进化式信息过滤方法研究*

田范江 李丛蓉 王鼎兴

(清华大学计算机科学与技术系 北京 100084)

E-mail: tfj@mail.cs.tsinghua.edu.cn

摘要 进化式信息过滤方法从多个角度描述用户的信息需求,通过类似自然选择的过程,达到系统整体过滤性能的优化.同时还从面向对象程序设计语言的设计思想中获得启发,引入了继承、类树的概念,增加了过滤系统的易用性.这种方法可以缩短训练时间,提高过滤质量,同时减小过滤结果与训练次序的相关性.

关键词 信息过滤,信息处理.

中图法分类号 TP393

随着因特网上信息量的迅速增加,信息过滤(information filtering)技术正得到越来越广泛的关注.信息过滤系统根据用户的信息需求对动态信息流进行过滤,仅把满足用户需求的文档传送给用户,可以提高获取信息的效率.对信息过滤最主要的需求是对文档与用户信息需求相关性的判断要准确,同时训练时间应尽可能短.在SIFT[1]等传统信息过滤系统中,用户手工输入关键词生成信息需求定义文件(profile),并提供相关反馈进行优化.这些系统的缺点是,过滤结果与起始状态及训练文档次序相关性很大.Amalthaea系统[2]引入自然选择的概念用于个人信息过滤,取得了较好的结果,但训练过程太长,而且是在用户端实现,无法实现离线操作,也无法借鉴其他用户的浏览经验.清华大学计算机科学与技术系设计的信息收集和服务系统TH-InfoBase引入了自然选择、突变、杂交、移植等概念,在服务器端实现了进化式信息过滤.该系统可提供针对大量用户的离线个人信息过滤服务,实验结果表明,该系统中采用的进化式信息过滤方法有效地缩短了训练时间,并提高了准确度.

1 进化式信息过滤

1.1 进化式信息过滤器概述

传统的信息过滤器都是定义profile,然后通过一定的学习方法进行学习,本文的进化式信息过滤器引入自然选择的概念,并借鉴了面向对象程序设计中类树及继承的思想.

简单来说就是,首先由系统管理员定义类的框架,并训练出基本的信息需求定义文件,用户通过多种方式根据系统中已有的信息需求定义文件,迅速得到一个较高质量的信息需求定义文件.同时,引入进化式信息需求定义文件(E-Profile),每个E-Profile中包含多个信息需求的描述,它们从不同角度反映用户的信息需求,它们之间互相竞争又互相合作,在竞争中失败的个体将被淘汰,优秀的个体可以繁殖后代.这样,通过多个个体的竞争和合作,经过一个类似于生物界中自然选择的过程,使系统性能达到最优.

1.2 进化式信息过滤器结构简图

在图1中,菱形框表示数据实体,方框表示行为实体.

* 本文研究得到国家863高科技项目基金(No.863-306-ZT01-03-1)和IBM中国研究中心基金资助.作者田范江,1974年生,博士生,主要研究领域为WWW应用系统,信息采集,信息检索与过滤.李丛蓉,1975年生,硕士生,主要研究领域为信息检索和过滤.王鼎兴,1937年生,教授,博士生导师,主要研究领域为并行分布计算系统,WWW应用系统.

