

基于大规模网络日志的模板提取研究

崔元¹ 张琢^{1,2}

(东北师范大学信息与软件工程学院 长春 130117)¹

(教育部数字化学习支撑技术工程研究中心 长春 130117)²

摘要 针对直接从大型网络日志中提取网络事件困难的问题,提出了基于大规模网络日志的模板提取方法。该方法可将海量的、原始的网络日志主动转换为日志模板,从而为了解网络事件的根因和预防网络故障的发生提供重要的前期准备。首先分析日志的结构,将日志中的词划分为模板词和参数词两类;然后从 3 个不同的角度切入,分别对日志进行模板提取研究;最后使用互联网公司中的实际生产数据,采用 Rand_index 方法来评估 3 种提取方法的准确有效性。结果表明,在从服务集群中收集来的 4 种不同消息类型中,基于标签识别树模型提取到的日志模板的平均准确率达到 99.57%,高于基于统计模板提取模型和基于在线提取模板模型的准确率。

关键词 切词,提取模板,统计聚类,标签识别树,在线聚类

中图法分类号 TP311 文献标识码 A

Research on Template Extraction Based on Large-scale Network Log

CUI Yuan¹ ZHANG Zhuo^{1,2}

(School of Information and Software Engineering, Northeast Normal University, Changchun 130117, China)¹

(Digital Learning Support Technology Engineering Research Center, Ministry of Education, Changchun 130117, China)²

Abstract Aiming at the problem of extracting network events directly from large-scale network log, a template extraction method based on large-scale network log was proposed. The method can automatically convert the massive and original network logs into log templates, so as to provide important pre-preparation for understanding the network events root causes and preventing the occurrence of network failure. Firstly, the structure of the log is analyzed, and the words in the log are divided into two types: template word and parameter word. Then, from three different angles, the log template extraction is studied respectively. Finally, the actual production data of the Internet company is used, and Rand_index method is used to evaluate the accuracy and validity of the three extraction methods. The results show that the average accuracy of the log templates based on the tag recognition tree model is 99.57%, which is higher than that of the four different types of messages collected from the service cluster.

Keywords Cut words, Extract template, Statistical clustering, Signature tree, Online clustering

1 引言

随着网络元素和网络服务数量的增加,所产生的网络日志被网络服务者视为用于监控网络健康和故障排除的重要数据源之一。而网络服务的多样性使得网络事件的关系变得更加复杂化,例如一个普通的连接断开事件不仅会导致邻近节点无法正常工作,还可能导致虚拟路径连接断开或导致相关的服务全都无法正常工作。而在大型生产网络中,由于其记录的日志信息数量庞大且格式多样化,直接从中分析出网络事件已经成为一个具有挑战性的任务。通常情况下,采用网络管理系统(NMS)来监控网络,并使用简单网络管理协议(SNMP)或系统日志触发器应对每天发生在大型网络中的网络事件。NMS采用预定义的规则,当出现报警时,表明网络已到达明显的临界状态。

尽管 NMS^{1),2)}具有根因分析功能,但是它需要领域知识且不能够识别出详细的网络事件。Meta^[2]和 TAR^[3]都是故

障定位系统,它们可以从事件数据集中自动学习故障事件,并通过索引网络事件快速找到故障的根因。但是,以上两者均依赖于 NMS 收集的网络事件数据。除此之外,NMS 的报警是模糊的,网路管理者不能够捕获例如网络层、协议或者服务依赖的网络事件的影响或结构。因此,在进行网络异常检测之前要先获取网络事件的结构。

本文在不依赖领域知识和 NMS 数据的前提下,将研究范围放在原始日志 syslog 上,从原始日志数据 syslog 中主动提取网络事件,从而为根因诊断、故障检测和预防打下基础。然而,直接从 syslog 中发现网络事件是极为困难的,其原因有两个:1)日志非结构化。日志消息是由不同的供应商提供的规则所产生的不同的非结构化文本消息。由于大型网络供应商包含了多样的元素,导致日志也非常多样化。2)日志复杂化。网络设备如路由器、交换机、跨多个地理位置的服务器等发生的网络事件均会产生日志

基于以上原因,需要转换思路,直接从 syslog 中提取网

¹⁾ https://en.wikipedia.org/w/index.php?title=Net-work_management_system&redirect=no

