Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia Environmental Sciences 11 (2011) 256 – 262
Procedia
Environmental Sciences

A method of SVM with Normalization in Intrusion Detection

Weijun li[1], Zhenyu Liu[2]

[1]*College of Automation Guangdong University of Technology Guangzhou, P. R. China*
[2]*College of Electronic and Information Engineering South China University of Technology Guangzhou, P. R. China*
weijun_lee@163.com, zhenyu.liu@163.com

Abstract

Network intrusion is always hidden in a mass of routine data and the differences between these data are very large. Normalization can help to speed up the learning phase and avoiding numerical problems such as precision loss from arithmetic overflows. Some normalization methods are analyzed and simulated. Experiments results show that the method using SVM with normalization has much better performance compared to the method using SVM without normalization in classing intrusion data of KDD99 and Min-Max Normalization has better performance in speed, accuracy of cross validation and quantity of support vectors than other normalization methods.

Keywords: Intrusion Detection; SVM; Cross Validation ;Min-Max Normalization; Max Normalization

1. Introduction

An intrusion detection system (IDS) inspects all inbound and outbound network activity and identifies suspicious patterns that may indicate a network or system attack from someone attempting to break into or compromise a system [3], [4].

The usual way to detect intrusion is gathering and analyzing information from various areas within a computer or a network. Some data of feature has large value range, such as many times of Maximal and minimum in bytes between the destination and source when normal and attack [9], [12].So detecting the abnormal action is a hard task to find out in hundreds of millions of routine logs.

SVM is a popular way to deal with 2-class questions. It is relatively insensitive to the number of data points and the classification complexity does not depend on the dimensionality of the feature space. So they can potentially learn a larger set of patterns and thus be able to scale better than some methods, such as neural networks.

So normalization is a good way to reduce the difference of the data and improve the speed. Some normalization methods have been brought forward. Because the routine data is very large, normalization method should have simple rules and fast speed. Max Normalization and Min-Max Normalization are

doi:10.1016/j.proenv.2011.12.040

used in the experiments. They compare with the non-normalization method in SVM training. The results prove the performance of Min-Max Normalization has excellent performance [7]-[9], [11].

The remainder of the paper is organized as follows. After the introductory part, Section II introduces the basic concepts and methods of SVM. In Section III, some normalization methods are discussed. In Section IV, experiments with KDD99 intrusion detection data are showed. The results prove the good performance of Scale-Normalization. In Section V our conclusion is proposed.

2. An Overview of SVM

2.1 SVM

An SVM model is a machine learning method that is based on statistical learning theories. SVM classifies data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in the feature space [1], [2], [10].

Given a training set of instance-label pairs $(x_i; y_i), i = 1, \cdots, l$ where $x_i \in R^n$ and $y \in \{1, -1\}^l$, SVM require the solution of the following optimization problem:

$$\min_{\omega,\alpha,\xi} \frac{1}{2}\omega^T \omega + C\sum_{i}^{l} \xi_i \qquad (1)$$

Subject to

$$y_i\left(\omega^T z_i + b\right) \geq 1 - \xi_i,\ \xi_i \geq 0, i = 1, \cdots, l, C > 0.$$

Training vectors x_i are mapped into a higher dimensional space by the function ϕ as $z_i = \phi(x_i)$.

C is the penalty parameter of the error term; it is the upper bound of all variables.

Usually the solution to (1) can be done by solving the following dual problem

$$\min_{\alpha} F(a) = \frac{1}{2} a^T Q a - e^T a \qquad (2)$$

subject to

$$0 \leq a_i \leq C, i = 1, \cdots, l \qquad y^T a = 0,$$

where *e* is the vector of all ones and *Q* is an *l* by *l* positive semi-definite matrix.

$Q_{ij} = y_i y_j K(x_i, x_j)$, where $K(x_i, x_j) = \phi(x_i)^T \phi(x_i)$ is called the kernel function.

