第37卷 第16期
Vol.37 No.16

计 算 机 工 程
Computer Engineering

2011年8月
August 2011

• 安全技术 •

文章编号：1000—3428(2011)16—0117—03 文献标识码：A 中图分类号：TP309

基于中间分类超平面的SVM入侵检测

牟 琦，毕孝儒，龚尚福，厍向阳

(西安科技大学计算机学院，西安 710054)

摘 要：在网络入侵检测中，大规模数据集会导致支持向量机(SVM)方法训练时间长、检测速度慢。针对该问题，提出一种基于中间分类超平面的SVM入侵检测方法。通过对正常和攻击样本的聚类分析，定义聚类簇中心的边界面接近度因子，实现对标准SVM二次式的改进；用簇中心对其训练，获取一个接近最优超平面的中间分类超平面；确定距离阈值，以选取潜在支持向量，实现训练样本的缩减。在KDDCUP1999数据集上进行实验，结果表明，与聚类支持向量机方法相比，该方法能简化训练样本，提高SVM的训练和检测速度。

关键词：中间分类超平面；样本缩减；潜在支持向量；支持向量机；入侵检测

SVM Intrusion Detection Based on Middle Classification Hyperplane

MU Qi, BI Xiao-ru, GONG Shang-fu, SHE Xiang-yang

(School of Computer, Xi'an University of Science and Technology, Xi'an 710054, China)

【Abstract】In network intrusion detection, aiming to the problem that high dimensional and large network data results in long training time and low detecting speed of Support Vector Machine(SVM), this paper proposes an approach for SVM intrusion detection based on middle classification hyperplane. Based on clustering normal and attack training samples, by defining approaching degree of boundary surface of every clustering center, quadratic expression of standard SVM is improved; improved SVM is trained with clustering centers to obtain a middle classification hyperplane; then training samples are reduced by defining distance threshold to obtaining Possible Support Vectors(PSV). Experimental results on KDDCUP1999 data-set show that the method is more effective than cluster SVM in reducing training samples and improving the training and detecting speed of SVM .

【Key words】middle classification hyperplane; sample reduction; Possible Support Vectors(PSV); Support Vector Machine(SVM); intrusion detection

DOI: 10.3969/j.issn.1000-3428.2011.16.039

1 概述

针对支持向量机(Support Vector Machine, SVM)[1]在大规模网络数据入侵检测中训练和检测速度慢、实时性差的问题。文献[2]将无监督聚类算法用于入侵检测SVM训练样本的化简；文献[3]将k-最近邻方法与k-means算法相结合，实现SVM训练样本的缩减。以上聚类支持向量机(Cluster SVM)入侵检测方法均采用聚类簇中心代替整个训练样本完成SVM训练，虽然提高了SVM方法的训练和检测速度，但由于舍弃了相当数量的有效支持向量，明显地降低了SVM分类精度。

本文提出一种基于中间分类超平面(Middle Classified Hyperplane, MCH)的SVM入侵检测方法。该方法在对正负类样本分别进行聚类的基础上，构造一个中间分类超平面，实现训练样本的简化。

2 基于中间分类超平面的SVM入侵检测方法

2.1 簇中心的边界面接近度因子定义

定义1(类间簇中心距离矩阵) 若正负类样本的聚类簇中心数目分别为C_+, C_-，则称矩阵$\boldsymbol{D}=\{d_{ij}\}$，$0<i\leqslant C_+$，$0<j\leqslant C_-$，为类间簇中心距离矩阵。

定义2(簇中心下标矩阵) 对于正类簇，称矩阵$\boldsymbol{Q}=\{\boldsymbol{q}_i\}$，$1\leqslant i \quad C_+$为其簇中心下标矩阵，其中，$\boldsymbol{q}_i$为一列向量，存储距离矩阵每列按升序排序后的正类簇中心排序前在距离矩阵中的下标；相应地，称矩阵$\boldsymbol{R}=\{\boldsymbol{r}_j\}, 1\leqslant j \quad C_-$为负类簇中心下标矩阵，其中，$\boldsymbol{r}_j$为一行向量，存储距离矩阵每行按升序排序后的负类簇中心排序前在距离矩阵中的下标。