²⁾ CA Spectrum. <http://www.ca.com/us/root-cause-analysis.aspx>

崔元(1990—),女,硕士生,主要研究方向为数据挖掘,E-mail:cuiy245245@yeah.net;张琢(1968—),女,博士,教授,主要研究方向为信息检索、数据挖掘,E-mail:zhangzhuo_ca@sina.com。

络事件→从 syslog 模板中提取网络事件,具体来说就是先从原始日志中提取日志模板,再在日志模板中提取与网络事件有关的信息^[5],从而达到从原始 syslog 中提取网络事件的目的。因此本文的目标是获取日志模板。虽然模板可以从供应商提供的手册中获得,但是由于网络设备的升级更新,日志格式可能会发生变化,并且一些供应商并不会公开自己的日志手册,因此这样获取模板的方式就会受到阻挠。基于此,本文提出了基于海量日志的模板提取模型,从 3 个不同的角度切入,分别对日志进行模板的提取,并使用某互联网公司的真实生成数据对 3 种方法进行比较。本文详细说明了 3 种提取日志模板的方法,并对 3 种方法做出相应的优化,最后在研究总结第 3 种方法时,提出了主动提取日志模板的方法,弥补了前两种方法的不足。

提取日志模板的研究是最为重要的,因为该分析是提取网络故障的隐藏网络事件、理解网络异常的根源和影响以及进行预防性维护操作的基石,后面的工作都需要围绕和使用日志模板来展开。

2 网络日志结构分析

网络日志数据,如路由器系统日志和警报,包含各种信息(网络故障信息、安全问题信息和控制台日志信息)。由于日志 syslog 的格式取决于服务类型或供应商,并且是自由形式的文本,因此其语法和语义也不尽相同^[4,11]。但我们依然可以在 syslog 中观察到固定不变的短文本(表 1 列出了来自两个路由器供应商的日志示例),包括以下 3 个核心结构。

- 1) Time stamp: 指日志消息生成的时间戳;
- 2) Message-type: 指日志消息的类型,通常有 notice, error 等;
- 3) Detail Message: 指日志消息的详细信息。该部分一般包括两个部分:代表状态变化(例如 up/down)的部分和代表参数(例如 IP 地址、主机名和进程 ID 等)的部分。

表 1 网络日志部分消息

Vendor	Time stamp	Router	Message-type	Detail Message
V1	2016 Feb 10 18:40	R1	notice	Interface FastEthernet 0/9
V1	2016 Feb 9 20:28	R4	notice	Interface te-1/1/8, changed state to up
V1	2016 Feb 10 14:01	R2	notice	Configured from console by vty2 (10.11.xxx.yyy)
V2	2016 Feb 8 15:14	R11	System	Interface etherer xxx, state down
V2	2016 Feb 6 20:32	R12	LINK_UPDOWN	GigabiteEthernet 1/0/18 link status is DOWN

3 基于统计模板提取模型

3.1 切词评分

既然要提取日志的模板,就需要将日志中的单词进行区分,确定哪些词是模板词,哪些词为参数词,而模板词就可以组成日志的模板。因此需要先对日志进行切词。

假设每个消息中的单词都是用空格分割开的,那么就可以以空格进行切词,从而将得到的单词分为两类:模板词(Template Words)和参数词(Parameter Words)。一般来说,模板词通常会出现在具有相等长度 len 的日志消息的相同位置 p 上。因此在切词过程中,记录下单词的位置和日志消息的长度(p, len)。使用基于条件概率的计算公式得出每个单

词作为模板词的可能性,将此概率作为该词的得分,评分规则如下。

如果一个单词出现在长度为 len 的日志消息的第 p 个位置上,那么这个单词的分数 $Score$ 可以依据式(1)计算:

$$Score(word, p, len) = P(word | p, len) \quad (1)$$

其中, p 表示 $word$ 出现在日志的位置下标, len 表示日志消息中单词的个数,即句子的长度。

3.2 基于 DBSCAN 聚类单词

通过 3.1 节对单词评分之后,需要确定一个词是参数词还是模板词。一个简单的方法是设置一个阈值,然而这种方法会存在问题,即阈值大小的确定。如果阈值设置偏大,就会使得本应该属于模板词的单词被误认为参数词,最终获取出来的模板中表达的信息较少,不利于为后面的相关分析提供帮助;但如果将阈值设置得偏小,便会出现很多伪模板词,使得日志模板中包含了较多的具有变化因素的参数词,这样会对网络事件的提取造成干扰。

