

文章编号:1001-9081(2005)01-0144-03

## AdaBoost 方法在入侵检测技术上的应用

郭红刚<sup>1,2</sup>, 方 敏<sup>1,2</sup>

(1. 综合业务网国家重点实验室, 陕西 西安 710071;

2. 西安电子科技大学 计算机学院, 陕西 西安 710071)

(ghg2001@tom.com)

**摘 要:**详细介绍了 AdaBoost 方法的原理与算法, 以及如何将 AdaBoost 方法应用于入侵检测的核心技术——入侵分析技术中, 并且用实验数据对该方法进行验证, 同时将 AdaBoost 的测试结果与基分类器 BP 的测试结果进行比较, 得出 AdaBoost 方法能够大幅度提高弱分类器识别率, 以及它适合于 IDS 数据的结论。

**关键词:**入侵检测; Boosting; AdaBoost; 分类器; 样本; 训练; 测试

**中图分类号:** TP277 **文献标识码:** A

## Application of AdaBoost method in IDS

GUO Hong-gang<sup>1,2</sup>, FANG Min<sup>1,2</sup>

(1. National key laboratory of integrated services networks, Xi'an Shaanxi 710071, China;

2. Institute of Computer Science, Xidian University, Xi'an Shaanxi 710071, China)

**Abstract:** Firstly, the principle and algorithm of AdaBoost method were described in detail. Secondly, the application of this method to intrusion analytical technique that was the core of the technique of intrusion detection was introduced. Finally, how to verify this method with experimental data was shown. Meanwhile, compared with that of base classifiers BP, the test result of AdaBoost method could remarkably increase the accuracy of weak classer and was fit for the IDS data set.

**Key words:** intrusion detection; Boosting; AdaBoost; classer; sample; train; test

### 0 引言

入侵检测是对计算机系统攻击行为的检测, 它是一种积极主动的安全防护技术, 提供了对内部攻击、外部攻击和误操作的实时保护。入侵攻击可以定义为试图渗透系统或者绕过系统安全策略获取信息, 更改信息或者中断系统的正常运行的活动。入侵检测系统(Intrusion Detection System)不但要能实时监控系统的活动、及时发现攻击行为, 而且要能够准确的识别其攻击类型, 只有在清楚对方的攻击方法后, 才能采取有效的措施减少攻击造成的危害。于是对攻击类型的识别成为一个热门的研究课题, 本文介绍用 AdaBoost 方法构造多分类器对攻击对象进行识别。

### 1 Boosting 方法概述

#### 1.1 基本概念介绍

**分类器(classer):**将一些已知类别的样本输入计算机, 分类算法通过学习这些样本总结并生成相应的规则, 这些规则具有通用性, 使用它们可以解决某一类的问题。

**样本:**所研究问题的实例, 如:

一个网络上的数据包, 一张人脸图像, 一条雷达的信息, 一个病毒代码。

**训练:**采用某种方法, 用已知属性的样本作为输入, 得到相应规则的过程。

**训练集:**由已知属性的样本组成的集合, 作为训练过程的输入数据。

**测试集:**由已知属性的样本组成的集合, 作为测试过程的输入数据。

**测试:**学习机对样本做出的判断, 即是否符合需要判定的事实。

**弱分类器(weak classer):**对一定分布的训练样本给出弱假设判断(仅仅强于随机猜测)。

**强分类器(strong classer):**能够给出稳定的高的识别率的分类器(最大程度上符合实际情况: almost perfect expert)。

#### 1.2 Boosting 方法概述

Boosting 方法是一种框架算法, 它主要是通过对样本集的操作来生成一系列的分类器。它可以用来提高其他分类算法的识别率, 也就是, 将其他的分类算法放于 boosting 框架中, 通过 boosting 框架对训练样本集的操作, 得到不同的训练样本集, 用该样本集去训练分类器, 每得到一个样本集就用该算法在该样本集上产生一个分类器, 这样在给定轮训次数  $n$  后, 就可产生  $n$  个分类器, 然后 Boosting 框架算法将这  $n$  个分

