doi:10.3969/j.issn.1001-4616.2021.02.017

日志模板提取的 FT-Tree 改进算法研究

顾海艳,郑淇文

(江苏警官学院计算机信息与网络安全系,江苏 南京 210031)

[摘要] 计算机日志能够真实、全面记录事件信息,已经作为一种电子证据在侦查办案中得到普遍应用. 要实现对海量日志信息的快速自动分析,需要有高效可靠的日志模板提取方法. 鉴于目前日志模板提取方法存在的不足,本文以网络服务器日志作为模板提取的研究对象,基于 FT-Tree 算法,提出了利用 Apriori 算法计算剪枝阈值,再利用该值控制剪枝的 FT-Tree 改进算法,并以某高校网络服务器真实日志进行了模板提取实验. 结果表明,改进算法较之 FT-Tree 算法显著提高了日志模板提取的准确度,能更好地满足实际应用需要.

[关键词] 日志模板,提取方法,FT-Tree 算法,Apriori 算法,改进算法

[中图分类号]TP183　[文献标志码]A　[文章编号]1001-4616(2021)02-0121-06

Research on Improved Algorithm of FT-Tree for Log Template Extraction

Gu Haiyan,Zheng Qiwen

(Department of Computer Information and Cyber Security,Jiangsu Police Institute,Nanjing 210031,China)

Abstract:Due to the authenticity and comprehensiveness of the event information recorded by computer log files,it has been widely used as electronic evidences in the investigation and handling of cases. In order to realize the fast and automatic analysis of massive log files,an efficient and reliable log template extraction method is needed. In view of the problems existing in the current log template extraction method,we take network server log as the research object of template extraction and propose an improved FT-tree algorithm. In the proposed algorithm,the pruning threshold is calculated by using Apriori algorithm,and then the pruning is controlled by using this threshold. An experiment of template extraction is carried out using the real log file of a university network server. The results show that the improved algorithm significantly improves the accuracy of log template extraction compared with the FT-tree algorithm,and can better meet the needs of practical application.

Key words:Log template,extraction method,FT-tree algorithm,Apriori algorithm,improved algorithm

计算机日志是用于动态记录系统操作事件的、具有特定格式和特定功能的文件,主要包含系统本身、应用程序、安全管理等方面的行为信息,具有自动性、真实性和全面性等特点. 通过对日志文件信息的提取分析,可为网络安全管理提供依据,也可为确定犯罪方法、手段、途径、时间、地点提供证据支持,乃至复现相关涉网犯罪过程,因而日志已经作为一种电子证据在侦查办案中得到普遍应用. 根据中国裁判文书网(http://wenshu.court.gov.cn/)提供的 2009 年至 2019 年相关案件信息中涉及日志的已宣判刑事案件数如图 1 所示,不难发现,自 2014 年以来,日志文件越来越多地作为证据在案件中被采信. 随着各种网络应用的快速发展,对日志文件分析的需求在迅速增加,如何在网络安全管理、案件侦办过程中实现对海量日志信息的快速准确分析变得越发重要. 因此,亟需研究日志文件模板的高效可靠提取方法.

日志文件模板提取方法目前主要采用基于聚类的算法. Tang 等[1]在预先确定日志聚类数目的情况下,研究建立了基于词对关系的聚类方法以提取日志模板. 李文杰等[2]改进了基于密度的空间聚类算法,通过自适应法设置 E 邻域,实现了将对象按簇进行聚类的日志模板提取方法. Vaarandi 等[3]提出使用聚合层次聚类,利用聚类结果,选择每个聚类中与其他序列距离最小的序列作为日志模板. Nandi 等[4]通过

收稿日期:2020-08-10.

基金项目:"十三五"江苏省重点建设学科建设工程资助项目(苏教研[2016]9 号)、江苏省教育厅教改项目(2019JSJG006、2019JSJG595).

通讯作者:顾海艳,副教授,研究方向:数据挖掘,信息安全. E-mail:ghy7388@126.com

引入日志时间序列关系辅助聚类构建过程,提高了模板挖掘的准确率. 双锴等[5]提出了基于归一化特征的日志模板挖掘算法,在先验信息较少时可取得更好的效果.

