EMAIL SPAM FILTERING USING R-CHUNK DETECTOR-BASED NEGATIVE SELECTION ALGORITHM

Vu Duc Quang[1*], Vu Manh Xuan[1],
Nguyen Van Truong[1], Phung Thi Thu Trang[2]
[1]College of Education–TNU, [2]Foreign Language Faculty- TNU

SUMMARY

Email spam is one of the biggest challenges when using the Internet today. It causes a lot of troubles to users and does indirect damages to the economy. Machine learning is a keyapproach for spam filtering. Artificial Immune System (AIS) is a diverse research area that combines the disciplines of immunology and computation.Negative selection mechanism is one of the most studied models of biology immune system for anomaly detection. In this paper, Negative Selection Algorithms (NSA), a computational imitation of negative selection, ismodeledfor spam filtering. The experimental results on popular TREC'07 spam corpus show that our approach is an effective solution to the problem on both time complexities and classification performance.

Keywords: *Artificial immune system, negative selection algorithm, spam filtering, r-chunk detector*

INTRODUCTION

Email is one of the most popular means of communication nowadays. There are billions of emails sent every day in the world, half of which are spams. Spams are unexpected emails for most users that aresent in bulk with main purpose of advertising, stealing information, spreading viruses.For example, Trojan.Win32.Yakes.fize is the most malicious attachment Trojan that downloads a malicious file on the victim computer, runs it, steals the user's personal information and forwards it to the fraudsters.

There are a lot of spam filtering methods such as Blacklisting, Whitelisting, Heuristic filtering, Challenge/Response Filter, Throttling, Address obfuscation, Collaborative filtering. However, most of anti-spam filters base on the headers of letters or the sending address to increase the speed. One uses complicated techniques to improve accuracy affects the speed of the whole system as well as the psychology of users. Recently, machine learning approaches have been paid more attention because they are highly adaptable to the spam digestion, such as Naïve Bayes, Support Vector Machine, K-Nearest Neighborsand Artificial Neuron Network.

AIS inspired by lymphocyte repertoires includes negative and positive selection, clonal selection, and B cell algorithms. Among various mechanisms in the immune system that are explored for AIS, negative selection is one of the most studied models. NSA is a computational imitation of self-nonself discrimination, it is first designed as a change detection method. Since its introduction in 1994, NSA has been a source of inspiration for many computing applications, especially for intrusion detection, computer virus detection and monitoring UNIX processes [8].

The outline of a typical NSA contains two stages [1]. In the generation (or training) stage (Fig. 1), the detectors are generated by some random processes and censored by trying to match given self samples taken from set *S*. Those candidates that match are eliminated and the rest are kept as detectors in set *D*. In the detection (or testing, classifying) stage, the collection of detectors (or detectors set) is used to verify whether an incoming data instance is self or nonself. If it matches any detector, it is claimed as nonself or anomaly, otherwise it is self.

[*] *Tel: 01652 340851; Email: vdquang1991@gmail.com*

The r-chunk and r-contiguous detectors are considered the most common ones in the AIS literature. The r-contiguous detectors are originally researched by many authors, and r-chunk detectors were later introduced to achieve better results on data where adjacent regions of the input strings are not necessarily and semantically correlated, such as network data packets. In this article, we only apply NSA under r-chunk detectors to solve the problem of spam filtering.

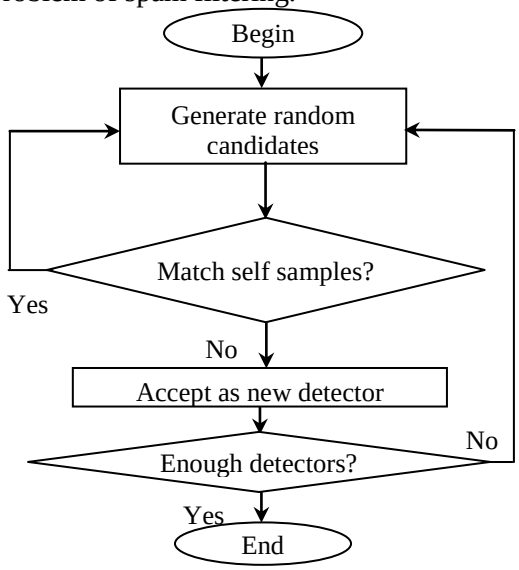

Figure 1. *Model of negative detector generation*

All existing NSAs for spam filtering use modified version of the classical one with real-valued vector representation for data and detectors. They are always combined with text mining algorithms. Our contribution is to apply an r-chunk detector-based NSAthat uses binary string representation to increase effectiveness of the detection process and reduce the runtime significantly.

