Computer and Information Science; Vol. 8, No. 2; 2015
ISSN 1913-8989 E-ISSN 1913-8997
Published by Canadian Center of Science and Education

Using J48 Tree Partitioning for Scalable SVM in Spam Detection

Mohammad-Hossein Nadimi-Shahraki[1], Zahra S. Torabi[1] & Akbar Nabiollahi[1]

[1] Faculty of Computer Engineering, Najafabad branch, Islamic Azad University, Najafabad, Isfahan, Iran

Correspondence: Zahra. S Torabi, Faculty of Computer Engineering, Najafabad branch, Islamic Azad University, Najafabad, Isfahan, Iran. Tel: 98-314-229-2440. E-mail: Zahra.torabi@sco.iaun.ac.ir

Received: January 31, 2015 Accepted: February 9, 2015 Online Published: April 25, 2015

doi:10.5539/cis.v8n2p37 URL: http://dx.doi.org/10.5539/cis.v8n2p37

Abstract

Support Vector Machines (SVM) is a state-of-the-art, powerful algorithm in machine learning which has strong regularization attributes. Regularization points to the model generalization to the new data. Therefore, SVM can be very efficient for spam detection. Although the experimental results represent that the performance of SVM is usually more than other algorithms, but its efficiency is decreased when the number of feature of spam is increased. In this paper, a scalable SVM is proposed by using J48 tree for spam detection. In the proposed method, dataset is firstly partitioned by using J48 tree, then, features selection are applied in each partition in parallel. Consistently, selected features are used in the training phase of SVM. The propose method is evaluated conducted some benchmark datasets and the results are compared with other algorithms such as SVM and GA-SVM. The experimental results show that the proposed method is scalable when the number of features are increased and has higher accuracy compared to SVM and GA-SVM.

Keywords: spam detection, support vector machine (SVM), partitioning, J48 Tree

1. Introduction

Spammers are continuously pioneering new methods to bypass email anti-spam filter solutions forcing companies to invest in spam filtering that can keep up with the evolving methods. Up to now, various methods have been presented in order to fight and spam detection (Jindal & Liu, 2007). According to this reason there is need to use hybrid multi-layer architectures method for Spam detection problems. Most of the hybrid multi-layer architectures filters include a combination of methods, such as applying key words, rule based filters, black and white list and other data mining for detecting the spams that are more important (Cook et al., 2006, Nakulas et al., 2009).

In the last decade one of the most important issues to detect spam based on content is machine learning and data mining algorithm (Chiroma et al., 2014). SVM is supervised learning models that recognize patterns and analyze data in machine learning. In many studies SVM performance in terms of error rate, speed and accuracy are higher than other algorithms such as Naïve Bayesian, Ripper, Rocchio, and Boosting Decision trees and non-parametric classification such as Neural Networks, Nearest Neighbor (Drucker et al., 1999; Liu et al., 2002; FENG & ZHOU, 2013). On the other hand, when the size of dataset increases due to the complexity of computation, it encounter to the lack of time and memory space (Tsang et al. 2005; FENG & ZHOU, 2013). In this paper, we proposed a method to solve this problem.

Partitioning is one of the method is used in modern database like IBM DB2 (McInerney, 2006), Microsoft SQL Server 2012 (Microsoft, 2012) and Oracle Database 11g (Oracle.DB) to improve manageability, performance and availability when we have big data. Then, a simple and liner algorithm is necessary to partition the dataset into minimum number of sub-tree. For an approach to be effective in the real-world continuous attributes deal sensibly with missing values, numeric attributes and pruning to deal with for noisy data. J48 is one of best well-known and widely used learning algorithm include the features is mentioned.

In this paper, we are going to offer a good method to remove the problem of SVM and applying it for detecting spam. We propose an approach which decomposes a large dataset into smaller ones and trains SVM algorithm on each part of them. This method can decrease the whole training time because the complexity of training time on SVM algorithm is in the order of N^2, where n is the number of features in training dataset (Chang et al., 2010). If each part deals with k samples, then to solve the complexity of the difficulties is in the order of $[(N/k) \times k^2 = N.k]$, if N<K then the complexity is so smaller than N^2. On the other hand, by decomposing a large dataset into smaller

dataset, it is so useful to select feature on each part and finally run SVM algorithm instead of decreasing the number of support vectors in each of the final result of SVM algorithm. We evaluate our model with other algorithms like SVM, GA-SVM.