本文通讯联系人:田范江,北京 100084,清华大学计算机科学与技术系

本文1998-12-07收到原稿,1999-03-30收到修改稿

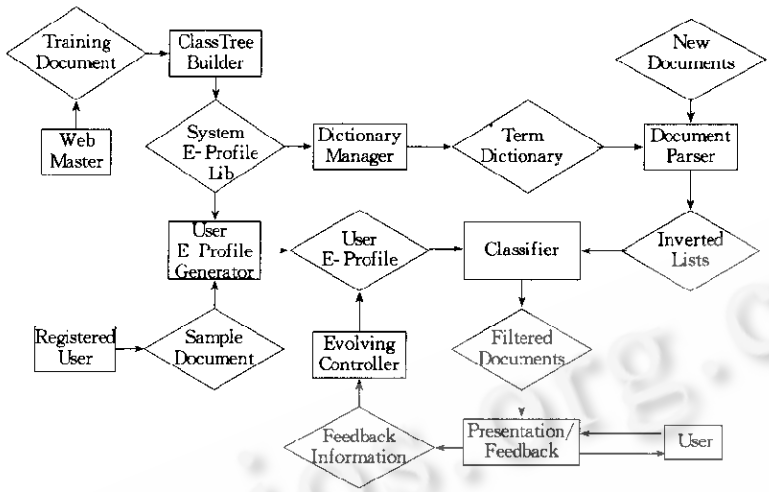

Fig. 1 Evolving information filter architecture
图1 进化式信息过滤器结构简图

1.3 概念及数据实体说明

1.3.1 进化式用户信息需求定义文件(E-Profile)(见表1)

Table 1 E-Profile

表1 E-Profile

Parent	T-Num	B-ProfileNum
B-Profile 1		Credit
B-Profile 2		Credit
.....		
B-Profile n		Credit

在传统信息过滤系统中,用户信息需求定义文件(profile)大多是一个包含若干项(T)及其权重(T_w)的序列,其形式为$\{(T_1,W_1)(T_2,W_2)\ldots(T_n,W_n)\}$.本文将profile扩展为E-Profile(evolving-profile),为清晰起见,传统形式的profile被称作基本profile(B-Profile).E-Profile的首部是属性域,主要包括指向其父节点的Parent指针、训练次数T-Num和B-Profile个数等.其后是多个B-Profile及该B-Profile的信任度Credit.

为了便于管理,所有由站点管理者(Web master)创建的E-Profile构成一个树形结构的系统E-Profile库(system E-profile lib),而所有由用户创建的E-Profile则保存在User E-Profile Lib中.

1.3.2 倒置列表(inverted lists)、特征项词典(term dictionary)

倒置列表包含最能反映该文档特征的特征项及其权重集合,形式与B-Profile相同.为降低计算量及存储空间,在生成倒置列表时系统并不是将所有出现的词都作为特征项,只有在E-Profile中出现的项才可能作为特征项,特征项词典根据System E-Profile Lib和User E-Profile Lib生成,作为分析文档时的依据.

1.3.3 文档(document)、过滤后的文档集(filtered documents)

文档是信息过滤系统处理的基本单位,一个文档就是一个具有相对独立意义的自然语言片断.过滤后的文档集是所有文档按照与用户信息需求相关的程度排序后,按照相关程度阈值截断的文档集合,形式为$\{(D_1,S_1)(D_2,S_2)\ldots(D_n,S_n)\}$.

1.3.4 反馈信息(feedback information)

用户可以对过滤后文档集中的文档按照与其信息需求的相关程度打分,分数分为5个级别,最低为0,最高为1,反馈信息形式为$\{(D_1,S_1)(D_i,S_i)\ldots(D_m,S_m)\}(m\leqslant n)$.

1.4 行为实体作用及算法说明

1.4.1 类树建造器(classtree builder)

在系统初始化时,WebMaster 使用类树建造器构建作为系统基础的系统 E-Profile 库.系统 E-Profile 库采用树形结构,最顶层是包含所有文档的 World 类,一篇文档可以被归入一个或多个类,不能归入任何类的都被归入"其他(others)"类.