定义3(簇中心的边界面接近度因子) 在取得簇中心下标矩阵基础上，定义：

$$B_i^+=(C_+-Line)/C_+,\quad i=1,2,\mathrm{L}\ ,C_+ \tag{1}$$

为正类簇中心i的边界面接近度因子。其中，$Line$为按行扫描$\boldsymbol{Q}$时，首次遇到簇中心i所在行号。相应地定义：

$$B_j^-=(C_--Column)/C_-,\quad j=1,2,\mathrm{L}\ ,C_- \tag{2}$$

为负类簇中心j的边界面接近度因子。其中，$Column$为按列扫描R时，首次遇到簇中心j所在行号。

2.2 中间分类超平面的构造

采用聚类分析算法[4]对正负类样本分别进行聚类分析后，得到相应的簇中心；并用其对标准SVM训练，获得一过渡性的分类超平面，则称此超平面为中间分类超平面。然而这种构造方法将各个聚类中心同等对待，并未考虑各簇中心重要度，因此，会降低中间分类超平面与最优分类超平面的相似度。本文将每一聚类中心边界面接近度因子以及所包含样本数目两因素作为簇中心重要度，对SVM二次式改进，生成以样本重要性为权值的SVM。并用簇中心对其训练，构

基金项目：陕西省自然科学基金资助项目(2009JM7007)

作者简介：牟 琦(1974－)，女，副教授，主研方向：网络安全，网络集成与数据库；毕孝儒，硕士研究生；龚尚福，教授；厍向阳，副教授

收稿日期：2011-02-18 **E-mail：**bi_xiao_ru@sina.com

造一个尽可能接近最优分类超平面的中间分类超平面，以便后续算法更为有效地提取潜在支持向量(Possible Support Vectors, PSV)。

定义 4(簇中心重要度) 为了反映各簇中心对中间分类超平面不同影响力，定义：

$$\tau_i^+ = B_i^+ \times (Sample\_Num_i^+ / Sample\_Num^+) \tag{3}$$

为正类簇中心 i 重要度；其中，$Sample\_Num_i^+$ 为正类簇 i 中的样本数目；$Sample\_Num^+$ 为正类样本数目；$i = 1,2,\mathrm{L}\ ,C_+$。类似地定义：

$$\tau_j^- = B_j^- \times (Sample\_Num_j^- / Sample\_Num^-) \tag{4}$$

为负类簇中心 j 重要度；其中，$Sample\_Num_j^-$ 为负类簇 j 中的样本数目；$Sample\_Num^-$ 为负类样本数目；$j = 1,2,\mathrm{L}\ ,C_-$。在定义以上簇中心重要度基础上，标准 SVM 最优化问题形式变为：

$$\min_{\boldsymbol{\omega},b,\varepsilon} 1/2\boldsymbol{\omega}^{\mathrm{T}} + C\sum_{i=1}^{l}\tau_i\varepsilon_i \tag{5}$$

$$\text{s.t. } y_i(\boldsymbol{\omega}^{\mathrm{T}}\phi(x_i)+b)+\varepsilon_i \geqslant 1, \varepsilon_i \quad 0,\ i = 1,2,\mathrm{L}\ ,l \tag{6}$$

其中，$\boldsymbol{\omega}^{\mathrm{T}}\phi(x_i)+b=0$ 为所要求解的中间分类超平面；$\boldsymbol{\omega}$ 是该超平面的法向量；b 是该超平面的偏移量；C 是惩罚因子；ε_i 是松弛变量；x_i 和 y_i 分别为正负类样本簇中心及类别标记；τ_i 依据式(3)和式(4)确定。