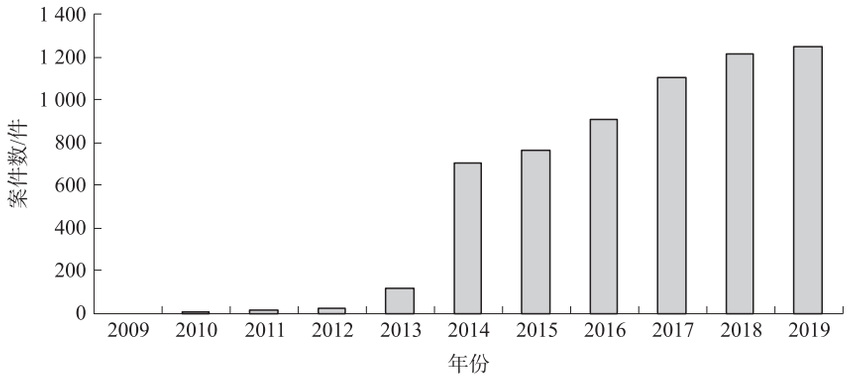

图 1　已宣判涉及日志的刑事案件数随年份的变化情况

Fig. 1　Changes of the number of criminal cases adjudicated involving logs along with the years

崔元等[6]对目前公认的基于统计模板的提取模型、基于标签识别树模板的提取模型和基于在线模板的提取模型进行了对比分析,发现基于标签识别树的提取模板模型是三者中最为稳定和可靠的算法模型,而基于统计模板的聚类算法模型在模板提取中并没有优势. 2017 年 Zhang 等[7]在标签识别树模型基础上提出了 FT-Tree(frequent template tree,频繁模板树)模型,该模型是通过识别日志单词的频繁组合来构建日志模板,实验表明,FT-Tree 模型与标签识别树模型的提取结果具有相似的准确度,但 FT-Tree 模型的计算成本和增量可追溯性比标签识别树模型更具优势. 通过对 FT-tree 算法进一步研究发现,模型构建中剪枝参数为人工设定,易导致模板提取结果的准确度受到影响. 因不同的日志文件记载不同的信息,不同计算机系统使用不同的日志记录方式[8],为此本文以最常见的网络服务器日志文件为对象,研究 FT-Tree 算法的改进方法,以提高日志模板提取的准确度.

1　FT-Tree 日志模板提取算法简介

1.1　FT-Tree 日志模板提取算法

FT-Tree 算法是在结合 FP-Grouth 算法(frequent pattern tree,频繁模式树)与标签识别树算法的优势的基础上提出的的模板提取算法[9]. 该算法源码可通过开源平台(https://github.com/slzhangsd/Craftsman)获取,用该算法提取日志模板的流程如下.

(1)第一次遍历整个日志数据,获得词频序列 L,即获得以频率大小为依据的词袋模型排序.

(2)第二次遍历整个日志数据,根据词频序列 L,对每条日志记录构造其日志单词链表{Mi}. 每个链表的第一个结点是在词频序列 L 中词频最高的单词,其它结点的排序根据单词在词频列表中从高到低顺序确定.

(3)依据是否共享共同前缀,将这些链表组合成多叉树林,即构建多棵 FT-Tree 树.

(4)第三次遍历日志数据,即遍历该多叉树林,根据每个结点的子结点数进行剪枝以构造 FT-Tree 模型. 剪枝过程中根据设定的阈值常量 K(K≥2),对每一个结点的子结点数进行统计,如果某结点的子结点数小于常量 K,就意味着该结点与大部分结点情况不匹配,则将该结点的所有子结点删去,使该结点成为叶结点.

(5)如果有新的数据需要处理,只需要在已有的 FT-Tree 模型的基础上重复进行生长和剪枝操作.

1.2　FT-Tree 提取日志模板算法优劣分析

FT-Tree 算法提取日志模板的基本思路是:日志模板应当是在日志记录中较为频繁出现的词袋模型的组合,该算法只要遍历 3 次日志数据即可得到模板结果. 相比于聚类算法需要进行多次迭代[10],FT-Tree 提取算法的优势:一是可大大减少计算时间和占用计算资源;二是如果新加入的日志数据不引起词频列表数据顺序的变化,则该算法构造的模型可以自动生长,这是需要重新训练的聚类算法所不能比拟的;三是生成的模板提取结果是结构化数据,便于做进一步分析研究.