类别个数、层次结构及每个类别的 E-Profile 使用标注过类别的训练文档集合进行训练,每个类别的 E-Profile根据被标注为该类别的训练文档生成.生成方法是,设类别个数为 m,类别集合 $C_{set}=\{C_1,C_2,\ldots,C_m\}$;训练文档数为 n,训练文档集合 $D_{set}=\{D_1,D_2,\ldots,D_n\}$,其中 $D_i=\{V_i;S_{i1},S_{i2},\ldots,S_{im}\}$,$V_i$ 是文档 D_i 的向量表示,S_{ij}是文档 D_i 与类 C_j 的相关程度,则类 C_j 的 B-Profile(P_j)可以用下式得到:

$$P_j=\sum_{i=1}^{n}(S_{ij}-\zeta)V_i \quad (0\leqslant\zeta\leqslant 1).$$

上式的含义是,属于类 C_j 的文档必须与分值高于 ξ 的文档相似,而与分值低于 ξ 的文档不相似.在经过上述处理的 Profile 中可能包含很多项,截取权重最高的 N 项构成最终的 B-Profile,N 是一个系统参数.取 ξ 的不同值,可以得到多个不同的 B-Profile,它们共同构成类 C_j 的 E-Profile.

1.4.2 用户信息定义文件生成器(user E-profile generator)

系统 E-Profile 库建立后,用户可以浏览类的结构及各类的典型文档,选择与自己的需求最接近的类,并在此类 E-Profile 的基础上生成自己的 E-Profile.生成的方法有

(1) 移植(transplant).记为 $P_1\rightarrow P_2$,实际上就是完全复制一个 E-Profile.

(2) 杂交(crossbreeding).记为 $P_1\times P_2;P_3,P_4$,根据已有的两个信息需求定义文件(P_1,P_2),生成两个新的信息需求定义文件(P_3,P_4),以如下方式实现:

首先得到两个交叉点:

$$n_1=rand(1,sizeof(P)-2),\qquad n_2=rand(n_1,sizeof(P)-1).$$

然后,P_3,P_4 分别从 P_1,P_2 继承:

$$p_3=\begin{cases}p_1 & 0<i<n_1,n_2<i\leqslant sizeof(P)-1\\ p_2 & n_1\leqslant i\leqslant n_2\end{cases},\qquad p_4=\begin{cases}p_1 & n_1\leqslant i\leqslant n_2\\ p_2 & 0<i<n_1,n_2<i\leqslant sizeof(P)-1\end{cases}$$

(3) 嫁接(inoculation).记为 $P_1\oplus D_i;P_2$,根据已有的信息需求定义文件(P_1)和示例文档集合(D_i),生成新的信息需求定义文件(P_2),以如下方式实现:

$$p_2=p_1+\sum_{i=1}^{n}(S_{ij}-\zeta)V_i \quad (0\leqslant\zeta\leqslant 1),$$

其中 $n=sizeof(D_i)$,训练文档集合 $D_i=\{D_1,D_2,\ldots,D_n\}$,$D_i=\{V_i;S_{i1},S_{i2},\ldots,S_{im}\}$,$V_i$ 是文档 D_i 的向量表示,S_{ij}是文档 D_i 与类 C_j 的相关程度.

通过以上 3 种操作,用户在注册时可以很方便地根据系统中已有的类别及自己提供的示例文档得到自己的初始 E-Profile.

1.4.3 词典管理器(dictionary manager)

词典管理器首先根据一个计算机术语词表生成特征项词典,并根据系统 E-Profile 库和用户的 E-Profile 不断扩充该词典.另外也负责同义词、词根化等预处理工作.

1.4.4 文档分析器(document parser)

文档分析器负责在不遗漏关键信息的同时把文档转换成便于高效处理的结构,文档使用向量空间模型表示.关键词 T_k 在文档 i 中的权重 W_{ik} 计算如下:

$$W_{ik}=T_{ik}\times\log((N+0.5)/n_k)/\log(N+1.0),$$

T_{ik}是文档 i 中关键词 T_k 的出现次数,$\log(N/n_k)$是训练文档集合中项 T_k 的倒置文档频率;N 是训练文档集合中文档的总数;n_k 是训练文档集合中包含项 T_k 的文档数.

1.4.5 比较器(classifier)

过滤时,E-Profile 中的每个 B-Profile 都与文档进行比较,并得到一个相似度,计算相似度的方法是余弦相似度[3]. 设 B-Profile 为 $P=[p_1,p_2,\ldots,p_t]^T$,文档向量为 $D=[d_1,d_2,\ldots,d_t]^T$,其中 t 是词典中所有的特征项的个数,向量中的每个单元是权重,它们的相似度可以用以下公式计算.