收稿日期: 2004-06-08; 修订日期: 2004-08-23 基金项目: 综合业务网国家重点实验室开放基金资助项目(ISN6-7)

作者简介: 郭红刚(1973-), 男, 陕西商州人, 硕士, 主要研究方向: 计算机网络安全与入侵检测; 方敏(1965-), 女, 湖南人, 副教授, 博士, 主要研究方向: 网络安全及数据融合。

类器进行加权融合,产生一个最后的结果分类器,在这  $n$  个分类器中,每个单个的分类器的识别率不一定很高,但他们融合后的结果有很高的识别率,这样便提高了该分类器的总体识别率。在产生单个的分类器时可用相同的算法,也可用不同的分类算法,这些算法一般是弱的分类算法,如神经网络,决策树等。

## 2 AdaBoost 算法

AdaBoost 是 boosting 算法家族中的一种,AdaBoost 主要是在整个训练集上维护一个分发权值向量  $D(x)_t$ ,用赋予权重的训练集产生分类假设  $H_t(x)$ ,计算它的错误率,用得到的错误率去更新分发向量  $D(x)_t$  对错误分类的样本分配更大的权值,正确分类的样本赋予更小的权值。每次更新后用相同的方法产生新的分类假设,这些分类假设的序列构成多分类器。对这些多分类器用加权的方法进行融合,最后得到决策结果。这种方法不要求单个的分类器有高的识别率,但是经过这些多个分类器融合的决策结果有高的识别率。

### 2.1 算法过程

输入:  $S = \{(x_1, y_1), \dots, (x_i, y_i), \dots, (x_n, y_n)\}, x_i \in X, y_i \in Y$ ; 循环数为  $T$

初始化  $D = \left(\frac{1}{n}, \dots, \frac{1}{n}\right)$

for  $t = 1, \dots, T$ :

1. 使用分发权值向量  $D_t$  训练弱分类器  $h_t = R(x, y, D_t)$ ;  $R$  为一弱的分类算法

2. 计算错误率  $e = \sum (h_t(x_i) \neq y_i) D_t$  (1)

3. if  $e \leq 0.5$ , break;

4. 计算分类假设的权值  $h_t: \omega_t \in \omega$

5. 更新权值:  $D_{t+1}(i) = D_t(i) \times F(e)$  (2)

其中  $F(x)$  为更新函数,它以该次得到的分类器的错误率  $e$  为自变量。

输出最后的分类器

$$H(x) = \arg \max_{y \in Y} \left( \sum_{t: h_t(x) = y} \omega_t \right) \quad (3)$$

在上面的算法中:

(1)  $x_i \in X, y_i \in Y, x_i$  表示样本属性组成的向量,  $y_i$  表示该样本的类别标签。

(2)  $D$  为样本的分发权值向量:

没有先验知识的情况下,初始的分布应为等概率分布,也就是训练集如果有  $n$  个样本,每个样本的分布概率为  $1/n$ ;

每次循环后提高错误样本的分布概率,分错的样本在训练集中所占权重增大,使得下一次循环的弱学习机能够集中力量对这些错误样本进行判断;

$D_t$  总和应该为 1。

(3)  $\omega_t$  为分类器的权值:准确率越高的分类器权重越大。

### 2.2 分类器的集成

如何将得到的多个分类器进行融合,我们采用的方法是用权值矩阵的方法进行处理:

首先构造一个以行数  $m$  (代表有  $m$  个类别),列数  $n$  (代表  $n$  个样本) 的零矩阵,假设生成了  $T$  个分类器,在测试样本时,将用这  $T$  个分类器分别对每一个测试样本进行测试,假设现在用第  $i$  个分类器对第  $j$  样本进行测试,测试结果为第  $u$  类,那么将第  $i$  个分类器的权值  $w_i$  加到该矩阵的第  $u$  行第  $j$  列的元素  $a_{uj}$  上,这样以次用  $T$  个分类器测试完  $n$  个样本,则最后得到下面权值矩阵:

$$\begin{bmatrix} a_{11} & \cdots & a_{1j} & \cdots & a_{1n} \\ \vdots & & \vdots & & \vdots \\ \vdots & & a_{ij} & & \vdots \\ \vdots & & \vdots & & \vdots \\ a_{m1} & \cdots & a_{mj} & \cdots & a_{mn} \end{bmatrix} \quad (4)$$