The decision function is

$$\operatorname{sgn}\left(\omega^T \phi(x) + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{l} a_i y_i K(x_i, x) + b\right) \qquad (3)$$

where

$$\omega = \sum_{i=1}^{l} a_i y_i \phi(x_i). \qquad (4)$$

Using a Gauss Radial Basis Function as kernel:

$$K(\tilde{x}, \bar{x}) = \exp\left(\frac{-\left\|\tilde{x} - \bar{x}\right\|^2}{2\sigma^2}\right). \qquad (5)$$

The expectation of the test error for a trained machine is therefore:

$$R(a) = \int \frac{1}{2}\left|y - f(x, a)\right| dP(x, y). \qquad (6)$$

When a density $P(x, y)$ exists, $dP(x, y)$ may be written $p(x, y)dxdy$. This is a nice way of writing the true mean error, but unless we have an estimate of what $P(x, y)$ is, it is not very useful.

$$R_{emp}(a) = \frac{1}{2l}\sum_{i=1}^{l}|y_i - f(x_i, a)| \qquad (7)$$

The quantity $R(a)$ is called the expected risk, or just the risk. Here we will call it the actual risk, to emphasize that it is the quantity that we are ultimately interested in. The empirical risk $R_{emp}(a)$ is defined to be just the measured mean error rate on the training set.

No probability distribution appears here. $R_{emp}(a)$ is a fixed number for a particular choice of *a* and for a particular training set (x_i, y_i).

The number of free parameters used in the SVM depends on the margin that separates the data points but not on the number of input features, thus SVM do not require a reduction in the number of features in order to avoid over fitting-an apparent advantage in intrusion detection. Another primary advantage of SVM is the low expected probability of generalization errors.

2.2 Cross Validation

The main methods of predictive accuracy evaluations are:

Re-substitution, Holdout, K-fold cross validation, Leave-one-out.

Leave-one-out is difficult to be used in large training set. Cross validation is a model evaluation method that is better than residuals. K-fold cross validation is one way to improve over the holdout method [6].The problem with residual evaluations is that they do not give an indication of how well the learner will do when it is asked to make new predictions for data it has not already seen. One way to overcome this problem is to not use the entire data set when training a learner. Some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on "new" data [5].

The data set $T = \{(x_1, y_1), \cdots (x_i, y_i)\} \in (X \times Y)^l$, $x_i \in X = R^n$, $y_i \in \{1, -1\}^l$, $i = 1, \cdots, l$ is divided into k subsets $S_1, S_2, \cdots S_k$, $\bigcup_{i=1}^{k} S_i = S$,and $S_i \bigcap S_j = \phi(i \neq j)$. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average error across all k trials is computed.

$$CV_{accuracy} = \frac{\sum_{i=1}^{k} l_i}{l} \qquad (8)$$

where *li* is the error of every *i*th decision function.

The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set k-1 times. The variance of the resulting estimate is reduced as k is increased.

3. Normalization

Normalization is the word coined by E. F. Codd in 1972[7].It is a preprocessing type which plays an important role in classification. Normalizing the input data will help speed up the learning phase. Some kind of data normalization also may be necessary to avoid numerical problems such as precision loss from arithmetic overflows. Attributes with initially large ranges will outweigh attributes with initially smaller ranges, and then dominate the distance measure.

In the feature space normalization is not strictly-speaking a form of preprocessing since it is not applied directly on the input vectors but can be seen as a kernel interpretation of the preprocessing. Normalization in the feature space essentially amounts to redefining the kernel functions of the SVM as it is applied to the unprocessed input vectors.

In intrusion detection data, some features have very large difference, such as 10^9 times between maximal and minimum value in normal and attack. Because SVM calculates the margin of the data, it can be considered as a “distance” classifier. Too large "distance" will bring great calculation in matrix and overflow problem has to be considered in computation.

In some words, normalization is another kernel method to map the data into an advantageous plane which is facility in calculation. Because the number of data is very large, complicated normalization algorithm will bring on a great deal of processing time. Fast and effective method is preferred.

Some normalization has been put forward, such as Zero-Mean normalization, Sigmoidal normalization Soft max normalization, Decimal Scaling, Max Normalization and Min-Max Normalization [8], [9], [11].

3.1 Zero-Mean normalization

Zero-Mean normalization method is based on the mean and standard deviation. Standard deviation is calculated by following:

$$x_{xtd} = \left[\frac{1}{N-1} \sum_{I=1}^{N} \left(x_i - x_{mean} \right)^2 \right]^{\frac{1}{2}} \quad (9)$$

x_{mean} is mean of data. The normalization equation is

$$x_i' = \frac{x_i - x_{mean}}{x_{std}} \quad (10)$$

This method works well in cases when we do not know the actual minimum and maximum of our input data or when we have outliers that have great effect on the range of the data.

