Journal of Security and Safety Technology 安防技术, 2016, 4(4), 45-50
Published Online December 2016 in Hans. http://www.hanspub.org/journal/jsst
http://dx.doi.org/10.12677/jsst.2016.44007

Hans汉斯

Improvement of Apriori Algorithm in Intrusion Detection System

Xin Wang

School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu
Email: zixinshinian@163.com

Received: Nov. 28th, 2016; accepted: Dec. 16th, 2016; published: Dec. 19th, 2016

Copyright © 2016 by author and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
http://creativecommons.org/licenses/by/4.0/
Open Access

Abstract

With the development of the Internet, the network security situation is becoming increasingly grim. Network security problems occur frequently. These security problems appear in all walks of life. The harm caused is more serious. And for companies who face a lot of data every day, how to detect a large number of data in the virus is a big challenge. The association rules in data mining play an important role in distributed computing, which is also suitable for the invasion of virus detection. This Apriori algorithm to association rules in data mining in intrusion detection system is improved, which makes it applicable to the invasion of virus detection.

Keywords

Intrusion Detection, Data Mining, Distributed Computing, Apriori Algorithm

Apriori算法在入侵检测系统中的改进

王　鑫

南京邮电大学通信与信息工程学院，江苏　南京
Email: zixinshinian@163.com

收稿日期：2016年11月28日；录用日期：2016年12月16日；发布日期：2016年12月19日

文章引用：王鑫. Apriori 算法在入侵检测系统中的改进[J]. 安防技术, 2016, 4(4): 45-50.
http://dx.doi.org/10.12677/jsst.2016.44007

摘　要

随着互联网的发展，网络安全形势日趋严峻，网络安全问题频发，这些安全问题出现在各行各业，造成的危害越发严重。而对于每天面临大量数据量的公司而言，如何在众多数据中检测出入侵病毒是个很大的挑战。用于数据挖掘的关联规则在分布式计算中发挥重要的作用，它们同样也适用于对病毒的入侵检测，本文将对数据挖掘中关联规则产生的Apriori算法在入侵检测系统下进行改进，使得它适用于对病毒的入侵检测。

关键词

入侵检测，数据挖掘，分布式计算，Apriori算法

1. 引言

Apriori 算法是数据挖掘中关联规则产生的一种经典算法，其过程为找出数据库中相关联的频繁项集，然后根据频繁项集产生的强关联规则，根据实际应用设定阈值，筛选出最终结果[1]。实践表明，Apriori 算法处理大数据时存在大量 I/O 操作和 CPU 的开销，使得运算效率出现瓶颈，这制约了其在数据挖掘中的应用性[2]。本文对 Apriori 算法进行改进，使其适用于入侵检测系统中，同时克服存在的问题。分析改进后性能，使其具有良好有效性。

2. 传统 Apriori 算法

Apriori 算法是关联规则挖掘频繁项集算法，其核心思想是通过两个阶段挖掘频繁项集，分别是生成候选集以及封闭的向下检测。挖掘步骤总体上分为两个部分，首先根据支持度生成所有频繁项集，然后在此基础上根据置信度生成关联规则[3]。

2.1. Apriori 传统算法步骤

Apriori 算法挖掘步骤如下：Apriori 算法采用迭代方法进行逐层搜索，搜索“1-K 项目集”。首先，找到“频繁 1 项集”，这是表示为 L1 的集合。L1 被用来寻找“频繁 2 项集”L2，再通过 L2 找到 L3。以此类推，直到没有找到的“K 项集”。每次找到 Lk 都需要扫描一次原始数据。其核心思想是：连接过程和剪枝过程。连接过程使用自连接，通过字母顺序连接并确保前 K-2 项相同。修剪过程是让任何频繁项集的所有非空子集也必须是频繁。反之，如果一个非空子集不是频繁集，那么候选集肯定是不频繁，所以可以从 CK 删除它[4]。

具体步骤如下：

1) 产生每个频繁集 L 的所有非空集；

2) 对于每个非空集合 L 的 S，如果 $P(L)/P(S) \geq min\_conf$，则输出规则 S→L-S 图 1 为一个执行 Apriori 算法的过程，最小支持度为 2。