所有文档向量为 0 的文档全部被归入"其他"类中,不需要计算相似程度.

$$s=\Big(\sum_{i=1}^{t}p_id_i\Big)\Big/\sqrt{\Big(\sum_{i=1}^{t}p_i^2\Big)\Big(\sum_{i=1}^{t}d_i^2\Big)}.$$

设在一个 E-Profile 中共有 n 个 B-Profile,那么对任意文档 d,将得到由 n 个相似度组成的相似度向量 $S=[S_1,S_2,\ldots,S_n]^T$,同时,E-Profile 中 n 个 B-Profile 的信任度(credit)也构成一个信任度向量 $C=[C_1,C_2,\ldots,C_n]^T$,文档的最终相似度向量 $S'=C^TS$,对于一批文档 $D_{set}=\{D_1,D_2,\ldots,D_j\}$,根据相似度,$S$ 可以得到每个 B-Profile的文档排序,根据最终相似度,S'可以得到该 E-Profile 的文档排序.

1.4.6 用户界面与反馈(representation/feedback)

过滤后的文档按照最终相似度排序后提供给用户. 用户首先见到的是系统判断与用户信息需求相似度最高的前 N 个文档,N 值可以由用户指定. 用户可以对见到的文档作出评价,评价分为 5 级.

1.4.7 进化控制器(evolving controller)

(1) 隐式学习策略

B-Profile 采用激励(reinforcement)算法[4]进行学习:

$$p'_i=p_i+\tau\times\sum_{j=1}^{N}(S_{ij}-\zeta)V_j\quad(0<\tau<1,0\leqslant\zeta\leqslant1),$$

其中 S_{ij}是第 i 个 B-Profile 对文档 d_j 的评分,V_j 是文档 d_j 的向量表示,ξ 是我们的准确度的期望,τ 是一个被精确选择的步长. 通过上式,每个 B-Profile 在每次反馈后都向用户的实际信息需求发生一次小的移动,从而达到提高过滤准确度的目的.

当同时存在多个 B-Profile 时,有可能出现一个 B-Profile 认为相关度很高的文档全部不能被系统选中的情况,在 Amalthaea 系统中,这意味着该过滤器永远无法提高其分值. 上面的方法可以称为隐式学习,所有的 B-Profile 有同等的学习机会,避免了这个问题.

(2) 奖惩策略

所有 B-Profile 的信任度被初始化为 $1/n$,n 是 E-Profile 中 B-Profile 的个数. 这样,在系统初始时,所有的 B-Profile具有相同的机会. 在进行过滤并得到反馈信息后,它们的信任度被更新. 更新的基本原则是,如果非常肯定却被用户否决或否决却被用户肯定,那么信任度降低;如果与被否决的一致或类似或者是与被肯定的一致或类似则信任度提高. 公式为

$$C'_i=C_i+\frac{1-\sqrt{\sum_{j=1}^{N}(S_{ij}-D_j)^2}}{\max\limits_{i=1,n}\Big(\sqrt{\sum_{j=1}^{N}(S_{ij}-D_j)^2}\Big)}.$$

计算过所有 B-Profile 更新后的信任度以后,将所有信任度映射到[0,1]区间:

$$C''_i=C'_i/\max_{i=1,n}(C'_i).$$

(3) 淘汰策略

用户提供的反馈次数达到一定值后,系统通过杂交操作进行更新. 方法是:首先选择信任度(credit)最高的两个 B-Profile 作为祖先,这是因为只有表现好的 B-Profile 才有繁殖后代的资格. 根据杂交操作得到两个新的 B-Profile 后,寻找两个表现不佳的 B-Profile,替代它们. 选择的依据是生命期信任度 C':

$$C'=C\times\tau(F_{max}-F_n+N)\quad0<\tau<1,$$

其中 C 是信任度;F_{max}是从当前 E-Profile 的各个 B-Profile 中得到的最多反馈次数;F_n 是该 profile 得到的反馈次数;N 是一次提交给用户的文档数,即一次反馈的最小单位;τ 是系统参数. 引入生命期信任度是基于以下假

设:即经过多次反馈和很少几次反馈达到相同的信任度的 B-Profile 质量是不同的,越"年轻"的越有前途.