3.2 Sigmoidal normalization

It transforms the input data nonlinearly into the range -1 to 1, using a sigmoid function.

$$x_i' = \frac{1 - e^{-a}}{1 + e^{-a}} \quad (11)$$

Where

$$a = \frac{x_i - x_{mean}}{x_{std}}$$

Data points within a standard deviation of the mean are mapped to the almost linear region of the sigmoid. Out-lier points are compressed along the tails of the sigmoidal function. Sigmoidal normalization is especially appropriate when you have outlier data points that you wish to include in the data set. It prevents the most commonly occurring values from being compressed into essentially the same values without losing the ability to represent very large outlier values.

3.3 Softmax Normalization

It is so called because it reaches "softly" toward its maximum and minimum value, never quite getting there.

$$x_i' = \frac{1}{1 + e^{-a}} \quad (12)$$

where a is the same as (11).The transformation is more or less linear in the middle range, and has a smooth non-linearity at both ends. The whole output range covered is 0 to 1 and the transformation assures that no present value lies outside this range.

3.4 Decimal Scaling

It normalizes by moving the decimal point of values. The number of decimal points moved depends on the maximum absolute value. This type of scaling transforms the data into a range between [-1, 1]. The transformation formula is

$$x_i' = \frac{x_i}{10^j} \tag{13}$$

where j is the smallest integer such that $MAX(|x_i'|) < 1$. This Scaling is useful when attributes values are greater than 1 in absolute value.

3.5 Max Normalization

In Max Value method, the normalization equation is

$$x_i = \frac{x_i}{x_{\max}}, x_{\max} = \max_{1 \le i \le N} x_i \tag{14}$$

When the maximal positive value is too small and the minimum negative value is too small, such as the value range (0.1, -10), it is possible to increase the difference and the normalized value range may be larger.

3.6 Min-Max Normalization

Min-Max Normalization performs a linear transformation on the original data *x* into the specified interval $(New_{\min}, New_{\max})$.

$$x_i = New_{\min} + (New_{\max} - New_{\min}) \times \left(\frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \right) \tag{15}$$

$$x_{\max} = \max_{1 \le i \le N} x_i, x_{\min} = \min_{1 \le i \le N} x_i$$

This method scales the data from $(x_{\min}, x_{\max})$ to $(New_{\min}, New_{\max})$ in proportion. The advantage of this method is that it preserves all relationships of the data values exactly. It does not introduce any potential bias into the data.

In these normalization methods, Zero-Mean normalization, Sigmoidal normalization and Softmax Normalization have to calculate the mean and standard deviation which will consume much time in large data. And the overflow problem will be worse. Decimal Scaling and Max Normalization is alike. In-Max Normalization has simple rules and adjustable range. From [11], Max method has better performance than Zero-Mean. So we used Min-Max Normalization and Max Normalization to compare in the following experiments.

4. Experiments

The data used in experiments originated from MIT's Lincoln Lab. It was developed for a KDD competition by DARPA and is considered a standard benchmark for intrusion detection evaluations. The approach is to train support vector machines to learn the normal behavior and attack patterns; then deviations from normal behavior are flagged as attacks.

The datasets contain a total of 22 training attack types, with an additional 14 types in the test data only.

Attacks fall into four main categories: DOS, R2L, U2R and Probing. It also has 41 continue and discrete features [3], [4], [12].

In the experiments, we composed first 10000 in the intrusion detection data. We partition the data into two classes: normal and attack, where the attack is the collection of all 22 different attacks belonging to four classes. The objective of our SVM experiments is to separate normal and attack patterns. In our case all attacks are classified as 0 and normal data classified as 1. In all experiments described the freeware package LibSVM is used [10]. Training is done using the RBF kernel option.

We use SVM to class the data and compare the Min-Max Normalization (MM-Norm) method with non-normalization (None-Norm) and Max Normalization (Max-Norm). 5-fold cross validation accuracy and calculation time are two important performances to prove the performance of the method. One computer with Intel Pentium M 1.7G CPU and 1G DDR RAM is used.