为了避免上述情况, Kimura^[6] 提出利用统计聚类算法来获取日志模板的方法。此处基于 Kimura 提出的思想,采用了基于密度的空间聚类算法(DBSCAN)聚类技术^[10],该技术需要确定单词之间的密度、距离、半径以及最少数点的值。确定完成之后,即可通过该聚类算法将日志消息中的单词分为多个类别,并且类别之间的距离大于某个阈值。

3.2.1 确定聚类的相关参数

首先确定聚类算法的相关参数: Eps 和 $MinPts$ 。这里需要指定 Eps 和 $MinPts$ 两个参数。但对于非专业或非领域内的人员来说,直接指定出恰当的值较为困难,故本文参考文献[1]中的参数自适应动态选择方法对 Eps 和 $MinPts$ 进行确定,基本思想为先采用 k-均值的聚类思想对点进行初步分类;然后统计每个类中的各个点之间的距离,该距离本文采用二维欧几里德距离¹⁾公式计算得到;最后再根据正态分布原理,为每个类指定恰当的 Eps 和 $MinPts$ 参数^[1]。

3.2.2 基于 DBSCAN 聚类日志模板

将 DBSCAN 的参数设置为 3.2.1 节得到的最优参数,然后将单词评分作为聚类算法的输入,将单词分为多个簇,并从概率最高的簇开始,按照概率降序的顺序,依次从对应的簇中取出单词并加入模板中,直到在句子长度相同的条件下,模板词所占句子的比例大于 β 为止,其中 $0 < \beta < 1$ 。图 1 为该模型的伪代码。

```

1. words = getWordsFromsyslog();
2. scores = getScore(words);
3. finalScore = convertScore(scores);
4. clusters = DBSCAN(finalScore);
5. for each cluster do;
6.     templateWords = getWordsFromCluster(cluster);
7.     len = len(templateWords);
8.     P = len/Len(word);
9.     If P < β then
10.        len += len;
11.        append templateWords to finalTemplateWords;
12.     else
13.        print finalTemplateWords;
14.     end if
15. end for

```

图 1 基于聚类统计提取模板

¹⁾ <http://shiyanjun.cn/archives/1288.html>

3.3 参数的选择

从获取模板词数量的角度来进行分析,根据以上算法,当 β 减小时, $P < \beta$ 条件较容易满足,使得很多模板词被丢弃,从而造成模板词和模板数量过少,进而提取不到有价值的信息;当 β 增大时,会使得很多不该成为模板词的单词被当成模板词添加到日志模板中。根据以上讨论并通过咨询互联网数据操作团队,最后选定 $\beta=0.75$,此时提取出来的模板词最为可靠。

4 基于标签识别树模板提取模型

4.1 消息类型与子类型

结合第2节和表1可知,日志的第二个结构——消息类型(Message-type)是用来描述消息的特性的。然而其所表达的粒度较大,很多关于消息类型的细节并不清楚,故在日志的第三个结构——详细信息(Detail Message)部分做出补充。因为该部分包含了大量的内容,故着重从该部分进行分析是本文模板提取模型的重点。可以从中发掘出多个子类型^[9],这些子类型就是我们需要的日志模板。如表2所列有8条日志,其中包含两类日志消息类型,notice和warning。

表2 日志类型示例

#	Time stamp	Message-type	Detail Message
1	12:31:40	notice	Interface ae2, changed state to down
2	12:32:51	notice	Interface ae3, changed state to up
3	12:33:12	notice	Interface ae4, changed state to down
4	12:34:24	notice	Interface ae5, changed state to up
5	12:35:35	warning	A single neighbor should be configured
6	12:36:57	warning	A single neighbor should be configured
7	12:37:33	notice	Vlan-interface vlan20, changed state to up
8	12:38:03	notice	Vlan-interface vlan18, changed state to down