上式的对偶问题为：

$$\min_{\boldsymbol{\alpha}} Q(\boldsymbol{\alpha}) = \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j k(x_i,x_j) - \sum_{i=1}^{l}\alpha_i \tag{7}$$

$$\text{s.t.} \sum_{i=1}^{l} y_i\alpha_i = 0 \tag{8}$$

$$0 \leqslant \alpha_i \quad \tau_i C,\ i = 1,2,\mathrm{L}\ ,l \tag{9}$$

其中，$\boldsymbol{\alpha}$ 是该对偶问题的解向量；$k(x_i,x_j)$ 是 SVM 的核函数。

通过求解以上对偶问题，可以获得一个与最优超平面更为近似的中间分类超平面。

2.3 邻界簇确定

定义 5(邻界簇) 若训练样本集为 $TrainData$，S 表示分类间隔，且 L_i^+ 为任一正类簇 V_i^+ 的中心 o_i 到中间分类超平面 $g=0$ 的距离，如果 L_i^+ 满足：

$$L_i^+ - S \leqslant \lambda^+,\ i = 1,2,\mathrm{L}\ ,C^+ \tag{10}$$

则称 V_i^+ 簇为正类邻界簇。其中，

$$\lambda^+ = (|TrainData^+|/|TrainData|)\times\eta,\quad 0<\eta\leqslant 1$$

η 为距离阈值；相应地，可以定义负类邻界簇。

邻界簇的确定如图 1 所示。

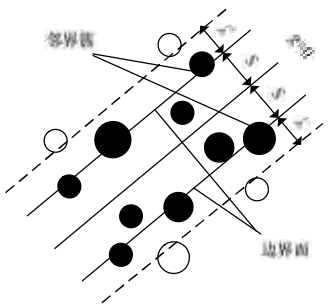

图 1 邻界簇的确定

由图 1 可知，邻界簇(带黑色标记的簇)中的训练样本大多靠近中间分类超平面 $g=0$，极有可能成为最优分类超平面 $g^*=0$ 潜在支持向量。因为 $g=0$ 在很大程度上近似所求的分类超平面 $g^*=0$，则靠近 $g=0$ 的样本也就靠近 $g^*=0$。而其他簇所包含的样本点由于远离 $g=0$，它们对 $g^*=0$ 生成没有影响，因此，可将这些样本从训练集中删除，仅保留从属于邻界簇的样本。

2.4 潜在支持向量的提取

由图 1 可知，虽然邻界簇从整体上看靠近中间分类超平面 $g=0$，但邻界簇所包含的样本点并不都靠近 $g=0$。因此，有必要对其所包含的样本进一步消减。

定义 6(邻界簇样本聚集度) 设正类邻界簇中所有样本到中间分类超平面 $g=0$ 的距离平均值为 dm^+，样本 $\{x_i,y_i\}$ 到 dm^+ 的差值为 v_i^+。则称：

$$CLose\_Degree^+ = 1/\sum_{i=1}^{|N^+|}(v_i^+)^2 \tag{11}$$

为正类邻界簇样本的聚集度。其中，$|N^+|$ 为正类邻界簇中样本数目。相应地，可以定义负类邻界簇样本的聚集度。

定义 7(潜在支持向量) 若 S 表示分类间隔，设正类邻界簇中任一样本 $\{x_i,y_i\}$ 到中间分类超平面的距离 $y_i g(x_i)$ 满足：

$$S-\sigma^+ \leqslant y_i g(x_i) \quad S+\sigma^+,\ i = 1,2,\mathrm{L}\ ,C^+ \tag{12}$$

则称样本 $\{x_i,y_i\}$ 为正类潜在支持向量。类似的，可以定义负类潜在支持向量。其中，$\sigma^+ = Close\_Degree^+ \times \alpha$，$0<a<0.1$ 为提取因子。

潜在支持向量的提取如图 2 所示。

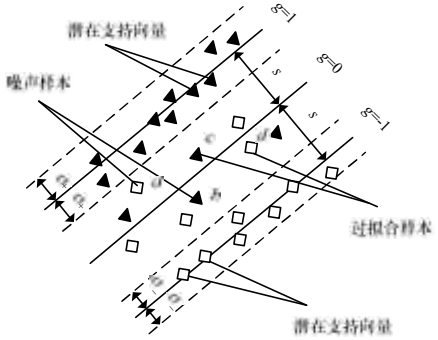

图 2 潜在支持向量的提取

通过计算邻界簇中心每一样本到中间分类超平面的距离，将距离在 $(S-\sigma,S+\sigma)$ 以内的样本(在图 2 中，靠近和恰在 $g=1$ 和 $g=-1$ 超平面上的样本)保留，而舍弃其余样本，实现潜在支持向量的选取。同时由式(12)和图 2 可知，该选取策略不仅删除与 $g=0$ 距离较远的样本点，同时也删除过分靠近 $g=0$ 的过拟合样本点(图 2 中的 c,d 样本点)和噪声样本点(图 2 中的 a,b 样本点)，抑制了 SVM 的过学习，提高了其泛化能力。