第一步骤中该算法的第一次迭代后，事务数据库将进行扫描，计算出包含在 D 中的每个项目的数目以生成一组候选 1 项集 C1。

第二步骤中，根据所述最小支持度的设置在 C1 的基础上来确定频繁 1 项集 L1。

第三步候选 2 项集 C2 由 L1 生成，然后对 C2 包含的数据项集进行扫描计数。

第四步，根据最小支持度的设置，从候选集 C2 中生成频繁集 L2。

第五步是由频繁2项集L2生成候选3项集C3，设置生成候选3项集的C3 = {{1,2,3}，{1,3,5}，{2,3,5}}，根据 Apriori 修剪的性质：所有的频繁项目集的任意自己都是频繁项目集，{1,2,3}的一个子集{1,2}，不包含在频繁 2-项集 L2 中，因此{1,2,3}被删除。项集{1,3,5}的{1,5}也不包含在 L2 中，所以删除{1,3,5}，而{2,3,5}项目集所有自己均为 L2 的元素，它被保留下来(图 1)。

3. Apriori 在入侵检测中的改进

传统的 Apriori 算法通过迭代的方法生成候选集。一些人通过减少候选项集,减少扫描时间改善算法效率。但是 Apriori 算法的性能瓶颈主要原因是自连接操作的所需要的迭代[5]。本文使用了一种基于非迭代的改进 Apriori 算法，主要应用于 IDS 告警日志分析。由于在 IDS 检测中较少的属性所产生的关联规则往往没有价值，更有价值的是那些通过多维属性得到的关联规则。

3.1. Apriori 算法改进及计算步骤

改进后的 Apriori 算法步骤如下：定义一个函数 count(X)，以此表示要获得项集 X 需要进行交集运算的次数。令 X 的支持度为 n，如果有任何两个项目之间有交集，交集的总数为 count(x)。s(X)与 count(X)之间的关系应为： $\mathrm{count}(x) = C^2_{S(x)}$。

综上所述，我们可以由项集的 count 计数函数得到它的支持度而不用每次重新扫描一次数据集。

算法总体分为如下几步：

第一步是转化阶段，为诸如协议类型，警报类型，以及攻击类别建立索引，以便简化交集操作[6]。

第二步是集合相交阶段，在交易之间做交集，以获得最大项集。如果结果被发现在候选项集的列表中，该计数器+1，否则，结果被插入到列表，计数器被重置为 1，计数器表示用于该事务的支持度。

