

# 基于文本聚类 and 权重调整的用户兴趣建模算法

费洪晓 穆 珺 刘 正

(中南大学 信息科学与工程学院 湖南 长沙 410075)

**摘 要** 文本分类和文本聚类在信息过滤系统对用户兴趣进行学习的过程中,都具有很普遍的应用。文中对两者的工作原理进行了对比和分析,从根本上指出了文本分类作为有监督学习方法所存在的固有缺陷,提出了一种在文本聚类后根据词条与聚类的分布特征调整词条权重的方法,并设计和实现了一个基于文本聚类和权重调整的用户兴趣模型构造算法。

**关键词** 文本分类的固有缺陷;文本聚类;权重调整;用户兴趣模型

中图分类号:TP301.6;TP391

文献标识码:A

文章编号:1673-629X(2007)02-0128-03

## Study on User Profile Learning Algorithm Based on Document Clustering and Feature Weight Adjustment

FEI Hong-xiao MU Jun LIU Zheng

(Information Science and Engineering College of Central South University, Changsha 410075, China)

**Abstract** In process of learning the user interests for an information filtering system, document classification, as well as document clustering has been widely used. In this paper, discuss the principle of document classification and document clustering, and demonstrate that as supervised learning, document classification has shortage in basis. Also present a user profile learning algorithm based on document clustering and feature weight adjustment.

**Key words** shortage in basis of document classification; document clustering; feature weight adjustment; user profile

## 0 引言

传统的搜索引擎技术作为在含量巨大且质量参差不齐的网络信息汪洋中快速定位资源的一种方法,一度发挥了极其重要的作用。然而随着计算机网络应用的进一步普及和深入,传统搜索引擎基于关键字和资源匹配带来的通用性与用户个性化信息需求之间的矛盾日益暴露,人们对于网络信息获取技术提出了更高的需求<sup>[1]</sup>。由此,个性化信息过滤系统应运而生。个性化信息过滤系统通过收集和分析用户信息来建立用户兴趣模型,将搜索引擎返回的结果与用户兴趣模型进行匹配,从而过滤掉与用户兴趣无关的冗余信息,进一步减轻用户负担和提高用户获取信息的效率。而作为个性化过滤的依据和知识源,用户兴趣模型能否准确地捕捉和描述用户兴趣并及时跟踪反映其变

化,将直接决定过滤的成功与否。

## 1 文本分类方法存在的固有缺陷

追溯个性化过滤技术的发展,在早期阶段个性化过滤系统通常是服务于某些特定领域的应用需求而产生的。在这些应用系统雏形中,信息规模虽然很大,但所涉及的主题和领域往往很有限而且比较集中。这一特点使得向过滤系统中添加领域专家知识以提高分类精确度具有较高的可行性。因此,以诸如贝叶斯分类、决策树等文本分类方法为代表,有监督学习在得到有效指导(如利用专家知识确定文本特征项的选择、为兴趣主题提供有代表性的训练样本)的前提下表现出很好的效果,从而在信息过滤系统的兴趣建模型中得到了广泛的应用<sup>[2,3]</sup>。

然而随着信息过滤应用领域的扩大和用户兴趣需求的复杂化,人们发现即使不断有新的算法和设计被用来从各个环节改善基于文本分类的兴趣模型,从准确度和用户负担这两个方面综合考虑,基于文本分类的兴趣模型的缺陷仍总是不能被完全消除。若在分类算法中将兴趣主题的粒度设置得较大,则分类后同一

收稿日期:2006-05-18

基金项目:国家自然科学基金资助项目(60173041);湖南省自然科学基金资助项目(02JJY2094);湖北省科技计划项目(2006JT1040)

作者简介:费洪晓(1967-),男,浙江嵊县人,副教授,研究方向为网络管理与网络安全。

类中文本之间的相似度和耦合度较小,由此得到的过滤结果可能仍然不符合用户兴趣,但若将兴趣主题的粒度设置减小,则潜在的主题层次结构的深度和广度都将增加,用户为了让系统学习到符合自己兴趣的模型所要付出的代价将增大。