2.5 算法描述

算法描述如下：

输入 正常训练样本集 $TrainData^+$，攻击训练样本集 $TrainData^-$，聚类簇数 K，距离因子 η，提取因子 α

输出 入侵检测最优分类超平面 g^*

(1) CenterSet$^+$=ClusterAnalyzing(TrainData$^+$,K)

(2) CenterSet $^-$=ClusterAnalyzing(TrainData$^-$,K)

(3) CenterSet= CenterSet$^+$ ∪ CenterSet$^-$

(4) D=CalculateKernelDistanceArray(CenterSet)

(5) B= CalculateApproachDegree(D)

(6) g=GetM_ClassifyHyperplane(B,C_SampleNum)

(7) B_Cluster=GetBoundarySurfaceCluster(g, η)

(8) for i=1:|B_Cluster|

(9)　s[i]=GetB_ClusterSample(B_Cluster)

(10) ReSample=ReSample∪GetB_S_Sample(s[i])

(11) endfor

(12) g*=SvmTrain(Re_Sample)

算法中主要函数功能如下：

ClusterAnalyzing(*TrainData*, *K*)：函数采用聚类算法对训练样本分析，返回聚类簇中心集 *CenterSet*。

CalculateKernelDistanceArray(*CenterSet*)：函数计算和返回核距离矩阵 ***D***。

CalculateApproachDegree(***D***)：函数计算每一聚类簇中心的边界面接近度因子，返回接近度因子矩阵 ***B***。

GetM_ClassifyingHyperplane(***B***, *C_SampleNum*)：函数计算并返回中间分类超平面 *g*。

GetBoundarySurfaceCluster(g, η)：函数返回值为靠近中间分类超平面的邻界簇集合 *B_Cluster*(根据式(10))。

GetB_ClusterSample(*B_Cluster*)：函数返回邻界簇中每一样本 *B_ClusterSample*。

getB_SurfaceSample(*B_ClusterSample*)：对邻界簇集每一样本 *B_ClusterSample* 根据式(12)判断其是否为潜在支持向量；若是，则返回该样本。

2.6　算法时间复杂度分析

本文算法耗费时间步骤主要有以下 4 个方面(假设训练样本数目为 n,聚类簇数为 K)：

(1)对正负类样本进行聚类分析[4]，其时间复杂度为 $O(InterTimes \times K \times n)$。

(2)由文献[5]可知，标准 SVM 的最坏时间复杂度为 $O(n^{2.2})$，因此通过式(3)和式(4)对计算每一簇中心权重，并对改进 SVM，训练获取中间分类超平面的时间复杂度为 $O((2\times K)^{2.2})$。

(3)对邻界簇中样本进行逐一判别时，其判别次数 t 需根据 η 来确定，其时间复杂度为 t 的函数 $O(f(t))$。

(4)用潜在支持向量(数目为 n_{PSV})对标准 SVM 训练，其时间复杂度为 $O((n_{PSV})^{2.2})$。则整个算法的时间复杂度为：

$$\Theta = O(InterTimes \times K \times n) + O(f(t)) + O((2\times K)^{2.2}) + O((n_{PSV})^{2.2}) \approx O(n)$$

因此，与标准 SVM 的时间复杂度相比，本文算法具有近似线性的时间复杂度和良好的样本扩展性。

3　入侵检测实验与分析

3.1　实验数据集与参数设置

本文实验采用 KDDCUP1999[6]中的 2 个 10%独立子集分别作为训练集与测试集的选取来源，在对选取样本进行预处理的基础上形成 5 组实验数据集。SVM 的核函数采用径向基函数(Radial Basis Function, RBF)，核参数 g 和 C 采用交叉验证寻优方法获取，聚类簇数目为 K=10。