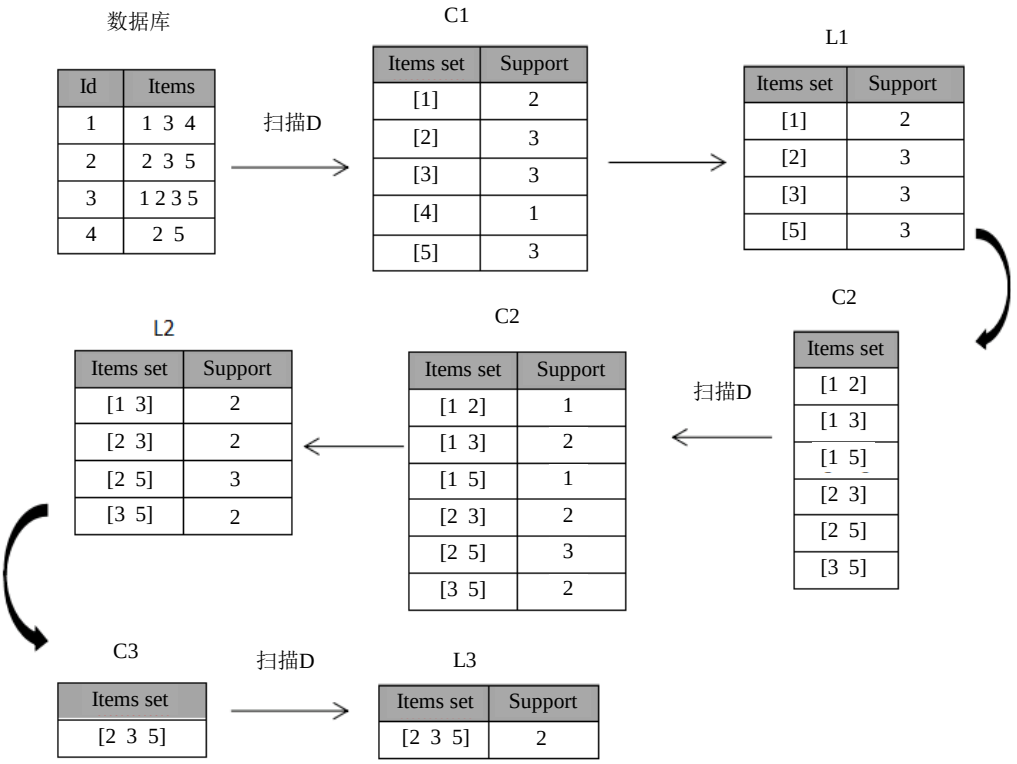

Figure 1. Apriori algorithm example
图 1. Apriori 算法实例

第三步是获得所有的频繁项集。在列表中的计数器被转换成支持度的值并通过比较找出最低支持度以上的项集，则其为频繁项集。

第四步为生成关联规则。在我们的系统中，关联规则只涉及攻击类，算法只生成形如 X→Y 的规则，其中 Y 是攻击行为。如果一个关联规则的置信度大于或等于 min_conf，则为强关联规则。

3.2. Apriori 算法改进后计算实例

本文通过一个实例来说明改进的 Apriori 算法的性能优势。若有数据集合如表 1 所示。

TID 为事务 id，Items 为项集，ItemsC 为在计算交集时需要进行比较的项集，Count 为项集支持的数量，Lenth 为最大交集元素个数，Flag 为标志位,标识该项集是否被删除。算法的具体步骤如下：输入为事务数据集合 D 以及最小支持数 min_sup，设置为 4，输出为最大频繁项集 MFS (MaxFrequentSet)，记为 L 首先对整个数据集合进行扫描，

```
for(each i,j i<n&&j<m){
    if(item j ∈TID i)
        convert(item j); //对上文所述的三个告警日志中的属性进行转换
}
```

然后为每个 item 建立一个列表 DList，列表中存储包念了该 Item 的事务 ID，即 TID，DList 与 itemList 如表 2 所示。

根据表 3 可以首先计算出非频繁 1-项集{4,5,6,7,8}，然后进行第二次扫描数据库，根据 il[]，i2[]，i3[]的数据，求 T1 与其他事务 T2，T3，T5，T6，T8，T9，T10 的交集得到频繁项集 Fl = {i2,i3}，支持数量为 6，因此将 i2，i3 加入频繁项集 FS，并且由于 T6 与 T10 中去除了 NFS 中的项后仅剩余一项，因此直接将其从数据集合中删除，然后更新 DList 如表 4。

由于 SC = 1 + 1 + 1 = 3，算法继续，接着求 T2 中的项的最大频繁项集，因为 T2 与 T1 产生了最大频繁项集 F1，只需比较 T1 比较的项即可，同理可知计算 T3，T5，T8，T9 时也仅需与 T1 比较的项进行交集，同时 F1 是在比较的过程中不断更新的，T2 比较完后 F1 仍为{i2,i3}，略去这一步骤，在进行第三次扫描时，T3 比较完后 F1 更新为{i1,i2,i3}，项集支持数量为 4，然后在 FS 中加入新的最大频繁项集{il,i2,i3}:4，然后更新 Dlist 如表 5。

Table 1. Data set D
表 1. 数据集合 D

TID	Items	ItemsC	Count	Lenth	Flag
1	i1 i2 i3 i4 i5 i6		1		
2	i2 i3		1		
3	i1 i2 i3 i4		1		
4	I3 i4		1		
5	i2 i3		1		
6	i3 i7		1		
7	i1 i4		1		
8	i1 i2 i3 i5		1		
9	i1 i2 i3 i6		1		
10	i3 i8		1		