This paper is organized as follows: section 2 is about related work. In Section 3 which is the main partition of this paper, we present our proposed method. Section 4 is about test results and finally, we conclude in section 5 and is about future work.

2. Related Work

The weakness of SVM algorithm is that the computational complexity that is not fitted for extensive dataset. Weight ratio is not constant; also it needs to decide to choose a good kernel functions and select a good value for parameter C. SVM is so suitable for the issues have limited training data features (Auria & Moro, 2008). In researches of (Chapelle et al., 1999; Kim et al., 2002), SVM is much more better and efficient than other non-parametric classifications, for example K-NN (Note 1), NN (Note 2) with the best classification's accuracy, have less computational time and set the parameters suitable. The result in the research of (Ye et al., 2008) represented that SVM occupied a high range of time and memory for big size of the data. One way for decrease and manage the size of dataset and database is feature selection. Another way to control with this problem is partitioning (Kerdprasop & Kerdprasop, 2003). Partitioning the data set to a proper subset is so beneficial the incremental mining to get its high accuracy. An improved approach for C4.5 had offered (Polat & Güneş, 2009). Experiment results run on three data set namely Lymphography, Image-segmentation, Dermatology from UCI. In the experiments a good accuracy in compare other algorithms was found but they didn't mention about performance and time against other type of datasets. In the research of (Chang, Guo et al., 2010) a decision tree to decompose dataset and train SVM is proposed. The method shows that the decision tree has several abilities for large-scale in training of SVM. First, it can classify the dataset with data points and reducing the cost of SVM training for data points. Second, it is so good to raise the accuracy. Third, the tree decomposition approach can decrease error rate. For data sets whose size can be control by current non-linear, or kernel-based, SVM training techniques, the proposed approach in (Chang, Guo et al., 2010) can speed up the training time and give good test accuracy. A number of researches have tried to hybrid SVM and decision trees. Some of them improved the accuracy of classification (Bennett & Blue 1998; Bennett, Cristianini et al., 2000; Tibshirani & Hastie, 2007). Some of the other researches speed up the SVM accuracy (Platt et al., 1999; Sahbi & Geman 2006; Gong et al., 2007).

In our previous work (Zahra S. Torabi, Mohammad et al., 2015), we have considered the evaluation criterions of SVM for spam detection and filtering.

In this article we use J48 tree to partition the original large data set into a data subsets that are manageable and learning effective, then on each partition we apply feature selection and use the selected features in training phase of SVM.

3. Methodology

Partitioning the given dataset to a suitable subset size is completely beneficial the incremental mining to get the high accuracy. The last research version of C4.5, implemented in Weka as J4.8 with Java. Indeed, J48 produces a decision tree for input dataset with recursive partitioning of dataset by Depth-first strategy. If classes C is denoted to $\{C_1, C_2,\ldots, C_k\}$. T has samples which depend on a mixture of classes. In this case, the idea is to purify T into proper subsets of samples which are heading to a single-class set of samples. A proper test is selected, based on single feature, which has one or more mutually exclusive results $\{O_1,O_2, \ldots,O_n\}$.T is divided into subsets $T_1, T_2, \ldots, T_n$ in which T_i has all the samples in T that have result O_i of the selected test. Entropy is a criterion of average uncertainty or ambiguity of collection of data. This represents the average much information we want to receive from result of a data source (Mazid, Ali et al. 2010). If *S* is a collection of samples, then *freq* (C_i, *S*) points to the set of samples in S that is in class C_i and also $|S|$ stand for the set of samples in the collection *S*. Equation1 shows the entropy of the set *S*:

$$Info(S) = -\sum \left(\left(\frac{freq(Ci,S)}{|S|}\right) \times log2 \left(\frac{freq(Ci,S)}{|S|}\right)\right) \quad (1)$$

When collection *T* has been divided in accordance with n results of one feature test *X*:

$$\text{Info}_x(\text{T}) = \sum \left(\left(\frac{|\text{Ti}|}{|\text{T}|}\right) \times \text{Info}(\text{Ti})\right) \quad (2)$$