从表2的示例中可以看出,Message-type分为notice和warning两类,以Message-type=notice时的日志的详细信息部分为例,即日志编号依次为1,2,3,4,7,8的detail message。从表2中可以发现,ae2,ae3字段在每条消息之间是不同的,更趋向于参数词。当把这两个字段用相同符号替换后(用星号*替换,如表3所列)会发现,只剩下3种不同的结构类型,即日志的子类型,故可以说在Message-type=notice时,日志有3个子类型。然而在实践中,因为所有需要替换的部分没有明显的模式,如IP地址、vrf ID、交换机ID等各不相同,在没有领域知识的前提下,手动区分和找到需要替换的部分将十分困难。

表3 日志子类型示例

Template No.	Message-subtype	Detail Message
T1	Interface*	, changed state to down
T2	Interface*	, changed state to up
T3	Vlan-interface*	, changed state to up

4.2 模板树的构造

根据4.1节的问题,即在不需要领域专家干预的条件下自动获取如表3所列的子类型的问题,本文的解决办法源于用于垃圾邮件检测的签名抽象^[4]。其思想是子类型标签节点

是具有高频单词的组合,这里把子类型标签节点映射为日志中频繁出现的单词组合,主要思路如下。

(1)切词:将消息通过空格分隔为单词。

(2)获取词频:统计在该日志块中不同Message-type下各个单词出现的次数。

(3)组合单词:判断在单词A出现时,单词B总是同A一起出现。如果A和B的词频相同且同时出现,则将A,B两个单词作为单词组合,作为模板树上的一个节点,如图2所示。

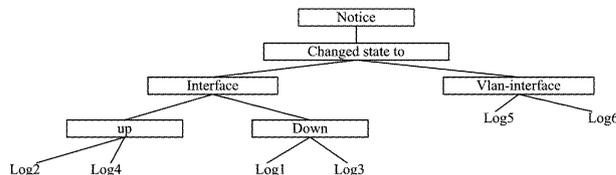


图2 日志子类型模板树

对于每种类型的消息,作为模型的输入,构造一棵树用来表示模板作为输出,如图2所示的日志模板树。当一个单词出现在消息中时称做关联。详细的构造算法的过程依赖广度优先搜索(BFS)遍历。首先将消息类型(例如notice)作为树的根。然后给定父节点,寻找最频繁单词组合作为子节点,重复此过程,根据剩余消息创建子节点,直到所有消息都已关联。最后继续递归到子节点并重复该过程。

这种方法是非常通用的,因为是根据单词频率进行模板提取而不是基于文本语义进行提取,因此解决了在不需要领域专家干预的条件下,自动获取子类型的问题。

4.3 模板树构造算法的优化

由于上节提到的递归方法需要频繁地重新切词来获取最高频率词组,会增加时间消耗,故优化此算法,只在最开始进行一次切词工作即可。采用一个对象容器WordSeg实时记录最新的词频和相应的句子索引集合,即key为单词,value为该词的词频和句子索引集合。具体数据结构如表4所列,其表示初始状态。在求解模板树的子节点的过程中,先将该词的句子索引集合与父节点的句子索引集合求交集,再与该词在同一层级的兄弟节点的句子索引集合求差集,从而可以快速找到词频最高的单词或单词组作为模板树的子节点。

表4 WordSeg数据结构(初始状态)

Word	词频	log_index_dict
[changed state to]	6	{1,2,3,4,5,6}
Interface	4	{1,2,3,4}
Up	3	{2,4,6}
Down	3	{1,3,5}
Vlan-interface	2	{5,6}

结合图2举个例子,为了叙述方便,先进行以下约定。

- 1)节点 changed state to: 节点 nodeA;
- 2)节点 Interface: 节点 nodeB;
- 3)节点 Vlan-interface: 节点 nodeC;
- 4)节点 Up: 节点 nodeD;
- 5)节点 Down: 节点 nodeE。

现已确定nodeA和nodeB为树节点,此时的WordSeg数据结构如表5所列。现构建nodeA的另一个子节点,即nodeB的兄弟节点。在最新WordSeg中,已经成为树节点的词或log_index_dict为 \emptyset 的单词不予考虑(即nodeA,nodeB),其他的单词(nodeC,nodeD,nodeE)均与nodeA求交集,然后再与nodeB求差集,得到临时wordSeg(temp WordSeg,如表