虽然 FT-Tree 算法优势明显,但也存在不足,主要表现在构造 FT-Tree 树时,其剪枝阈值常量 K 是由人工凭经验设定,随机性大、缺乏理论依据. 由于各类日志数据的差异较大,仅凭经验确定剪枝阈值,很有可能出现幸存者偏差,因此需要对剪枝参数的确定方法进行改进.

2 FT-Tree 算法的改进思路及其实现

2.1 FT-Tree 算法的改进思路

经分析比较,本文选用 Apriori 算法对 FT-Tree 算法的剪枝过程进行改进. Apriori 算法是经典的关联规则挖掘算法,最早由 Agrawal 等人于 1993 提出[11]. 一般而言,关联规则可表示为形如 $A \rightarrow B$ 的蕴含表达式,其中 A 和 B 是不相交的项集. 关联规则的强度可用支持度(support)和置信度(confidence)来度量[12].

$A \rightarrow B$ 规则的支持度是指该规则在给定事务集中出现的概率,其计算如下:

$$\text{support}(A \rightarrow B) = P(A \cup B), \tag{1}$$

$A \rightarrow B$ 规则的置信度是指 B 在包含 A 的事务中出现的概率,其计算如下:

$$\text{confidence}(A \rightarrow B) = P(A|B) = P(A \cup B)/P(A), \tag{2}$$

通过发现满足最小支持度的频繁项集,可进一步提取满足最小置信度的强规则[13]. 本研究利用 Apriori 算法通过抓取日志记录中词关联的强规则,来获得日志模板.

改进后的 FT-Tree 算法具体步骤如下:

(1)-(2)步骤与 FT-Tree 算法相同.

(3)依据是否共享共同前缀,将这些链表组合成多叉树林. 多叉树的结点以列表中的词命名,同时结点中保存该词在构造过程中出现的次数.

(4)计算各个结点置信度 $\{x_i\}$,并利用 Apriori 算法计算置信度阈值 X. 第 i 个结点的置信度计算公式为:

$$x_i = n_i/n, \tag{3}$$

式中,n 为所在树的根结点出现次数,n_i 为第 i 个结点在对应位置出现的次数. 对于每棵 FT-Tree 树,其根结点是词频最高的频繁词,结点的置信度也就是{根结点}→{非根结点}这个关联规则的置信度,置信度较大的结点(即出现次数排在前列,人工设定置信度前列的标准为 ε)也就是强规则结点. 利用 Apriori 算法,根据 ε 计算强规则结点的置信度阈值 X.

(5)根据置信度阈值 X,对非强规则结点进行剪枝. 置信度标准 ε 虽然也是人工设定,但它是根据置信度排在前列的原则确定,要比直接根据人工选定的子结点数阈值常量 K 进行裁剪更加科学.

2.2 改进 FT-Tree 算法的实现

网络服务器日志文件的特点:(1)不存在以汉字为基础的日志记录格式,没有必要考虑汉字对日志模板的影响;(2)日志记录各成分间主要以“[ ]”“,”“{ }”为隔断. 针对这些特点,为兼顾算法准确度与速度,本文采用正则表达式作为分词的依据,以研究 FT-Tree 算法的改进算法.

2.2.1 数据预处理

日志文件数据预处理主要包括数据清洗和构造词袋模型两个过程. 数据清洗的主要工作包括删除冗余的日志记录、对日志记录进行分类并剔除日志记录中的具体报错信息、删除日志记录中的 URL,以提高模板提取的准确性. 构建词袋模型就是生成字典格式记录的数据,这是模板提取程序段的输入数据.

(1)数据清洗

清洗过程中主要使用 4 个函数.

①Deduplication1 函数

由于网络在建立端与端之间连接时常常会失败,从而留下冗余的日志记录. 本文使用 python 的 re 标准库中的函数 Deduplication1 去除重复日志记录.

②Classify_ * 函数、check_Classify 函数

网络服务器在提供 java 等服务出错时会报错,而日志会详细记录报错内容. “WANGRING”“INFO”“ALERT”和“NOTICE”这 4 类日志记录中都可能会记录报错信息,本文编写了 Classify_ * 函数和 check_Classify 函数,实现对日志记录的分类和报错日志中错误内容的处理.

③Deduplication5 函数

记录在网络服务器日志中的 URL 常常会因调用同一种服务而重复出现. 例如提供邮件服务的网络服务器在用户每次登陆时都会调用对应的 jsp 服务. 如果不进行清洗,很有可能会成为后期日志模板提取的噪点,从而影响日志模板提取的准确性. 为此本文选用 re 标准库中 Deduplication5 的函数去除日志数据中的 URL.