2 实验结果

2.1 数据集合与评价指标

实验使用的数据集合是 500 个 HTML 文档.用 10 种不同的信息需求代表 10 个用户,对 500 个文档进行人工标注,每个文档与 10 个信息需求的相关程度记为$\{S_1,S_2,\ldots,S_{10}\}$.评价准确度的指标是正规化的精度和查全率(normalized precision and recall)[3].

$$Precision_{norm}=1-\frac{\sum_{i=1}^{REL}\log RANK_i-\sum_{i=1}^{REL}\log i}{\log(N!/(N-REL)!\,REL!)},\quad REcall_{norm}=1-\frac{\sum_{i=1}^{REL}RANK_i-\sum_{i=1}^{REL}i}{REL(N-REL)},$$

其中 N 表示过滤集合中的文档总数,REL 表示相关文档的总数.

精度值越高表明选择出的文档与用户的实际信息需求越相关,查全率值越高表明遗漏的有用文档越少,理想情况是两个值同时取得较高值.

学习速度用系统过滤性能稳定地达到用户满意程度时的文档总数来衡量,学习速度没有严格的定义,后文将通过数据和图示直观地说明在学习速度方面的性能.

为了研究过滤结果与训练次序之间的关系,现在引入变形均方差指标 $Diff$ 来衡量两次过滤结果之间的差异.设过滤集合为 $D_{set}=\{D_m,D_{m+1},\ldots,D_{m+n}\}$,两次对该过滤集合的相关度判断分别为$(S_m,S_{m+1},\ldots,S_{m+n})$和$(S'_m,S'_{m+1},\ldots,S'_{m+n})$,则 $Diff$ 为

$$Diff=\sqrt{\Big/\Big(\sum_{i=m}^{m+n}(S_i-S'_i)^2\Big)\Big/n}.$$

$Diff$ 值越大表明过滤结果差异越大,在我们的实验中,则表明过滤器过滤效果与训练次序的关系越大.因此,$Diff$ 越小越好.

2.2 对照实验设计

实验对 3 种过滤器进行比较:

(1) 进化式过滤器.每个 E-Profile 有 4 个 B-Profile,每获得 20 个文档的反馈信息进行一次淘汰,每次更新两个 B-Profile.

(2) 自学习的基本过滤器.每个 E-Profile 包含 1 个 B-Profile,只学习,不淘汰.

(3) 无学习的基本过滤器.每个 E-Profile 中包含 1 个 B-Profile,不学习,不淘汰.

为测试过滤结果与训练次序的关系,使用随机生成的两组文档次序进行对照实验.

2.3 实验结果

2.3.1 学习速度与过滤质量

从图 2 和图 3 可以看出,进化式信息过滤器学习快,过滤质量(包括精度和查全率)都比基本过滤器的过滤效果要好.在实验过程中我们发现,在部分文档和个别过滤器中,进化式信息过滤器的过滤有波动,这是由淘汰操作所引起的.但就整体而言,进化式信息过滤器的性能是很优秀的,即使出现波动,其性能也比另外两种方法要好,其影响仅仅是表现在不稳定性上.当然,研究更加稳定的淘汰策略是我们进一步的工作之一.