The remaining of the paper is organized as follows: In the next section, we define r-chunk detectors. The subsequent section, the main part of the paper, shows the r-chunk detector-based NSA for spam filtering. In the last section, we summarize our approach and discuss future works.

BINARY CHUNK-BASED DETECTORS

In this paper, we consider NSA as a classifier operating on a binary string space Σ^{ℓ}, where $\Sigma = \{0, 1\}$. We also use the following notations: Let $s \in \Sigma^{\ell}$ be a binary string. Then $\ell = |s|$ is the length of s and s[i,…, j] is the substring of s with length j – i + 1 that starts at position i. In the following section, we will show how to convert anarbitrary string to binary one.

Definition 1 (Chunk detectors). An r-chunk detector (d, i) is a tuple of a string $d \in \Sigma^{r}$ and an integer $i \in \{1,\ldots, \ell - r + 1\}$. It matches another string $s \in \Sigma^{\ell}$ if s[i,…, i + r - 1] = d.

Example 1. Given a self set S having 6 binary strings, with $\ell = 5$ and r = 3: S = {s_1 = 00000; s_2 = 00010; s_3 = 10110; s_4 = 10111; s_5 = 11000; s_6 = 11010}, all 3-chunk detectors that do not match any string in S are listed as following:D = {(001,1); (010,1); (011,1); (100,1); (111,1); (010,2); (110,2); (111,2); (001,3); (011,3); (100,3); (101,3)}.

Each 3-chunk detector can detect a sub-set of nonself strings. For example, detector (111,1) can classify four strings 11100, 11101, 11110, 11111 as nonself strings or spams because they all match string 111 at their first position.

Using chunk detectors may reduce number of undetectable strings, or holes, in comparison to r-contiguous detectors based approaches [8].

NEGATIVE SELECTION ALGORITHM FOR SPAM FILTERING

A two-dimensionalarrayused as a main data structure in our studyis just for easy understanding ouralgorithm. The readers can refer to [4, 7] for more effective r-chunk detectors generation on treesor automata. The algorithm is divided into two phases:training phase to generate detectors and testing one to check whether a given string is ham (self) or spam (nonself)as follows.

The training process

Input: A self set S of the binary strings converted from hams with the same length of ℓ; an integer r, $1 < r < \ell$.

Output: Set of r-chunk detectors D.

Firstly,a temporary array A with the size of $2^{r} \times (\ell\text{-}r+1)$ is used as a hash table of S. Then detectors are created from the above array.

*Procedure*Chunk_Generation;

Begin

Create array Ahavingall elements are assigned to 0;

Foreach s in S do

For j:=1 to ℓ-r+1 do

Begin

i := the integer number of binary sub-string of s whose length is r and starting position is j within the string s;

A[i, j] := 1;

End;

D = ∅;

For i:=0 to 2^r do

For j:=1 to ℓ-r+1 do

If A[i,j]=0 then D := D ∪ (i_2, j);

End;

For example, with s_3 = 10110 as in Example 1, three elements A[5, 1], A[3, 2] and A[6, 3] are assigned to 1. These then create three 3-chunk detectors (101, 1), (011, 2) and (110, 3).

The testing process.

Input: Set of detectors D, a string s, and two integer ℓ, r.

Output: Detection of s if it is spam or ham.

This process is easier than the first one.A Boolean variable check_spamis used to check if the given string s is spam or not.