3.2　实验结果比较和分析

未进行样本缩减的 SVM 入侵检测结果如表 1 所示，经 MCH 样本缩减后的 SVM 入侵检测结果如表 2 所示 。

表 1　未进行样本缩减的 SVM 入侵检测结果

实验数据序号	未缩减样本数	支持向量	训练时间/s	检测时间/s	检测率/(%)	误报率/(%)
1	13 740	224	2.094	4.141	93.24	0.25
2	11 651	221	2.203	4.047	97.12	0.22
3	14 695	220	2.094	3.735	93.47	0.22
4	10 980	204	1.765	3.266	91.15	0.20
5	12 763	226	2.187	4.125	97.88	0.26

表 2　经 MCH 样本缩减后的 SVM 入侵检测结果

实验数据序号	缩减后样本数	支持向量	训练时间/s	检测时间/s	检测率/(%)	误报率/(%)
1	192	187	0.047	2.719	93.68	0.18
2	185	179	0.094	2.750	98.26	0.20
3	183	180	0.063	2.843	94.22	0.19
4	161	157	0.015	2.360	91.99	0.15
5	191	190	0.078	2.891	98.86	0.20

可以看出，采用本文提出的 MCH-SVM 算法，在不同实验数据集下提取的支持向量仅为原训练样本的 2%左右，明显减少了训练时间，而且该方法对支持向量也实现了有效化简，平均约减率约为 20%，显著提升了 SVM 检测速度。

不同数据集下 2 种算法的检测率比较如图 3 所示，误报率比较如图 4 所示。可以看出，本文提出的 MCH-SVM 算法的入侵检测率均高于聚类支持向量机 CLU-SVM 算法，且误报率均低于 CLU-SVM 算法，尤其是在第 4 数据集上，其检测率提高近 23%，误报率降低近 46%。

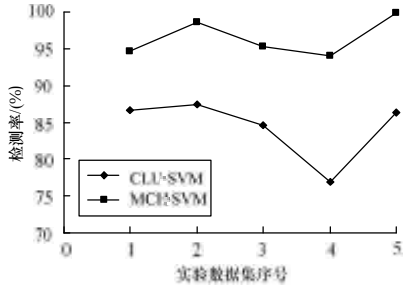

图 3　不同数据集下 2 种算法的检测率比较

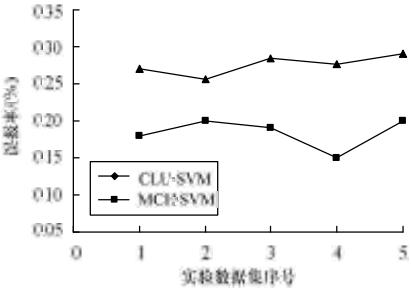

图 4　不同数据集下 2 种算法的误报率比较

4　结束语

针对大规模网络数据致使 SVM 入侵检测方法训练和检测速度过慢的问题，本文提出一种基于中间分类超平面样本缩减的 SVM 入侵检测方法。实验结果表明，该方法能有效缩减训练样本，提高 SVM 方法的入侵检测性能。如何确定邻界簇距离因子 η 以及潜在支持向量提取因子 α 的最佳取值范围，使算法性能更可靠，是下一步研究的方向。

参考文献

[1] Vapnik V. Statistical Learning Theory[M]. New York, USA: Springer, 1998.

[2] 曾志强, 高　济, 朱顺痣. 基于约简 SVM 的网络入侵检测模型[J]. 计算机工程, 2009, 35(17): 132-134.

[3] 邹汉斌, 周学清. 基于聚类的模糊支持向量机入侵检测算法[J]. 情报杂志, 2009, 28(3): 175-178.

[4] 王向阳, 于雁春. 基于改进 K-均值聚类的分形图像编码算法[J]. 计算机科学, 2008, 35(2): 219-222.

[5] 王　磊, 孙世新. 适于大规模数据集的块增量学习算法[J]. 计算机应用研究, 2008, 25(1): 98-100.

[6] KDD99Cupdataset[DB/OL]. [2010-12-07]. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

编辑　顾姣健