Gain information creates with the split is really the difference between the amount of information require to

classify a situation after and before making the split. Equation3 shows the new gain rate:

$$Gain(X) = F.(\ Info(T) - \ Info_{x}(T)) \tag{3}$$

J48 adds a multiplier *F* in front of the gain calculation. F equals number of samples in dataset with known value for a given features divide total number of samples to control missing values. The gain uses alone is not sufficient to make a tree. The gain measure proper splits with many results. Equation 4 defines gain ratio as follow to solve this problem:

$$\text{GainRatio (X)} = \frac{\text{Gain(X)}}{\text{SplitInfo(X)}} \tag{4}$$

The gain ratio divides the gain with the evaluated split information. This penalizes splits with many results in equation 5.

$$SplitInfo(X) = -\sum_{i=1}^{n+1} \frac{|Si|}{|S|}.log2\frac{|Si|}{|S|} \tag{5}$$

Split information is the weighted average of the information utilizing the ratio of states that are sent to each child. The best evaluation function can measure the excellence of a subset that produces from generation function and compared with the previous subset. A certain evaluation function is requred to detect the best subset. If U is an *inconsistency* measure *rate* for an input dataset and a *pattern* is a part of a sample without definition of label for class. S is a feature subset with $n_{f\,1}$, $n_{f2},\ldots,n_{f|S|}$ number of values for features $f_1,f_2,\ldots f_{|s|}$ patterns. Consistency measure is related to the inconsistency rate concept (Lin, Lee et al. 2008). For each discrete feature, one test with results is considered as many as the number of distinct values of the feature. On the other hand, for each continuous feature, binary tests are required for every distinct values of the feature (Zhao and Zhang 2008). Figure 1 shows the propose partition algorithm:

Algorithm Partitioning(root, **S**)
input: One root node, Set **S** of training samples
output: A partition tree, attached to root node
Consider each samples
if **S** is pure **then** Assign label class of pure sample to root;
return;
end
if no split can output minimum number of samples split off
then
Allocate label class of the node as the main class of **S**
return;
end
Detect the best split which divides **S** into subsets
For each feature *a*

- Detect the normalized information gain ratio from splitting on *a*
- Let *a_best* be the feature with the maximum normalized information gain
- Create a decision *node* that splits on *a_best*
- Select a best feature base on Consistency measure
- Make sublists get by splitting on *a_best*, and also add those nodes as child of *node* ->Ssub
- Partitioning(*a_best*, Ssub)

End

Figure 1. The propose partition **algorithm**

The above pseudocode in figure1 shows the steps of partitioning where *S* represents the set of training states. In order to get the optimal split while the tree is growing (see the section of the pseudo-code above) the gain ratio must be calculated. We find the best split with Consistency measure. Consistency measures is type of evaluation measures are characteristically different from other measures because of their heavy reliance on the training dataset and use of Min-Features bias in choosing a subset of features. Consistency measure is employed in (Almuallim and Dietterich 1994). When partition dataset is finished, we apply GA algorithm on each partition for feature selection simultaneously and then we use selected features in training phase of SVM.

4. Results and Discussion

To validate our method, we conducted a thorough experimental evaluation over email dataset. We used the Accuracy and 10 folds cross validation. The results are shown on 4 datasets Spambase(Arthur Asuncion 2007) and (SpamCorpus, SpamData, mail_corpus)(Katakis 2008).We compare our method on SVM, GA-SVM. The algorithms are implemented with Java source code in java Net Beans. We add some dlls and jar files from WEKA to our program. All experiments have been run on a machine with Core i5 CPU and 4G MB of RAM. We used the accuracy rate and error rate with 10 folds cross validation to evaluate the method. Accuracy and error rate formulas calculated with equation 6 and 7:

$$\text{Accuracy} = (n_{L\to L} + n_{S\to S})/(n_{L\to L} + n_{L\to S} + n_{S\to L} + n_{S\to S}) \quad (6)$$

$$\text{Error rate} = (n_{L\to S} + n_{S\to L})/(n_{L\to L} + n_{L\to S} + n_{S\to L} + n_{S\to S}) \quad (7)$$

In these formulas nL→ S and nS → L denote legitimate emails and spam emails that have not been classified. nL→ L and nS → S represent the number of legal emails and spams are correctly classified.