*Procedure*Chunk_Detection;

Begin

check_spam:=false;

For j:=1 to ℓ-r+1 do

Begin

i := sub-string of s whose length is r and starting position is j within the string s;

If (i, j) in D then

Begin

check_spam:=true;

Break;

End;

End;

Ifcheck_spamthen “s is spam” else “s is ham”;

End;

The time complexities of the training process and testing process are O(|S|.(ℓ-r+1)) and (ℓ-r+1), respectively.

EXPERIMENT

In this section, theexperiment on theTREC’07 spam corpus [6] is implemented and its results are compared with those of most recentones [3].

TREC’07 spam corpus stored 75.419 emails including 50.199 spams and 25.220 hams. That is one of the largest and most reputable data co-sponsored by the National Institute of Standards and Technology (NIST) and U.S. Department of Defense.This Spam Corpus is suitable for our research because of two factors: Firstly, it is publicly available, making it possible for new and old researchers to verify the results or test against the same corpus. Secondly, the spam corpus is gathered from multiple email addresses that provide better experimental results than when it is collected from a single address.

Before performing binary-based NSA, we remove the structure information of emails, i.e. the header tags, to retain only the text content, as seen in Fig. 2.

OEM software at greatest bargains!

Ms Office 2007, Windows Vista, Photoshop all are below $50. Why waiting??

http://www.justsoftwares.info

Figure 2. *Typical text content of a spam email from TREC’07 spam corpus*

Then each email content is processed by removing all punctuation marks and spaces,then converted (each character’s ASCII code) into the binary form. Naturally, hams and spams are considered as self and

nonself, respectively. Therefore, only binary strings that represent hams are used for the training phase.

In 75.419 emails, we choose 5000 hams and 5000 spams randomly, then used 5000 hams onlyfor training by Chunk_Generation algorithm.

We used the common performance measurements: TP (True positive: the number of spam emails classified correctly), TN (True negative: the number of ham emails classified correctly), FP (False positive: the number of ham inaccurately classified as spam) and FN (False negative: The number of spam wrongly classified as ham).

Other measurementslike Detection Rate (DR), False positive rate (FPR) and Overall accuracy (Acc) are listed as follows:

$$DR = TP/(TP + FN)$$

$$FPR = FP/(TN + FP)$$

$$Acc = (TP + TN) /(TP + TN + FP + FN)$$

Table 1. *Nine-fold experiment on TREC'07*

HAM	SPAM	TP	FP	FN	TN	DR	FPR	Acc
100	900	894	0	6	100	99.33	0	99.40
200	800	793	0	7	200	99.13	0	99.30
300	700	695	0	5	300	99.29	0	99.50
400	600	596	0	4	400	99.33	0	99.60
500	500	496	0	4	500	99.20	0	99.60
600	400	399	0	1	600	99.75	0	99.90
700	300	297	0	3	700	99	0	99.70
800	200	200	0	0	800	100	0	100
900	100	100	0	0	900	100	0	100
Average						99.45	0	99.67

We used 9 test cases: each test contains 1000 emails taken randomly from the original set 10000 emails and change corresponding percentage between the number of hams and spams as used in [3]. Two arguments ℓ, r are assigned to 55 and 17, respectively. These optimal arguments are chosen after several runs of the algorithm. The results are showed in Table 1.

The experimental results shows a remarkable performance with overall 99.67% accuracy. This results support our approach to the spam filter using NSA under r-chunk detectors with binary representation.

In [3], the average performance measurements DR, FPR and Acc when usingNSA are 51.5%, 0%, 76.44%, and when using a combination of Naïve Bayesand Clone Selection and NSA are 98.09%, 0%, 98.82%, respectively. These results are much lower in comparison with our ones, the corresponding measurementsshowed in the Table 1, 99.45%, 0%, 99.67%.

The binary representation proposed in our approach is main factor that lead to the good results.The optimal argument ℓ, r also play an important role in the algorithm. Moreover, in terms of execution time, the their program runs 9:31s on average, while our program to train only takes 50s only (we use Visual C# 2013 as IDE on Windows 8.1 Pro, Chip Core i5, 3210M, 2.5Ghz, RAM DDR3 2GB).