(2)构造词袋模型

自然语言处理(NLP)技术通过切词将目标文本划分为不同的单元,并计算其权重. 网络服务器日志文件以“Time stamp”“Message-type”“Detail message”为切入点,由于“Detail message”数据量庞大,在使用空格区分“Time stamp”“Message-type”之后,只能选择使用逗号来分隔“Detail message”. 本文使用 re 标准库中的切词函数 abstract2,构建根据逗号切割的正则表达式(,[^(,|\{|\|\[|\])]*?,);再利用 re.findall( )函数将所有匹配结果以字典形式返回. 根据字典的键值对,可清晰展现各单词的出现次数.

2.2.2　算法实现

算法实现就是把已经完成切词的数据按照一定的方法构建成有逻辑结构的多叉树林. 改进的 FT-Tree 日志模板提取算法主要实现步骤如下:

(1)构建词频序列. 由于切词过程中已经将日志中的单词及其出现次数写入了对应字典中,故该步骤只需要调取对应字典即可.

(2)调用 seedcreate 函数并按行接收日志记录,按词频序列由高到低,建立该记录的单词链表,然后将该链表作为返回值输出.

(3)构建多叉树林. 基于 treelib 标准库,根据 seedcreate 函数返回值构建改进的多叉树林.

(4)计算置信度和置信度阈值. 根据公式(3)计算每个非根结点置信度 x_i. 一般研究认为置信度在前 30%的为强规则结点,因而本文设定置信度前列的标准 $\varepsilon = 1/e$(e 为自然常数,取 e=2.718),即选取置信度在前 36.79%的结点为强规则结点.

(5)剪枝构建改进 FT-Tree 模型. 根据得到的置信度阈值 X,对构建的多叉树林进行剪枝.

3　结果与讨论

为了检验本文改进算法的日志模板提取效果,在联想电脑拯救者 r720-15IKBN、Windows10 家庭中文版、python3 环境中,利用某高校真实网络服务器的 10 MB 日志文件 wmsvr.log.2019-08-21 进行了日志模板提取实验.

3.1　实验过程

(1)数据清洗. 数据预处理结束后,形成一个具有 712 个单词的词频列表,列表中排在前 10 的单词集和出现次数如表 1 所示.

表 1　词频列表中前 10 位的单词出现次数

Table 1　Occurrence times of the top 10 words in the word frequency list

序号	单词	出现次数	序号	单词	出现次数
1	requestURL=	22 200	6	remote=	11 666
2	opTime=	21 777	7	reqTime=	11 342
3	result=	13 256	8	respTime=	11 342
4	requestVar=	12 377	9	user:	6 977
5	func=	11 666	10	uid:	5 658

(2)提取日志模板. 经检查,10 MB 的日志文件共包含 28 417 条日志信息,用改进算法提取发现共有 64 条日志模板. 部分提取结果如图 2 所示.

(3)对比实验. 将本文改进算法与 FT-Tree 算法进行实验对比分析. 因 FT-Tree 算法需要人为设定剪枝常量 K 值,为检验不同 K 值对提取模板结果的影响,使 K 值依次从 2 到 10 循环运行,并根据 apache 系统通用日志格式,对提取的模板数据进行准确度分析. 两种算法运行后提取的模板数量、符合通用格式的模板数量对比如表 2 所示.

图 2　改进 FT-Tree 算法提取的部分日志模板

Fig. 2　Partial log templates extracted by the improved FT-Tree algorithm

表 2　两种 FT-Tree 算法提取的日志模板数对比

Table 2　Comparison of the number of log templates extracted by two FT-Tree algorithms

算法	剪枝参数	提取的模板数	符合通用格式的模板数
FT-Tree 算法	$K=2$	37	32
	$K=3$	11	4
	$K=4$	11	4
	$K=5$	8	0
	K=6	8	0
	$K=7$	8	0
	$K=8$	8	0
	$K=9$	8	0
	$K=10$	8	0
本文的 FT-Tree 改进算法	Apriori 算法计算得到的 X	64	61

3.2　实验结果分析

(1)从表 2 可以发现,随着 FT-Tree 算法中剪枝常量 K 值增加,提取的模板数量随之减少,说明剪枝参数设定对模板提取的结果有明显的影响. 这也进一步说明对剪枝参数设定方法进行研究很有必要.