6 所列),在该 *temp WordSeg* 中找到词频最高的词 *nodeC*,将其作为树节点,画树并更新 *WordSeg*,结果如表 7 所列。

表 5 *WordSeg* 数据结构

Word	词频	log_index_dict
[changed state to]	0	∅
Interface	0	∅
Up	3	{2,4,6}
Down	3	{1,3,5}
Vlan-interface	2	{5,6}

表 6 *temp WordSeg* 数据结构

Word	词频	log_index_dict
[changed state to]	0	∅
Interface	0	∅
Up	1	{6}
Down	1	{5}
Vlan-interface	2	{5,6}

表 7 *WordSeg* 数据结构(更新之后)

Word	词频	log_index_dict
[changed state to]	0	∅
Interface	0	∅
Up	1	{6}
Down	1	{5}
Vlan-interface	0	∅

4.4 模板树的剪枝

当父节点超过 k 个子节点时,将丢弃所有子节点,使父节点本身成为一个叶子节点,这样从 *root* 节点到叶子节点的每条路径均为一个日志模板,如图 2 中“Interface * changed state to up”等。实验证明 $k=10$ 可以有效保留节点且日志模板完整,伪代码如图 3 所示。

```

1. words = cutMessage();
2. wordFrequency_logIndex_dict = getFrequency(words);
3. for each word do;
4.   If wordFrequency1 == wordFrequency2 and wordFrequency_logIndex_dict1 ∩ wordFrequency_logIndex_dict2 == wordFrequency_logIndex_dict1 then;
5.     append word to combinations;
6.   end if
7. end for
8. rootNode = messageType;
9. parentNode = rootNode;
10. parentNode_logIndex_dict = getLogIndex(parentNode);
11. childNode = getChildNode(wordFrequency_logIndex_dict);
12. paintTree(childNode);

```

图 3 基于标签识别树提取模板

5 基于在线模板提取模型

5.1 词分类

通过对上述两个模型的研究,发现存在两大问题。

(1) 单词分类粒度大:日志句子中的单词简单划分为模板词和参数词,非此即彼,没有对单词进行再细一层的划分。

(2) 均采用离线批处理的方式,其不足之处在于:因为日志消息的格式可能会在将来发生改变,而且还需要观察很长时间才能捕获所有模板,灵活度不足。

因此, Kimura 提出了一种在线模板提取方法^[8],可以以增量的方式学习并提取模板。

该方法的主要思想是通过计算日志与日志模板的相似度,使该日志自动快速地将日志消息转换为日志模板。

该方法的主要过程是,系统一开始并没有日志和日志模

板,当新到模板进入系统之后,该模型可以不断地聚类 and 更新簇的集合,即通过自动计算日志消息和日志模板之间的关系进行聚类,而不使用任何以前的日志知识。将主要思想进行细化:

(1) 基于属于日志模板的趋势对每个单词进行分类,分为只有符号、只有字母、只有符号和字母、只有数字和字母、只有数字或者只有数字和符号 5 类,表 8 列出了单词分类定义的示例。

表 8 单词分类

#	分类定义	例子
1	只有数字或只有数字和符号	1,0/0,10.1.1.1
2	只有数字和字母	host-01,IPv4,L2TP,vty0,Fa0/0
3	只有符号和字母	class-a,udp-port,aaa.cfg,line-protocol
4	只有字母	linkdown,state,interface
5	只有符号	<,>,,:

(2) 通过将日志模板作为消息簇,并通过计算消息簇和新增消息之间的日志相似性(log similarity)来在线实时聚类新增的消息到相应的模板簇中。

5.2 设置单词权重

从对日志消息的观察,诸如“=”或“:”符号可能属于日志模板,诸如 IP 地址、进程 ID 等可以被认为是参数,首先对单词分类进行详细的定义,定义 $\omega = [\omega_i] (i=1,2,\dots,5)$ 作为成为每个类型 i 的日志模板词的权重向量。根据该定义, ω 的值通常设置为 $\omega_1 \leq \omega_2 \leq \omega_3 \leq \omega_4 \leq \omega_5$ 。

