征词出现的文章的记录数为 MM，则 M/N 的取值在[0, 1]之间，若 M/N 的值越小，查询的性能改进越明显。

Banner 位为过程标志位。保证了搜索第 2 次出现的特征词的文章时只在出现第 1 次该特征词的文章里搜索。

对内容的分析可以只考虑以下 3 个内容，所以只将这 3 项提取出来。邮件数据表如下：

ID	Content(邮件内容)	Subject(邮件主题)

本文对邮件数据表进行初步分析，找到全部包含第一个特征词的邮件。并用其 ID 做为主码在结果存放表(表 1)中插入相应行。

表 1　结果存放表

ID	数据库	数据挖掘	机器学习	分类	…	Banner
1	0	0	0	0	…	1
234	0	0	0	0	…	1
4567	0	0	0	0	…	1
6678	0	0	0	0	…	1
…	…	…	…	…	…	1

匹配每篇文章的特征词的代码如下：

```
where (subject like word or content like word) and ID not in (select ID from resultable);
```

然后循环找出所有特征词分别出现过的邮件，建立相应行。利用循环在邮件数据表包含的文章中搜索每个特征词的出现次数。采用的方法是：第 1 次对 word='%newword%'进行 LIKE 匹配，把 select 到的文章的相应特征词出现次数+1。(%号的作用是忽略特征词左右相临的内容)下次只需在上次的结果上对 word=(word||'%newword%')进行二次匹配，把 SELECT 到的文章的相应特征词出现次数+1。找到出现 2 次该词的文章。接着循环直到结束。这样就可以通过循环递归找到每个文章中出现每个特征词的次数。具体代码如下：

```
select count(*) into w from datatable where banner=1 and subject like word or content like word;
if w=0
then
exit;
end if;
t:=t+1;
declare
cursor like_cursor
is
select id from datatable where banner=1 and subject like word or content like word;
begin
for i in like_cursor loop
key:=i. id;
update resultable set banner=0;
execute immediate 'update resultable set '||newword||'='||t||' and banner=1 where resultable.eid='||key;
end loop;
end;
```

代码运行得到的结果如表 2 所示。若要得到某个特征词在其出现的文章中的权重。先要知道文章的长短。因此，本文利用 LENGTH 这一字符串函数获得文章长度，作为后期分析的参数。

表 2　数据存放结果

ID	数据库	数据挖掘	机器学习	分类	…	banner
1	4	0	2	0	…	0
234	0	0	3	12	…	0
4567	15	0	0	4	…	0
6678	0	0	5	9	…	0
…	…	…	…	…	…	0

3　基于贝叶斯算法的文本类别计算

将文本表示为向量的形式。设等待分类邮件文本为 d_x，n 为特征数量，特征在这里就是特征词条，c_i=1/0 分别表示 2 个类别。特征空间($t_1, t_2,\cdots, t_n$)中的特征就是该封邮件中出现的具体特征词条。

计算文本 d_x 属于某个类别的概率，将文本归入具有最高后验概率的类别 c_i 中去。根据贝叶斯定理[3]:

$$P(c_i \mid d_x) = \frac{P(d_x \mid c_i)P(c_i)}{P(d_x)} \tag{1}$$

其中，$P(c_i)$ 是类的先验概率；$P(d_x \mid c_i)$ 是类条件概率；$P(d_x)$ 对于所有的类为常数。

由于 $P(t_k \mid c_i)$ 可以由训练样本集估值：s_{ik} / s_i 得到。其中 s_{ik} 是词条 t_k 属于类别 c_i 的训练样本数；s_i 是类 c_i 中的邮件训练样本数，因此可以得到文本 d_x 的类条件概率:

$$P(d_x \mid c_i) = P(t_1 \mid c_i)P(t_2 \mid c_i)\cdots P(t_n \mid c_i) = \prod_{k=1}^{n} P(t_k \mid c_i) \tag{2}$$

由于 $P(c_i) = s_i / s$，得到 $P(c_i \mid d_x)$。当 $P(c_i \mid d_x)$ 大于测定的阈值时，c_i 表示的类别即 d_x 表示的邮件文本所属的类别[4]。

4　实验分析

实验系统的软、硬件环境配置如下：主机：联想；操作系统：Windows2003；数据库：Oracle 10g；接口的实验平台：JAVA, ADO，算法试验平台：PL/SQL, Oracle 10g。数据总记录数为 l 000 000。