分析基于文本分类的信息过滤工作原理可以发现,在用于学习和更新兴趣模型的文本中,以词条在文本/文本集的分布情况为基本信息源,词条及其分布特征经选择后形成文本表示模型,再由分类算法进行分类。这是一个表示模型简化精练的过程,由原始信息得到知识表示,由此带来的信息损耗不可避免。问题在于系统试图在所有领域、所有用户普遍适用的前提下建立分类框架,而这其实是不可能实现的。一个兴趣广泛的用户和一个兴趣专精的用户,需要的分类框架是不同的,即使同一个用户,在兴趣逐渐变化了的时候,既定的分类框架也是不能适用的。

这正是分类算法作为有监督学习的固有缺陷之所在,信息须在既定框架的指导下被分类和处理,形成知识表示,这和信息自发聚合处理形成知识表示相比,多出了一个使信息损耗的过程,而在分类框架和原始信息分布不匹配的情况下损耗尤其严重。而使用无监督的聚类方法,则聚类结果已经自动包含了分类框架,并且不存在既定分类方法带来的精度损失的问题。

## 2 特征选择算法的确定及讨论

在使用文本分类学习方法的过滤系统中,文本表示阶段对文本特征项的选择有很多方法,如  $TF * IDF$  法、信息增益、信息熵、互信息等<sup>[4,5]</sup>。除  $TF * IDF$  法所使用的词条频度因子(Term Frequency)和反向文档频度因子(Inverse Document Frequency)可以直接从文本、文本集统计得到,其他的特征项选择方法都考虑了词条对于每个类别间的区分作用,而聚类算法作为无监督学习,其文本表示阶段是不适于引入分类信息的。因此主要考虑使用  $TF * IDF$  法来选择特征词条,得到文本表示。

根据  $TF * IDF$  方法,词条的权重被定义为  $TF * IDF$ ,其中,词条频度因子  $TF$  为词条在文本中出现的频度,反向文档频度因子  $IDF = \log(|D|/DF(W))$ , $|D|$  代表文本集中文本总数, $DF(W)$  代表出现了  $W$  的文本数。 $TF * IDF$  法的思想是基于这样一个假设:对于某个特定词条,在整个文本集中包含该词条的文本数量越多,该词条越普遍,越缺乏区分能力,所以权重相应地减小。由于该方法具有一定的抑制冗余信息的作用,同时又易于实现,所以在信息检索和信息过滤领域中得到非常广泛的使用。

但值得注意的是, $TF * IDF$  法虽然有一定的抑制冗余信息的作用,但是它所作的“包含该词条的文本越多该词条的区分能力就越小”的假设也不完全正确。譬如,假设用户的兴趣集中在两个主题:“音乐”和“体育”,而包含了“音乐”这一词条的文本在所有文本中所占比例约为 50%,并且集中分布在同一类文本中,如果同时有某个词条均匀地分布在各类文本中且所占比例恰巧也为 50%,那么按照  $TF * IDF$  方法的评价,这两个词条的权重都将被做同样程度的削减,但显然“音乐”这一词条对它所在类而言是有代表性的,其权重应当远高于在文本集比例相同但分布均匀的另一个词条。这说明  $TF * IDF$  法,由于在文本表示阶段没有将词条和类之间关系加以考虑进行权重评价,所以对聚类结果的表示存在一定程度的失真,因此需要在聚类之后根据词条在类中/类间的分布情况对词条的权重进行调整。

## 3 文本聚类算法的设计

在文本聚类领域, $K - Means$  算法以其效率优势得到了最为广泛的应用<sup>[6]</sup>。但  $K - Means$  算法存在两个问题:第一个问题是该算法需要输入参数  $K$  来确定聚类结果中簇的数目,如果该参数由用户指定,则既增加了用户负担,也违背了选择无监督的学习方法以求最大程度地维持数据分布原貌的初衷。第二个问题是  $K - Means$  算法对孤立点比较敏感,孤立点的存在对聚类准确性有较大的影响。

针对第一个问题,有针对性地设计一个改进的凝聚的层次聚类算法对原始数据进行初始聚类。改进的凝聚的层次聚类算法描述如下:

输入:文本集合

输出:特定精度的聚类簇

处理:

- 1) 计算各文本之间的相似度,形成相似度矩阵;
- 2) Repeat;
- 3) 合并两个距离最近的文本;
- 4) 修改相似度矩阵;
- 5) Until 达到结束条件。