5.3 在线日志聚类

对于每个新到来的日志消息,执行在线聚类算法,使得消息被分配到具有最高相似性的消息簇中,具体如下。

当新增的 syslog 消息进入模型时,根据消息中不同类型的单词的数量,该方法将新消息分配到现有的模板集群中或为该新消息创建新的模板集群。该方法的重点在在线学习 syslog 消息模板的词类而非单词本身,因此属于不同子类型的系统日志消息可以很容易地分配到一个模板集群中。其中,将消息 X 分配到集群 C 的相似度计算如下:

$$\log \text{Similarity}(C, X) = \omega'x / \omega'c_x \quad (2)$$

其中, $x = [x_i]$ 表示 X 中的属于类 i 的单词的数量,并且 $c_x = [c_{x,i}]$ 表示同时属于 C 和 X 中出现的类的 i 单词的数量。基本思想是如果 $\log \text{Similarity}$ 的最高值大于或等于阈值 E ,就将消息 X 合并到对应 C 中,否则为 X 创建一个新的模板集群。该模型的伪代码如图 4 所示。

```

1. template cluster set C = ∅;
2. for each message X do;
3.   getClassOfWord(X);
4.   C = getClusterHasHighestLogSimilarity(C, X);
5.   HighestLogSimilarity = getHighestLogSimilarity(C, X);
6.   If HighestLogSimilarity ≥ E then;
7.     append X to cluster C;
8.   else
9.     create a new cluster from X;
10.    C = CUC;
11.   end if
12. end for

```

图 4 基于词类别在线提取模板

图 4 中首先提取了新增日志 X 的词分类。然后搜索出 X 与簇集中具有最高 $\log \text{Similarity}$ 的消息簇,如果相似度值大于 E ,则 X 被聚合到最高消息簇中;否则,为 X 创建一个新的集群。最后得到的消息簇即为日志模板。

5.4 参数的选择

根据 E 的定义,如果 E 的取值越大,便会创建出更多的集群;而 E 的取值越小,则会使得当前集群 C 包含过多伪模板词,从而集群的总数量也会减少。通过多次运行算法且根据互联网数据操作团队对提取到的模板进行分析之后,选取 $E=0.93$ 时,模板是最合理的。

6 评估 3 种提取模板模型

通过以上 3 种方法模型分析 syslog 并提取模板,我们认为基于标签识别树提取模型在本文设定的场景中最为稳定。为了验证该结论,使用互联网公司的实际数据进行验证。

由于数据操作中心团队每天都在接触日志信息,因此他们能够熟练地分析出所给日志信息所代表的网络事件。因此,基于操作团队提供给我们的分类依据,可以手动分类消息类型,并作为评估 3 种方法的真实依据。选择了 3 种服务集群下的实例,即分析了数十亿的系统日志消息,从中随机收集一份日志样本进行评估,结果如图 5 所示。

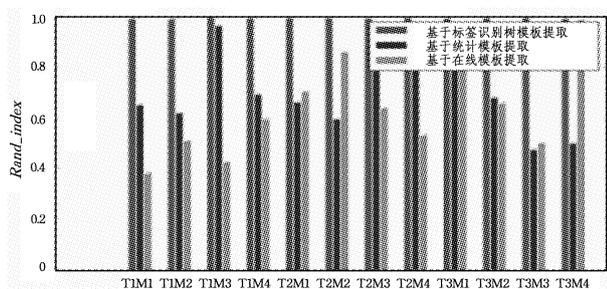


图 5 3 种提取模板方法的准确率比较

对于每种类型的实例,首先从中随机选择 4 种消息类型,对于每种消息类型,随机收集了 500 条日志消息,即 2000 条数据进行比对。然后操作团队根据每条消息代表的事件手动分类日志消息,最后我们使用基于标签识别树的方法、基于统计模板的方法和基于在线模板提取的方法分别提取每种消息类型的模板。为了定量比较这 3 种方法的准确性,采用了基于手动分类结果的 $Rand_index$ 方法。 $Rand_index$ ¹⁾ 是一个用于评估两种数据聚类方法之间的相似性的方法,可以将通过手动分类的结果模板与通过上述 3 种方法提取的结果模板作为该评估方法的输入,通过 $Rand_index$ 来评估每种方法的准确性。具体来说,重复地从 100 条日志中随机选择两个消息 X 和 Y ,并定义 A, B, C, D 4 个指标。