(2)由表 2 还可以发现,FT-Tree 算法在 $K=2$ 时提取模板数最多,共 37 种,提取的结果准确度最高达到 86.5%;而改进算法可提取 64 种模板,提取结果的准确度达到 95.3%. 这充分表明改进算法提取的模板结果不仅丰富,而且准确度高.

(3)另外根据实验中记录的中间数据还可以发现,Apriori 算法计算的置信度阈值 $X=0.001\ 081$,这是人工凭经验无法确定的数值,说明用算法选取阈值的方法更科学准确.

综上,两种算法的模板提取结果对比,充分说明改进的算法有效提高了日志模板提取的准确度.

4　结论

随着涉网案件数量的快速增长,日志文件作为电子证据在侦查办案中越来越受到重视,日志文件的分析研究也将受到更多关注. 本文鉴于当前日志模板提取中存在的不足,提出了基于 FT-Tree 算法的日志模板提取改进算法. 针对 FT-Tree 算法因剪枝阈值人为设置而导致模型准确度不高的问题,将 Apriori 算法利用置信度抓取强规则的思想引入模型构建中,改进了原算法中剪枝参数的确定方法,从而提高了日志模板提取的准确度;并利用某高校真实网络服务器日志文件进行了实验验证,结果表明改进的 FT-Tree 算法是可靠、高效的,能更好地满足实际应用需要.

[参考文献]

[1]　TANG L, LI T, PERNG C S. LogSig: Generating system events from raw textual logs[C]//Proceedings of the 20th Association for Computing Machiner(ACM) International Conference on Information and Knowledge Management. New York, NY, USA:

ACM,2011:785-794.
[2] 李文杰,闫世强,蒋莹,等. 自适应确定 DBSCAN 算法参数的算法研究[J]. 计算机工程与应用,2019,55(5):1-7,148.
[3] VAARANDI R,PIHELGAS M. LogCluster—A data clustering and pattern mining algorithm for event logs[C]//2015 11th International Conference on Network and Service Management(CNSM).Barcelona,Spain:IEEE,2015:1-7.
[4] NANDI A,MANDAL A,ATREJA S,et al. Anomaly detection using program control flow graph mining from execution logs[C]// Association for Computing Machinery International Conference on Knowledge Discovery and Data Mining(ACM SIGKDD). New York,NY,USA:ACM,2016:215-224.
[5] 双锴,李怡雯,吕志恒,等. 基于归一化特征判别的日志模板挖掘算法[J/OL]. 北京邮电大学学报:1-6[2020-02-09]. https://doi.org/10.13190/j.jbupt.2019-033.
[6] 崔元,张琢. 基于大规模网络日志的模板提取研究[J]. 计算机科学,2017(11A):448-452.
[7] ZHANG S L,MENG W B,BU J H,et al. Syslog processing for switch failure diagnosis and prediction in datacenter networks [C]//2017 IEEE/ACM 25th International Symposium on Quality of Service(IWQoS). Vilanova i la Geltrú,Spain:ACM, 2017:1-10.
[8] 刘洪歧,陈远平,马建化. 系统日志模板提取方法研究[J/OL]. 计算机系统应用,2019,28(10):239-244. http://www.c-s-a.org.cn/1003-3254/7112.html.
[9] ZHANG S L,SONG L,ZHANG M,et al. Efficient and robust syslog parsing for network devices in datacenter networks[J]. IEEE access,2020,8:30245-30261.
[10] 李峰,李明祥,张宇敬. 局部迭代的快速 K-means 聚类算法[J/OL]. 计算机工程与应用:1-11[2020-07-01]. http://kns.cnki.net/kcms/detail/11.2127.tp.20190815.1706.027.html.
[11] 廖纪勇,吴晟,刘爱莲. 基于布尔矩阵约简的 Apriori 算法改进研究[J]. 计算机工程与科学,2019,41(12):2231-2238.
[12] 郭涛敏. 基于轻量化关联规则挖掘的安全日志审计技术研究[J]. 现代电子技术,2019,42(15):83-85.
[13] TAN P N,STEINBACH M,KUMAR V,等. 数据挖掘导论[M]. 北京:人民邮电出版社,2011:202-207.

[责任编辑:顾晓天]