其中结束条件为:当前距离最近的文本距离/第一次合并的文本间距离  $\leq \alpha$  ( $\alpha$  为一个设定的阈值)。

该方法设置了精度比达到指定值时停止凝聚的层次聚类过程,这样没有引入附加知识的干预,能够得到反映数据自身分布特点的初始聚类结果。以此作为  $K - Means$  算法的输入,既确定了参数  $K$ ,又很好地解决了  $K - Means$  算法初始中心向量的选择的问题。

针对  $K - Means$  算法存在的第二个问题即孤立点

的存在影响聚类准确性,在  $K - \text{Means}$  算法的每一次进行划分之后,先取得  $K$  个簇的聚类结果,根据文本与所在簇相似度对聚类结果进行排序,将所有相似度小于特定值的文本视为孤立点从其所所在簇中删除。算法描述如下:

输入:簇的数目  $K$  和  $K$  个簇的中心向量

输出:  $K$  个簇

处理:

- 1) Repeat;
- 2) 将每个文本划分到最类似的簇中;
- 3) 去除孤立点;
- 4) 更新每个簇的中心向量;
- 5) 计算准则函数;
- 6) Until 准则函数收敛。

#### 4 根据聚类结果进行特征项权重调整

在前面已经讨论过,为消除先验知识的干预,在文本表示阶段没有将词条和类之间关系加以考虑进行权重评价,因此得到的聚类结果中,特征词条的权重在体现其类型分布特点上存在一定程度的失真,有必要根据聚类结果对词条的权重重新进行调整和计算。

Shrikanth Shankar 和 George Karypis 在文献 [7] 中也讨论了文本分类之后根据分类结果对词条权重进行调整的问题。在该文献中,作者定义了“纯度”的概念用来描述特征词条对类的区分能力。对于一个含有  $m$  个类的文本集合,  $m$  个类的质心向量构成的矩阵为  $\{C_1, C_2, \dots, C_m\}$ , 对每个特征词条  $i$ , 取  $m$  个质心向量中的词条  $i$  对应的权重构成向量  $T_i$ , 再对  $T_i$  进行规范化处理得到向量  $T_i'$ , 则第  $i$  个词条的纯度为:  $P_i = \sum_{j=1}^m (T_{j,i})$ 。由于  $P_i$  的取值范围在  $[1/m, 1]$  之间, 且当词条在每个类的质心中权重相同时  $P_i$  取得最小值, 当词条仅在一个类的质心中存在时  $P_i$  取得最大值, 因此可以由纯度得到词条在类中分布的大概规律。但因纯度算法中仅仅以类的质心向量来提供词条信息, 故词条在类内部的分布特征没有得到充分体现。

综合考虑词条在类内和类间的分布特征, 文中设计了一个新的权重调整算法, 用来对文本聚类后得到的中心向量所包含词条的权重进行调整。调整后的词条权重  $\text{adjusted\_weight}$  由下式计算得到:

$$\text{adjusted\_weight}_{i,j} = \text{weight}_{i,j} * \text{size}_i * \text{scale}_{i,j} * E_j$$

其中,  $i \in \{1, 2, \dots, m\}$ ,  $m$  为文本聚类所得到的簇的数目; 而  $j \in \{1, 2, \dots, n\}$ ,  $n$  为各个簇的中心向量中包含的所有特征词条构成的统一的向量空间的维度。 $\text{adjusted\_weight}_{i,j}$  表示对第  $i$  个簇对应的中心向量的

第  $j$  维词条进行权重调整后得到的新权值。 $\text{weight}_{i,j}$  表示第  $i$  个簇对应的中心向量的第  $j$  维词条的初始权重, 由于中心向量中特征词条的初始权重是由簇中文本向量的均值得到, 因此只能反映出该词条在文本内的重要程度。 $\text{size}_i$  表示第  $i$  个簇中包含的文本总数。 $\text{scale}_{i,j}$  表示在第  $i$  个簇中包含词条  $j$  的文本在该簇文本总数中所占比例。 $E_j$  表示先将  $\text{scale}_{1,j}, \text{scale}_{2,j}, \dots, \text{scale}_{m,j}$  这  $m$  个比例值进行一规范化, 再对规范化后的比例值所求得的方差。