Table 1 Shows the datasets that use in experiment:

Table 1. Dataset in use

Dataset	Num Instances	Num Features
Spambase	4601	58
SpamData	9324	500
mail_corpus	4942	27894
SpamCorpus	1842	39917

Table 2 shows the comparison time between the proposed method and the other algorithms:

Table 2. Compare the time of proposed method with SVM and GA-SVM

Dataset	Num Features	SVM	GA-SVM	Proposed method
Spambase	58	4080.35 s	64.05 s	103.74 s
SpamData	500	10800.76 s	793.36 s	575.41 s
mail_corpus	27894	-	18485.3 s	10321.1 s
SpamCorpus	39917	-	-	15321.08 s

The sign of "–" in tables represents that the algorithm couldn't run on that point. As it can be seen in table 2, the execution time increases for both SVM and GA-SVM algorithms as the size of the dataset increases. The performance of GA-SVM base on execution time is better than SVM but when size of the dataset increases both algorithms couldn't run and face to the lack of memory. On the other hand, by increasing the size of dataset and features, our proposed method has a less execution time than others and doesn't face to the lack of memory because of using partitioning the datasets with J48 and eliminate redundant and irrelevant features, and capable of handling some noise.

Table 3. Compare the Error Rate Between Proposed Method, SVM and GA-SVM

Dataset	Num Features	SVM	SVM-GA	Proposed Method
Spambase	58	0.2554	0.1108	0.0741
SpamData	500	0.066	0.0339	0.0323
mail_corpus	27894	-	0.0722	0.0569
SpamCorpus	39917	-	-	0.0223

As it can be seen in table 3 by increasing the number of attributes, the error rate of proposed approach reduced and this represents an increase in the precision of this method. Also, the error rate of SVM is lower than SVM-GA. Compared with other methods, the error rate of proposed method is less than the other methods because in the partitioning phase the noise and missing values are ignored with J48 tree.

Table 4 shows the accuracy on datasets and compares the algorithms with proposed method:

Table 4. The comparison of accuracy with GA-SVM, SVM and Proposed method

Dataset	Num Features	SVM	GA-SVM	Proposed method
Spambase	58	72.32%	84.91%	91.04%
SpamData	500	91.35%	93.21%	95.24%
mail_corpus	27894	-	92.8%	94.3%
SpamCorpus	39917	-	-	96.32%

Table 4 represents the accuracy of proposed method, SVM and GA-SVM. As the results show by increasing the number of feature, the accuracy of GA-SVM is better than SVM due to feature selection with GA and the accuracy of our proposed method is higher than both algorithms.

5. Conclusion and Feuture Work

Today, Spam has become a problem for users of Internet, IT companies and organizations. In some studies, SVM's performance is more than other categories, but in the computational complexity of high-dimensional data collection, its performance decreases (Tsang, Kwok et al. 2005, FENG and ZHOU 2013) because of the complex mathematical calculations in the kernel function, SVM faces to the lack of memory and run time. So, SVM is suitabale and has the best performance from the other classifications when the number of features of the data set is small. In this paper, a scalable SVM is proposed by using J48 tree for spam detection. In the proposed method, dataset is firstly partitioned by using J48 tree, then, features selection are applied in each partition in parallel. Consistently, selected features are used in the training phase of SVM. We show that by increasing features, the accuracy of propose method has been increased because, when size of dataset become large, we face to lack of memory and couldn't have calculated time and accuracy that's the things occur in other algorithms like SVM and GA-SVM. Therefore the performance of proposed algorithim is better than SVM and GA-SVM. For future work the other trees or other effective features selection can use instead of J48 algorithm.

References

Almuallim, H., & Dietterich, T. G. (1994). Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence, 69*(1), 279-305. http://dx.doi.org/10.1016/0004-3702(94)90084-1

Arthur Asuncion, D. N. (2007). Retrieved from http://archive.ics.uci.edu/ml/datasets/Spambase

Auria, L., & Moro, R. A. (2008). Support vector machines (SVM) as a technique for solvency analysis, Discussion papers. German Institute for Economic Research. http://dx.doi.org/10.2139/ssrn.1424949

Bennett, K. P., & Blue, J. A. (1998). A support vector machine approach to decision trees. Neural Networks Proceedings (1998). IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on, IEEE. http://dx.doi.org/10.1109/IJCNN.1998.687237

Bennett, K. P. et al. (2000). Enlarging the margins in perceptron decision trees. *Machine Learning, 41*(3), 295-313. http://dx.doi.org/10.1023/A:1007600130808

Chang, F. et al. (2010). Tree decomposition for large-scale SVM problems. *The Journal of Machine Learning Research, 11*, 2935-2972.