Table 2. Dlist
表 2. Dlist

TID	Items	ItemsC	Count	Lenth	Flag
1	i1 i2 i3 i4 i5 i6		1		
2	i2 i3		1		
3	i1 i2 i3 i4		1		
4	I3 i4		1		
5	i2 i3		1		
6	i3 i7		1		
7	i1 i4		1		
8	i1 i2 i3 i5		1		
9	i1 i2 i3 i6		1		
10	i3 i8		1		

Table 3. Item list
表 3. Item list

Item	TIDS
i1	1 3 7 8 9
i2	1 2 3 5 8 9
i3	1 2 3 5 6 8 910
i4	3 4 7
i5	1 8
i6	1 9
i7	6
i8	10

Table 4. Updated DList
表 4. 更新后的 DList

TID	Items	ItemsC	Count	Lenth	Flag
1	i1 i2 i3 i5 i6		1	0	0
2	i2 i3	i1	1	2	1
3	i1 i2 i3 i4	i1	1	2	1
4	I3 i4		1	0	0
5	i2 i3	i1	1	2	1
6			1		−1
7	i1 i4		1	0	0
8	i1 i2 i3 i5	i1	1	2	1
9	i1 i2 i3 i6	i1	1	2	1
10			1		−1

Table 5. Second Updated Dlist
表 5. 第二次更新后的 Dlist

TID	Items	ItemsC	Count	Lenth	Flag
1	i1 i2 i3 i5 i6		1	0	0
2	i2 i3		1	2	0
3	i1 i2 i3 i4	i1	1	3	1
4			1	0	−1
5			1	3	−1
6			1		−1
7			1	0	−1
8			1	3	−1
9			1	3	−1
10			1		−1

此时 SC = SC + T4.count + T5.count + T7.count + T8.count + T9.count = 10 > 10 − 4 = 6，因此算法结束，在 FS 中找出最大频繁项集为{il,i2,i3}，支持数量为 4，支持度为 40%。对比该实例的算法与经典的 Apriori 算法实现，显然可以看出改进的算法只扫描了 3 次数据集合，并且第 2，3 次都只扫描了数据集合的一部分，因此效率显著提高。

4. 结束语

本文基于 IDS 检测中较少的属性所产生的关联规则往往没有价值该原则，使用了一种基于非迭代的改进 Apriori 算法，对传统算法存在的额 I/O 负载大，产生过多的候选项目集等缺点进行改进，极大提高效率，减少对数据库的访问量。改进后的算法实现通过多维属性得到关联规则适用于 IDS 告警日志分析。为了应对大数据的分布式计算，需要对该算法进行进一步的改进，使其可以很好的适应大数据环境下的分布式入侵检测问题，在海量数据中快速而准确的找出入侵源头，维护系统安全，使其更加适用于分布式计算平台诸如 Hadoop 中的 Map、Reduce 等的计算过程。

参考文献 (References)

[1] 薛严冬, 韩秀玲, 戴尚飞, 等. 基于 Snort 的分布式协作入侵检测系统[J]. 计算机工程, 2010, 36(19): 165-167.

[2] 刘东鑫, 刘国荣, 王帅, 等. 面向企业网的 APT 攻击特征分析及防御技术探讨[J]. 电信科学, 2013, 29(12): 158-163.

[3] Maatta, M. and Raty, T. (2012) Automatic Creation of Models for Network Intrusion Detection. 2012 *Computing, Communications and Applications Conference*, 11-13 January 2012, 231-237. https://doi.org/10.1109/ComComAp.2012.6154805

[4] Mitchell, R. and Chen, I.-R. (2013) Effect of Intrusion Detection and Response on Reliability of Cyber Physical Systems. *IEEE Transactions on Reliability*, **62**, 199-210. https://doi.org/10.1109/TR.2013.2240891

[5] Lei, J.Z. and Ghorbani, A.A. (2012) Improved Competitive Learning Neural Networks for Network Intrusion and Fraud Detection. *Neurocompnting*, **75**, 135-145. https://doi.org/10.1016/j.neucom.2011.02.021

[6] The White House (2011) International Strategy for Cyberspace. http://www.whitehouse.gov/sites/default/files/rss_viewer/international_strategy_for_cyberspace.pdf

期刊投稿者将享受如下服务：

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击：http://www.hanspub.org/Submission.aspx
期刊邮箱：jsst@hanspub.org