将分词建立索引的方法和本文提出的通过存储过程实现的邮件分析方法进行比较。实验结果如表 3 和表 4 所示，表 4 的总记录数为 1 000 000。结果表明后者的检索性能更优。由于前者需要对所有文章分词并且维护庞大的索引表，所以耗费大量时间。后者效率明显优与前者。能够满足应用需求。

表 3　不同数据量下对同一类别文本的分类时间

总记录数	查询某一类别检索时间/s
4 000	5.3
10 000	62.2
100 000	822.0
1 000 000	9 534.0

表 4　对不同类别文本的分类时间

查询不同类别文本的返回记录数	检索时间/s
245	8 708
347	9 032
822	9 455
2 276	15 534

为了验证结果的正确性，人工加入了一个原来没有的邮件正文分类，包含文本 312 篇。本文选取了其中的 40 篇作为特征文本，从中提取特征词。然后经过筛选进行测试。测试程序的正确性数据如表 5 所示，总记录数为 1 000 000。

表 5　不同阈值下的准确率和召回率

B-threshold	准确率	召回率
0.4	0.820	0.915
0.6	0.965	0.860
0.9	0.985	0.735

5　结束语

本文通过传统的关系型数据库存储过程实现了海量邮件数据主题和内容的数据挖掘，在百万条以上数据记录的情况下，与分词建立索引的方法比较优势明显，提高了快速分类的效率，解决了海量文本信息快速分类的难题，对文本挖掘的实现有相当大的参考价值。对邮件地址的关系挖掘是下一步的工作重点。

参考文献

[1] 谈竹贤，王　毅，赵景亮，等. Oracle9i PL/SQL 从入门到精通[M]. 北京：中国水利水电出版社, 2002.

[2] 徐宝文，张卫丰. 搜索引擎与信息获取技术[M]. 北京：清华大学出版社, 2003.

[3] 盛　骤，谢式千，潘承毅. 概率论与数理统计[M]. 北京：高等教育出版社, 2005.

[4] 毛国君，段立娟，王　实. 数据挖掘原理与算法[M]. 北京：清华大学出版社, 2006.

编辑　金胡考

（上接第 39 页）

的规则；如果 S=0.5，那么规则 $A\rightarrow B$, $A\rightarrow B$ 的兴趣度是一对相反数；否则，规则 $A\rightarrow B$, $A\rightarrow B$ 的兴趣度不一定相反。

(6)当规则、规则前提、规则结论三者的支持度都等于 S 时，规则的兴趣度最高，为 0.25；当规则、规则前提、规则结论三者的支持度分别等于 0, S, S 时，规则的兴趣度最低，为-0.25；当规则、规则前提、规则结论三者的支持度分别等于 0, 0, 0 或者 S, S, S 时，规则的兴趣度为 0。

4　结束语

本文对 PS 公式的数学特性进行了深入的讨论，指出了它的优点和不足，并在此基础上提出了一个新的度量规则兴趣度的方法。这种度量方法综合考虑了用户主观偏好、规则准确度、规则相关度对规则兴趣度的影响[8]，克服了支持度-可信度框架的缺陷，可以用来简化寻找令人感兴趣规则的过程，会对优化现有的关联规则挖掘算法起到很好的作用。

参考文献

[1] 朱　明. 数据挖掘[M]. 合肥：中国科技大学出版社, 2002.

[2] 王　军. 数据挖掘技术[M]. 北京：中国科学院计算研究所, 2000.

[3] 窦祥国，胡学刚. 关联规则的评价方法研究[J]. 安徽技术师范学院学报, 2005, 19(4): 44-47.

[4] 罗　可，吴　杰. 关联规则衡量标准的研究[J]. 控制与决策, 2003, 18(3): 277-280.

[5] Agrawal R, Imielinski T, Swami A. Mining Association Rules Between Sets of Items in Large Database[C]//Proc. of the ACM SIGMOD International Conference on Management of Data. New York, USA: ACM Press, 1993: 207-216.

[6] 马建庆，钟亦平，张世永. 基于兴趣度的关联规则挖掘算法[J]. 计算机工程, 2006, 32(17): 121-122.

[7] 伊卫国，卫金茂，王名扬. 挖掘有效的关联规则[J]. 计算机工程与科学, 2005, 27(7): 91-93.

[8] Tan Pangning, Steinbach M. 数据挖掘导论[M]. 范　明，范宏建，译. 郑州：郑州人民邮电出版社, 2006.

编辑　顾逸斐