下面对该计算方法的数学意义和效果进行分析: 考虑统一中心向量空间中的某一个特征词条  $j$ , 由该词条在各簇中所占比例  $\text{scale}_{1,j}, \text{scale}_{2,j}, \dots, \text{scale}_{m,j}$  可以得到表示簇间分布均匀程度的方差  $E_j$ , 对于一个确定的词条  $j$  而言,  $E_j$  是一个确定值。此时, 对于不同的簇而言, 如果  $\text{size}_i$  值越大, 该簇中心向量中词条  $j$  调整后得到的权重就越高, 这是因为  $\text{size}_i$  值表示该簇包含文本数目, 该簇包含文本数目多, 则暗示用户对该簇所包含的主题越感兴趣; 如果  $\text{scale}_{i,j}$  值越大, 该簇中心向量中词条  $j$  调整后得到的权重也越高, 这是因为在其他条件相同的情况下, 词条  $j$  在出现比例高的簇中的重要程度必定高于出现比例低的簇。

再考虑在确定的簇  $i$  中, 比较不同特征词条  $j$  的权重。假设在该簇中选取到这样两个不同的特征词条  $j_1$  和  $j_2$ ,  $j_1$  和  $j_2$  具有类似的初始权重  $\text{weight}_{i,j}$ , 两者在该簇中出现的比例也类似, 由于在同一个簇中, 两者的簇规模  $\text{size}_i$  值也必然相同。此时这两个特征词条之间的差异就主要由方差  $E_j$  来体现: 如果词条  $j_1$  在每个簇中的出现比例分布差异很大, 则说明该词条对于标识聚类之间的差异很有作用, 由方差的数学意义可知, 该词条对应的  $E_{j_1}$  值也很大, 调整后得到的权重也很高; 反之, 如果词条  $j_1$  在每个簇中的出现比例分布非常均匀, 则说明该词条很有可能是没有什么内容的词语, 对于标识聚类之间的差异也没有太大作用, 通过方差计算, 该词条对应的  $E_{j_1}$  值必定很小, 权重调整得到的权重相应就很低。假设一个极端的情况, 譬如某词条在所有簇中都以相同的比例出现, 那么方差计算的结果就会是零, 则该词条在簇中通过权重调整之后完全被当作冗余信息滤除掉了。

#### 5 结束语

文中对用户兴趣建模中常用的文本分类方法与文本聚类方法的工作原理进行了对比和分析, 从根本上分析了文本分类作为有监督学习方法所存在的固有缺陷, 提出了一种在文本聚类后根据词条与聚类的分布



if  $v$  is odd then  $\{v \leftarrow v - 2 ; R \leftarrow R - 2Q ;\}$

else  $\{v \leftarrow v - 1 ; R \leftarrow R - Q ;\}$

return  $R ;$

其中  $u[i-1 \rightarrow i-w+1] || 1$  表示取  $u$  的第  $i-1$  至  $i-w+1$  位组成位串与 1 组成新的位串  $[2^i]R$  表示对  $R$  进行  $i$  次倍点运算。算法 2 首先置标量  $u, v$  为奇数, 而对奇数标量  $u, v$  按照类似 NAF 算法转换成  $0 \dots 0x \dots 0 \dots 0x$  的形式, 其中  $x$  为奇数, 且  $x \in \{\pm 1, \pm 3 \dots, \pm 2^{w-1} - 1\}$ , 具体方法: 如果  $u[i] = 0$  则  $t_1 \leftarrow t_1 - 2^w$ , 如果  $u[i] = 1$  则  $t = u[i-1 \rightarrow i-w+1] || 1$ , 其中  $a || b$  表示将  $a$  和  $b$  连接, 这样保证转换后的结果成为  $0 \dots 0x \dots 0 \dots 0x$  的形式<sup>[5]</sup>。

## 2.2 性能分析

### (1) 内存空间需求。

由于  $x$  为奇数, 且  $x \in \{\pm 1, \pm 3 \dots, \pm 2^{w-1} - 1\}$ , 则需要预计算  $3P, 5P, \dots, (2^{w-1} - 1)P, 3Q, 5Q, \dots, (2^{w-1} - 1)Q$ 。因此需要 2 次倍点运算和  $2^{w-1} - 2$  次点加运算, 需要预存储  $2^{w-1}$  个点。