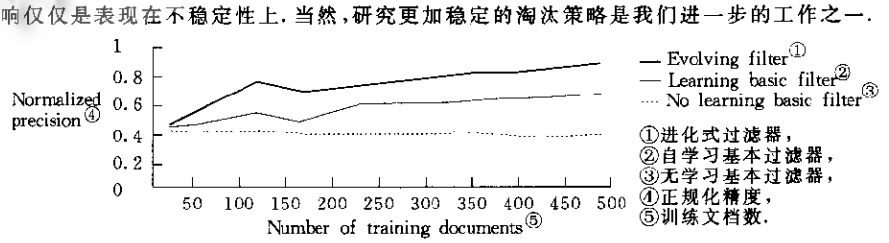

Fig. 2 Comparison of the precision of three filters

图2 3种过滤器过滤精度的比较

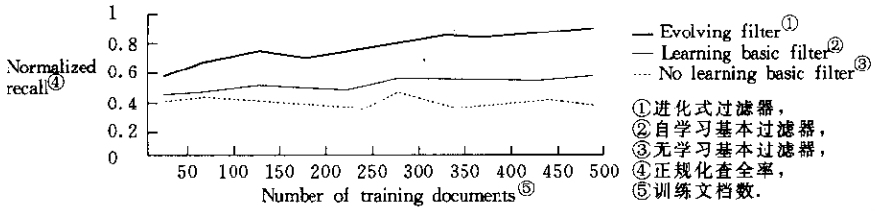

Fig. 3 Comparison of the recall of three filters
图3 3种过滤器查全率的比较

2.3.2 与训练次序无关性

图4显示进化式过滤器在与训练次序无关性方面表现较好,而且值得注意的是,从一开始其表现就比另外两种过滤器要好得多.因为每个E-Profile本身就包含多个B-Profile,也就是说,每个E-Profile本身就试图从多种角度来对文档进行判断,所以表现比较出色.

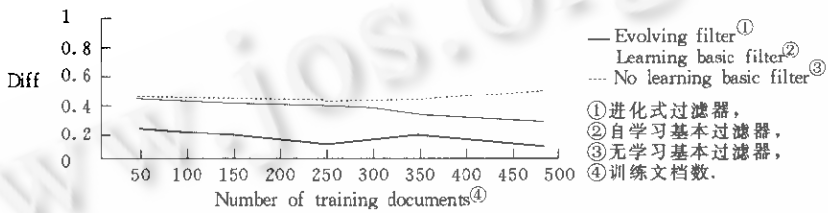

Fig. 4 Comparison of three filters' diff value (diff is the difference between filtering results caused by different training sequences)
图4 3种过滤器diff值的比较(diff是不同训练序列导致的过滤结果之间的差异)

3 结 论

本文介绍了一种进化式信息过滤方法,该方法在分类方案生成、结果合并、过滤器训练等方面都作出了新的尝试.初步的实验结果表明,该方法具有过滤准确、训练时间短、与训练次序无关的优点,适合于高准确度的定制化信息服务系统.

参考文献

1 Yan T Y, Garcia-Molina H. Sift——a tool for wide-area information dissemination. In: USENIX Association ed. Proceedings of the 1995 USENIX Technical Conference. Berkeley, CA: USENIX Association, 1995. 177~186
2 Moukas A. Amalthaea: information discovery and filtering using a multiagent evolving ecosystem. Applied Artificial Intelligence, 1997,11(5):437~457
3 Salton G, McGill M J. Introduction to Modern Information Retrieval. New York: McGraw-Hill, 1983
4 Narendra K S, Thathachar M A L. Learning Automatic——An Introduction. Englewood Cliffs, NJ: Prentice-Hall, 1989

Evolving Information Filtering Method

TIAN Fan-jiang LI Cong-rong WANG Ding-xing

(*Department of Computer Science and Technology Tsinghua University Beijing* 100084)

Abstract Evolving information filtering method presented in this paper describes user's information needs from multi-aspects simultaneously, and improves system filtering performance through a process like natural selection. Inspired by object-oriented programming language, this method also introduces concepts such as inherit and classtree, which make filtering systems easy to use. This method can shorten training time, improve filtering precision, and reduce the relevance between filtering results and training sequence.

Key words Information filtering, information processing.