Chapelle, O. et al. (1999). Support vector machines for histogram-based image classification. *Neural Networks, IEEE Transactions on, 10*(5), 1055-1064. http:dx.doi.org/ 10.1109/72.788646

Chiroma, H. et al. (2014). A Framework for Selecting the Optimal Technique Suitable for Application in a Data Mining Task. Future Information Technology. *Springer,* 163-169. http://dx.doi.org/10.1007/978-3-642-40861-8_25

Cook, D. et al. (2006). Catching spam before it arrives: domain specific dynamic blacklists. Proceedings of the 2006 Australasian workshops on Grid computing and e-research, 54, Australian Computer Society, Inc.

Drucker, H. et al. (1999). Support vector machines for spam categorization. *Neural Networks, IEEE Transactions on, 10*(5), 1048-1054. http://dx.doi.org/10.1109/72.788645

Feng, Y., & Zhou, H. (2013). An Effective and Efficient Two-stage Dimensionality Reduction Algorithm for Content-based Spam Filtering⋆. *Journal of Computational Information Systems, 9*(4), 1407-1420.

Jian-qing, S. et al. (2007). A novel SVM decision tree and its application to face detection. Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on, IEEE. http://dx.doi.org/10.1109/SNPD.2007.72

Jindal, N., & Liu, B. (2007). Review spam detection. Proceedings of the 16th international conference on World

Wide Web, ACM. http://dx.doi.org/10.1145/1242572.1242759

Katakis, I. (2008). Retrieved from http://mlkd.csd.auth.gr/concept_drift.html

Kerdprasop, N., & K. Kerdprasop (2003). Data partitioning for incremental data mining. The 1st International Forum on Information and Computer Science, Citeseer.

Kim, K. I. et al. (2002). Support vector machines for texture classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24*(11), 1542-1550. http://dx.doi.org/10.1109/TPAMI.2002.1046177

Lin, S. W. et al. (2008). Parameter determination of support vector machine and feature selection using simulated annealing approach. *Applied soft computing, 8*(4), 1505-1512. http://dx.doi.org/10.1016/j.asoc.2007.10.012

Liu, H. et al. (2002). A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. Genome Informatics Series, 51-60. http://dx.doi.org/10.11234/gi1990.13.51

Mazid, M. M. et al. (2010). Improved C4. 5 algorithm for rule based classification. Proceedings of the 9th WSEAS international conference on Artificial intelligence, knowledge engineering and data bases, World Scientific and Engineering Academy and Society (WSEAS).

McInerney, P. (2006). DB2 Partitioning Features. Retrieved from http://www.ibm.com/developerworks/data/library/techarticle/dm-0608mcinerney

Microsoft. (2012). SQL Server. Retrieved from http://technet.microsoft.com/enus/sqlserver/ff898410

Nakulas, A. et al. (2009). A review of techniques to counter spam and spit. Proceedings of the European Computing Conference. *Springer*. http://dx.doi.org/10.1007/978-0-387-84814-3_50

Oracle. DB. Oracle Database 11g. Retrieved from http://www.oracle.com/technetwork/database/options/partitioning/index.html

Platt, J. C. et al. (1999). Large Margin DAGs for Multiclass Classification. nips.

Polat, K., & Güneş, S. (2009). A novel hybrid intelligent method based on C4. 5 decision tree classifier and one-against-all approach for multi-class classification problems. *Expert Systems with Applications, 36*(2), 1587-1592. http://dx.doi.org/10.1016/j.eswa.2007.11.051

Sahbi, H., & Geman, D. (2006). A hierarchy of support vector machines for pattern detection. *The Journal of Machine Learning Research, 7*, 2087-2123.

Tibshirani, R., & Hastie, T. (2007). Margin trees for high-dimensional classification. *The Journal of Machine Learning Research, 8*, 637-652.

Tsang, I. W. et al. (2005). Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research.*

Ye, M. et al. (2008). The Spam Filtering Technology Based on SVM and DS Theory. Knowledge Discovery and Data Mining, 2008. WKDD 2008. First International Workshop on, IEEE. http://dx.doi.org/10.1109/WKDD.2008.42

Zahra, S. T. et al. (2015). Efficient Support Vector Machines for Spam Detection: A Survey. *International Journal of Computer Science and Information Security (IJCSIS), 13*(1), 1947- 5500.

Zhao, Y., & Zhang, Y. (2008). Comparison of decision tree methods for finding active objects. *Advances in Space Research, 41*(12), 1955-1959. http://dx.doi.org/10.1016/j.asr.2007.07.020

Notes

Note 1. K-nearest neighbor.

Note 2. Neural Networks.