### (2) 非零密度。

标量在进行点乘运算时转换成了  $0 \dots 0x \dots, 0 \dots 0x$  形式, 其中  $0 \dots 0x$  为连续  $w-1$  个 0 和 1 个奇数  $x$ , 故非零密度为  $1/w$ , 因此新的算法主计算阶段需要进行  $n+1$  次倍点运算和  $2n/w$  次点加运算。

算法 1 由于在主计算之前必须分别将标量  $u, v$  转换成其相应的  $w$ NAF 表示形式, 而  $w$ NAF 表示形式必须从右到左进行计算(即从标量的最底位向最高位进行), 故需要首先计算并存储标量的 NAF 表示形式。而由于新的算法对标量编码是从最左到右进行, 因此新算法的编码阶段和主计算阶段合并在一起, 不需要存储标量  $u$  和  $v$  的新的编码, 这样可以节省存储标量  $u, v$  的 NAF 表示形式的编码, 故可以节省内存空间, 这对于内存空间受限的设备来说尤其有益。

### (3) 安全性分析。

攻击者虽然可以通过测试电量的消耗来区分点加和倍点运算, 但由于新提出的算法通过采用固定模式  $0 \dots 0x \dots 0 \dots 0x$  对标量进行处理, 始终得到相同的序

列  $D \dots DA \mid D \dots DA \mid \dots \mid D \dots DA$ , 其中  $D$  表示倍点运算,  $A$  表示点加运算, 因此通过 SPA 攻击得不到秘密  $u$  和  $v$ 。故新的算法是抗 SPA 攻击的。

## 3 结束语

文中, 在内存空间和计算时间负担增加不多的情况下, 基于 interleaving 多点乘算法, 提出了一个新的抗 SPA 的多点乘算法。虽然本算法是抗 SPA 的多点乘算法, 是针对多点乘运算的, 但通过同构等方法, 本算法也同样适用于安全的点乘运算。

### 参考文献:

- [1] Kocher P, Jaffe J, Jun B. Introduction to Differential Power Analysis and Related Attacks[EB/OL]. 1998. URL: <http://www.cryptography.com/dpa/technical/index.html>.
- [2] Kocher P, Jaffe J, Jun B. Differential Power Analysis[C]//In Proceedings of CRYPTO '99, LNCS vol 1666. [s. l.]: Springer-Verlag, 1999: 388-397.
- [3] Coron J S. Resistance Against Differential Power Analysis for Elliptic Curve Cryptosystems[C]//In Proceedings of CHES '99, LNCS vol 1717. [s. l.]: Springer-Verlag, 1999: 292-302.
- [4] Montgomery P L. Speeding the Pollard and Elliptic Curve Methods for Factorization[J]. Mathematics of Computation, 1987, 48: 243-264.
- [5] Okeya K, Takagi T, Vuillaume C. On the Exact Flexibility of the Flexible Countermeasure against Side Channel Attacks[C]//In The 9th australasian conference on information security and privacy, ACISP 2004, LNCS vol 3108. [s. l.]: Springer-Verlag, 2004: 466-477.
- [6] Okeya K, Takagi T. The Width- $w$  NAF Method Provides Small Memory and Fast Elliptic Scalar Multiplications Secure against Side Channel Attacks[J]. IEICE Transactions, 2004, E87-A: 75-84.
- [7] Lee Mun-Kyu. SPA-Resistant Simultaneous Scalar Multiplication[C]//In Approaches or Methods of Security Engineering Workshop, LNCS vol 3481. [s. l.]: Springer-Verlag, 2005: 314-321.

(上接第 235 页)

### 参考文献:

- [1] 张剡, 夏辉, 柏文阳. 数据库安全模型的研究[J]. 计算机科学, 2004, 31(10): 101-103.
- [2] Li-Yan Yuan. The Documentation of LogicSQL[M]. Canada: Alberta University, 2005.
- [3] National Computer Security Center. A guide to understanding covert channel analysis of trusted systems[R]. NCSC-TG-030. [s. l.]: [s. n.], 1993.
- [4] 张敏, 徐震, 冯登国. 数据库安全[M]. 北京: 科学出版社, 2005.
- [5] 卿斯汉, 刘文清, 温红子. 操作系统安全[M]. 北京: 清华大学出版社, 2004.
- [6] 卿斯汉. 高安全等级安全操作系统的隐蔽通道分析[J]. 软件学报, 2004, 15(12): 1837-1849.