行数  $m$  代表有  $m$  个类别,列数  $n$  代表  $n$  个样本,矩阵中的元素代表  $a_{ij}$  代表分类器将第  $j$  个样本分类到第  $i$  类的权值大小,我们从该权值矩阵中可以得到第  $j$  个样本的类别,即该  $j$  列中权值最大的行代表的类别数即为该样本的类别。

## 3 Boosting 的理论分析

Freund 与 Schapire 分析并证明了 Boosting 的最终分类器的训练误差最多为:

$$\prod_i \left[ 2 \sqrt{\varepsilon_i (1 - \varepsilon_i)} \right] = \prod_i \sqrt{1 - 4\gamma_i^2} \leq e^{(-2 \sum_i \gamma_i^2)} \quad (5)$$

其中  $\varepsilon_i$  为  $h_i$  的训练误差,  $\gamma_i = 1/2 - \varepsilon_i$ ,从该式可以看到只要学习算法略好于随机猜测,训练误差将随  $i$  以指数级下降。在 AdaBoost 之前提出的 Boosting 算法要求事先知道  $\gamma_i$  的下限,但这一点在实际问题中很难做到。而 AdaBoost 却没有这样的要求。

Freund 与 Schapire 用 VC 维从训练误差的角度分析 Boosting 的泛化误差 (generalization error), VC 维是学习算法的复杂度及其学习能力的度量,设训练例为  $n$  个,弱学习算法的 VC 维为  $d$ ,循环次数为  $T$ ,其泛化误差最多为:

$$\bar{p}_t(H(x) \neq y) + O\left(\sqrt{\frac{Td}{n}}\right) \quad (6)$$

$\bar{p}_t$  表示对训练集的经验概率。该公式表明若训练轮数过多,Boosting 将发生过配。因此选择恰当的训练轮数对分类器的精度有很大的影响,我们对特定的样本集应该计算它的最好的训练轮数,以提高分类器的精度。

## 4 AdaBoost 在 IDS 中的应用

### 4.1 IDS 数据集的特点

在实验中使用的数据全部来源于 KDD99。这样有利于在相同的基准上将我们的实验结果与其他的方法相比较。KDD99 使用的是 MIT Lincoln Labs98 的数据,该数据集来源

于一个模拟的局域网,共有 24 种攻击,其中有 14 种未在训练集中出现。攻击主要分为四大类:

DOS:拒绝服务攻击。例如 syn\_flood, land, Neptune, pod, smurf, teardrop;

R2L:远端未授权进入。例如 ftp\_write, imap, multihop, phf;

U2R:本地特权用户进入。例如 ipsweep, loadmodule, perl, rootkit;

Probing:监测或探测。例如 port\_scanning, nmap, potsweep, satan。

数据集中每条数据(一个样本)有 41 个特征。

#### 4.2 AdaBoost 对 IDS 数据集的分类处理

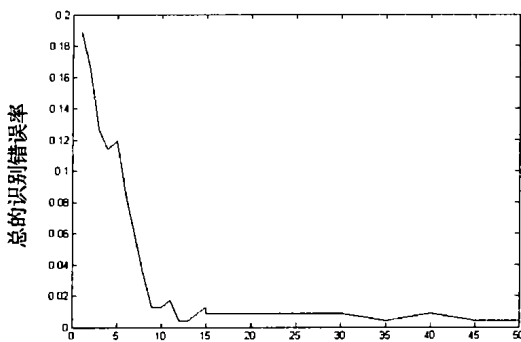
数据集的选定:首先从 KDD99 中取出 1 183 条数据做实验数据,该集合包含 8 类数据类型: normal, snmpgetattack, snmpguess, smurf, apache2, mscan, neptune, ipsweep 其中 1 类为正常数据,其余 7 类为攻击数据。我们用集合的 80% 做训练样本,20% 做测试样本。