1) A : X 和 Y 被手动分在同一类中且 X 和 Y 被方法定义为同一个模板;

2) B : X 和 Y 被手动分到不同类中且 X 和 Y 被方法定义为不同的模板;

3) C : X 和 Y 被手动分到不同类中且 X 和 Y 被方法定义为同一个模板;

4) D : X 和 Y 被手动分在同一类中且 X 和 Y 被方法定义为不同的模板。

$Rand_index$ 被定义为:

$$Rand_index = \frac{A+B}{A+B+C+D} \quad (3)$$

图 5 给出了 3 种方法分别提取 4 种消息类型下的 $Rand_index$ 评分。从图中可以看到基于标签识别树提取模板模型

的 $Rand_index$ 的平均分为 99.57%, 高于基于统计模板提取模型的 77.92%, 高于基于在线模板提取模型的 57.06%。因此,基于标签识别树提取模板模型是最为稳定和可靠的提取日志模板方法。

结束语 本文在不需要领域知识的前提下,直接且在线地从海量的原始日志中提取日志模板,为以后从日志模板中提取网络事件进行根因分析以及进行故障的自动检测和预防奠定了重要的基础。本文首先分析日志的基本结构,然后从 3 种不同的角度(基于统计模板提取模型、基于标签识别树模板提取模型和基于在线模板提取模型)对原始日志进行模板提取研究。3 种模型均可以在无领域知识、无结构文本和不依赖于日志分析平台的前提下,从原始大量的日志中提取日志模板。

本文在第一个模型,即基于统计模板提取模型中,借助 k -均值聚类算法,对 DBSCAN 中的参数进行优化和设置,从而保证 DBSCAN 聚类结果的准确性;在第二个模型,即基于标签识别树模板提取模型中,优化了 BFS 过程,将重复切词的工作优化为一次完成,并巧用集合的运算方式提高了提取日志模板的效率;在第三个模型,即基于在线模板提取模型中,首先指出了前两个模型的不足之处,然后弥补不足,不但细化了单词的分词粒度,还将提取模板过程优化为在线实时更新模板。最后利用互联网公司实际生产的数据进行检验评估,得出基于标签识别树模型提取的日志模板准确率最高的结论。

参考文献

- [1] 王兆丰. 一种基于 k -均值的 DBSCAN 算法参数动态选择方法[J]. 计算机工程与应用, 2017, 53(3): 80-86.
- [2] WANG T, SRIVATSA M, AGRAWAL D, et al. Learning, Indexing, and Diagnosing Network Faults[C]//Proc. of KDD. 2009.
- [3] WANG T, SRIVATSA M, AGRAWAL D, et al. Spatio-temporal Patterns in Network Events[C]//Proc. of CoNEXT. 2010.
- [4] QIU T, GE Z, PEI D, et al. What Happened in my Network? Mining Network Events from Router Syslogs[C]// Proc. of IMC. 2010.
- [5] OLINER A, GANAPATHI A, XU W. Advances and Challenges in Log Analysis[J]. Communications of the ACM, 2012, 55(2): 55-61.
- [6] KIMURA T, ISHIBASHI K, MORI T, et al. Spatio-temporal Factorization of Log Data for Understanding Large-scale Network Events[C]//Proc. INFOCOM. 2014.
- [7] XIE Y L, YU F, ACHAN K, et al. Spamming botnets: Signatures and characteristics[C]//Proc. ACM SIGCOMM. 2008.
- [8] KIMURA, WATANABE A, TOYONO T, et al. Proactive Failure Detection Learning Generation Patterns of Large-scale Network Logs[C]// 2015 11th International Conference on Network and Service Management(CNSM). IEEE, 2015: 8-14.
- [9] 庄军, 郭平, 周杨. 路由器日志序列模式挖掘[J]. 计算机科学, 2005, 32(11): 179-181.
- [10] JIA W H, KAMBER M, PEI J. 数据挖掘: 概念与技术(第 3 版)[M]. 机械工业出版社, 2012: 306-309.
- [11] 张曼琪. 基于前缀树的日志模式聚类[D]. 上海: 华东理工大学, 2013.

¹⁾ https://en.wikipedia.org/wiki/Rand_index