The results are shown in the Table1 and Table2. The experiment of table1 is used with 5-fold cross validation. The parameters of SVM in table1 are $C = 1$ and $\sigma = 1$. The experiment of table2 is implemented without cross validation.

The target is to train a classifier and compare their performance of forecast. The parameters of Table2 are $C = 3$ and $\sigma = 1$.

Table 1 5-fold cross validation test of SVM training used different normalization methods

5-fold cross validation		**None-Norm**	**Max-Norm**	**MM-Norm**
Time(s)	1	136	5	5
	2	148	6	6
	3	176	7	6
	4	152	5	5
	5	158	6	6
Iterative	1	10451	52	47
	2	9943	68	65
	3	11328	79	71
	4	9957	57	57
	5	10535	69	65
SV(BSV)	1	6187(53)	80(74)	45(37)
	2	6187(54)	81(75)	50(38)
	3	6204(59)	89(81)	56(41)
	4	6180(51)	79(72)	49(38)
	5	6197(53)	80(74)	50(37)
Accuracy		99.4%	99.83%	99.93%

From Table1, the calculation time of every fold in None-Norm is much more than Max-Norm and MM-Norm. The reason is the times of iterative are about ten thousand. The iterative of Max-Norm and MM-Norm with normalization is not more than one hundred. An interesting instance is, though the None-Norm has many SVs, it has fewer BSVs than Max-Norm. The accuracy of cross validation in MM-Norm is the best which means the MM-Norm is good at classing the training data. MM-Norm has least SVs which will save much time in classing the testing data.

Table 2 Results with parameter optimized

	None-Norm	Max-Norm	MM-Norm
Time(s)	165	6	5
Iterative	12918	72	57
SV(BSV)	7697(0)	62(55)	35(22)

From the comparison of calculation time in Table2, the times of iterative and quantity of SVs, MM-Norm has excellent performance. Max-Norm has close performance to MM-Norm in calculation time and the times of iterative. SVM without normalization will waste much time in calculation and get too many SVs. It will consume computation in function (4) to estimating the new data.

5. Conclusion

This paper researches the normalization in intrusion detection data which SVM are used to classification. Through analyze and experiments, the Min-Max normalization method have better performance than max-normalization method. It also indicates that normalization is a very important processing which can save much time of calculation and get a good performance classifier.

References

[1]G. V.N.Vapnik,"The Nature of Statistical Learning Theory," New York: Springer-Verlag, 1995.

[2]C.J.C.Burges."A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*,Vol.2, No.2,pp.1-47,1998.

[3]S.Mukkamala,G.Janoski,A H Sung, "Intrusion detection using neural networks and support vector machines," *Proceedings of IEEE International Joint Conference on Neural Networks(IJCNN 02)*, IEEE Computer Society Press, pp.1702-170,2002..

[4]A.H.Sung,S.Mukkamala, "Identify important features for intrusion detection using support vector machines and neural networks". *IEEE Proceedings of the 2003 Symposium on Application and the Internet*, IEEE Press, pp209-217,2003.

[5]K.Kobayashi,F.Komaki, "Information criteria for support vector machines," *Neural Networks, IEEE Transactions on,*Volume 17 ,pp.571-577,May 2006.

[6]D.Anguita,S.Ridella,F.Rivieccio, "K-fold generalization capability assessment for support vector classi¯ers", *Proceedings of IEEE. International Joint Conference on Neural Networks,(IJCNN 05)*,Vol.2, pp.855-858,2005.

[7]E.F.Codd, "Further Normalization of the Database Relational Model".In *Database System*, R.Rustin ed., Prentice-Hall, pp.33-64,1972

[8]G.Guo,D.Neagu, "Similarity-based Classi¯er Combination for Decision Making," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC05)*,IEEE press,pp.176-181,2005.

[9]C.X Dong,"study of support vector machines and its application in intrusion detection system[D]",Xi`an,China,XiDian University,pp.83-86,2004

[10] R.E.Fan, P.H.Chen,C.J.Lin, "Working Set Selection Using the Second Order Information for Training SVM", *Journal of Machine Learning Research,2005 6*,pp.1889-1918,2005

[11] *http://www:informatics:indiana:edu=predrag=classes/*2005*springi*400 */lecturelecture notes* 4_1*:pdf*

[12] *http : //kdd:ics:uci:edu/databases/kddcup*99*/kddcup*99*:html*