CONCLUSIONS

In this paper we performed content-based spam filtering using NSA. The standard benchmark spam corpusTREC'07 is used for experiment with9-fold cross experiment technique.The results show a much better classification performance than most recent results in [3]. We predict that better results would be obtain if more techniques are used in data preprocess, such as removing all stop words, compressing data, and removing words that appear in both hams and spams. This expansion will be presented in detailed in our next article.

In future works, we seek to extend the model to other data representations and apply itto awide range of spam types, such as Blog spam, SMS spam and Web spam. Moreover, combining immune algorithms with classical statistical models maybe a very good idea for the problem.

REFERENCES

1. Forrest et al, 1994, Self-Nonself Discrimination in a Computer, in Proceedings of 1994 IEEE

Symposium on Research in Security and Privacy, Oakland, CA, 202-212.
2. Gordon Cormack, 2007, TREC 2007 Spam Track Overview, University of Waterloo, Waterloo, Ontario, Canada.
3. MarwaKhairy et al, 2014, An Efficient Three-phase Email Spam Filtering Technique, British Journal of Mathematics & Computer Science 4(9): 1184-1201.
4. Nguyen Van Truong, Vu Duc Quang, Trinh Van Ha, 2012, A fast r-chunk detector-based negative selection algorithm, Journal of Science and Technology, Thai Nguyen University, 2 (90), 55-58.
5. Terri Oda, 2004, A Spam-Detecting Artificial Immune System, Master thesis of Computer Science, Ottawa-Carleton Institute for Computer Science School of Computer Science Carleton University Ottawa, Canada.
6. T. Stibor et al., 2004, An investigation of r-chunk detector generation on higher alphabets, GECCO 2004, LNCS 3102, 299-307.
7. J. Textor, K. Dannenberg, and M. Liskiewicz, 2014, A generic finite automata based approach to implementing lymphocyte repertoire models. In Proceedings of the 2014 Conference on Genetic and Evolutionary Computation, GECCO'14, 129-136, USA.
8. Z. Ji and D. Dasgupta, 2007, Revisiting negative selection algorithms. Evol. Comput., 15(2):223-251.

TÓM TẮT

LỌC THƯ RÁC SỬ DỤNG THUẬT TOÁN CHỌN LỌC ÂM TÍNH DỰA TRÊN BỘ DÒ R-CHUNK

Vũ Đức Quang[1*], Vũ Mạnh Xuân[1], Nguyễn Văn Trường[1], Phùng Thị Thu Trang[2]
[1]Trường Đại học Sư phạm - ĐH Thái Nguyên,
[2]Khoa Ngoại ngữ - ĐH Thái Nguyên

Hiện nay, thư rác là một trong những vấn đề đáng lo ngại khi sử dụng Internet. Nó gây nhiều phiền toái cho người dùng và gián tiếp làm thiệt hại về kinh tế. Học máy là một cách tiếp cận chính cho lọc thư rác. Hệ miễn dịch nhân tạo là một lĩnh vực nghiên cứu phong phú kết hợp các nguyên lý miễn dịch học và tính toán. Cơ chế chọn lọc âm tính là một trong những mô hình được nghiên cứu nhiều nhất của hệ thống miễn dịch sinh học cho phát hiện bất thường. Trong bài báo này, thuật toán chọn lọc âm tính, một mô phỏng của chọn lọc âm tính trên máy tính, được mô hình cho bài toán lọc thư rác. Kết quả thực nghiệm với bộ dữ liệu thư rác TREC'07 cho thấy đây là một phương pháp hiệu quả để xử lí cho vấn đề này trên cả hai tiêu chí là độ phức tạp thời gian thực hiện và hiệu suất phân loại.

Từ khóa: *Hệ miễn dịch nhân tạo, thuật toán chọn lọc âm tính, lọc thư rác, bộ dò r-chunk*

Ngày nhận bài:25/9/2015; Ngày phản biện:10/10/2015; Ngày duyệt đăng: 31/5/2015
Phản biện khoa học: *PGS.TS Nguyễn Văn Tảo – Trường Đại học Công nghệ Thông tin & Truyền thông- ĐHTN*

[*] *Tel: 01652 340851; Email: vdquang1991@gmail.com*