从前面的算法描述中我们已经知道,在用 AdaBoost 方法时,必须选定一弱分类算法,将其放于 Boosting 框架之中,在对 IDS 数据集的分类处理时,我们用神经网络(BP)作为弱分类算法(基分类算法)。

根据模式识别理论,并不是将已经提取出的样本的所有特征直接使用。一般是较多使用对分类有利的特征,而避免使用对正确分类造成干扰的特征。因此我们对样本用主成分分析法(PCA)先做特征提取,然后再用 AdaBoost 方法对其进行处理。前面已经提到,选择恰当的训练轮数对分类器的精度有很大的影响,因此在处理中我们首先得出对于 IDS 数据的最佳训练轮数。

#### 4.3 试验结果

从图 1 中可以看到,在分类器数目为 12 时,测试错误率最低为:0.0042,因此它的最佳训练轮数为 12。



最小错误率为: 0.0042 分类器的数目为: 12

图 1 总的错误率随分类器个数的变化曲线图

表 1 该 AdaBoost 方法与基分类器(BP)神经网络的识别结果的比较(各类别的错误率)

方法	类 别							
	Normal 24 条	Snmpgetattack 26 条	Snmpguess 32 条	Smurf 29 条	apache2 35 条	Mscan 34 条	Neptune 26 条	Ipsweep 31 条
BP	0.1667	0.0385	0.3125	0.0345	0.1429	0.2353	0.2692	0.1613
AdaBoost	0.0417	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

我们选 12 作为它的训练轮数,用前面得到得 80% 数据集进行训练,生成 12 个分类器,用 20% 的数据集进行测试结果如表 1、表 2 所示。

表 2 总的测试结果与测试时间的比较

	总的样本集的错误率	总的测试时间/秒
BP	0.1730	0.5150
AdaBoost	0.0042	0.8590

从上面的结果中,我们可以明显的看到,AdaBoost 提高了弱预测算法 BP 的预测精度。同时也说明,AdaBoost 方法对 IDS 的数据是适合的,因为它的错误率只有 0.0042,完全满足测试精度。从实时性来说,AdaBoost 的测试时间只比 BP 的测试时间多 0.34 s,从绝对值来看,也很短,可以达到实时性的要求。

## 5 结语

除了对 IDS 数据的实验,我们还对公共数据集,如: iris, glass 等做了实验,同样得到非常高的正确率,同时大量关于 AdaBoost 的实验和应用也证明了这种方法的有效性。因此,这使得在实际应用中,我们可以不再寻找通常很难获得的预测精度很高的强学习算法,只需找出一个精度略好于随机预测的弱学习算法,就可以通过 AdaBoost 方法大幅度提高弱预测算法的准确率。目前,AdaBoost 方法仍有许多方面有待于进一步的研究。

#### 参考文献:

- [1] 闫巧, 喻建平, 谢维信. 基于系统调用的神经网络异常检测技术[J]. 计算机工程, 2001, 27(9).
- [2] LEBANON G, LAFFERTY J. Boosting and maximum likelihood for exponential models[A]. Advances in Neural Information Processing Systems[C], 2001.
- [3] MACLIN R. Boosting classifiers regionally[A]. Proceedings of the Fifteenth National Conference on Artificial Intelligence[C]. Madison, WI., 1998. 700 - 705.
- [4] BRIEMGJ, BENEDIKTSSONTA, SVEINSSONJR. Boosting, Bagging, and Consensus Based Classification of Multisource Remote Sensing Data[A]. Proceedings of the Second International Workshop on Multiple Classifier Systems table of contents[C], 2001. 279 - 288.
- [5] SCHAPIRE RE. The boosting approach to machine learning an overview[Z]. MSRI Workshop on Nonlinear Estimation and Classification[C]. Berkeley, CA, Mar 2001.
- [6] RUTA D, GABRYS B. Application of the evolutionary algorithms for classifier selection in multiple classifier systems with majority voting[A]. Proceedings of the MCS 2001 Workshop[C]. Cambridge, UK, 2001. 399 - 408.
- [7] FREUND Y, SCHAPIRE RE. A decision-theoretic generalization of on-line learning and an application to boosting[J]. Journal of computer and system sciences, 1997, 55(1): 119 - 139.