Tolerant V-Detector algorithm

Andrzej Chmielewski
Faculty of Computer Science, Białystok University of Technology, Wiejska 45a Street, 15-351 Białystok, Poland
E-mail: a.chmielewski@pb.edu.pl

Abstract. V-Detector algorithm is an example of negative selection algorithm, proposed to distinguish between self and nonself samples, represented as real-valued vectors. It is used mainly for detecting anomalies in high dimensional datasets. For this purpose, the set of detectors is generated, usually in random way, based only on information about normal behaviour. The main problem with this algorithm is scalability and high computational complexity. This paper presents a new approach for building the detectors, inspired by ideas from the tolerance rough sets. With this algorithm it is possible to detect outliers as well as those samples which are very similar to both classes and their correct classification requires additional verification. Partially, it also solves the scalability problem as detectors has a higher discrimination power in comparing to its basic version.

1. Introduction

The Negative Selection Algorithm (NSA) is inspired by the process of thymocytes (i.e. young T-lymphocytes) maturation in Nature Immune System, where only those lymphocytes survive which do not recognize any self molecule (own cell) [1]. A nice feature of NIS is that it does not need any example of pathogen (called nonself) to detect them. It means, such systems are able to detect even new and never met before harmful activities.

Many different NSA have been proposed over the last two decades (e.g. [2, 3, 4]). They have found wide range of applications in areas such as e.g. computer security, spam filtering, virus detection, handwritten character recognition (for more details, see, e.g. [5]). However, in many domains, there was encountered uncertain data and discrimination between self and nonself not always was possible. Some samples were very similar, regardless of used affinity (distance) metrics. They are very dangerous, especially in the security systems. In the case of uncertainty, they should be passed to additional post processing, to perform additional tests which allow for correct classification.

One of effective approaches for handling uncertainty in data is rough set theory [6, 7]. Some rough set inspirations have already been applied to NSA to deal with this problem. In [8] a lymphocytes, which are responsible for seeking and destroying the pathogens in NIS, were implemented as a decision rules, necessary to specify both: self and nonself classes. The proposed solution was implemented with rough valued lymphocytes to emulate approximate binding performed by NIS. In [9] rough sets were used for reducing the number attributes in AIS. In this case, the tests were conducted on KDD Cup 1999 high-dimensional dataset to detect anomalies in network connection.

In this paper, it is proposed the tolerant V-Detector algorithm, also inspired by RST. Similar
approach was already presented in [10], where upper approximation, for both: binary and real-valued representations, were arbitrary selected base on the experimental results. Here, this value is dependent on number of the few nearest self samples.

2. Negative Selection Algorithm

The NSA, i.e. the negative selection algorithm, proposed by Forrest et al., [1], is inspired by the process of thymocytes (i.e. young T-lymphocytes) maturation: only those lymphocytes survive which do not recognize any self molecules.

Formally, let $\mathcal{U}$ be a universe, i.e. the set of all possible molecules. The subset $\mathcal{S}$ of $\mathcal{U}$ represents the collection of all self molecules and its complement $\mathcal{N}$ in $\mathcal{U}$ represents all nonself molecules. Let $\mathcal{D} \subset \mathcal{U}$ stand for a set of detectors and let $\text{match}(d,u)$ be a function (or a procedure) specifying if a detector $d \in \mathcal{D}$ recognizes the molecule $u \in \mathcal{U}$. Usually, $\text{match}(d,u)$ is modelled by a distance metric or a similarity measure, i.e. we say that $\text{match}(d,u) = \text{true}$ only if $\text{dist}(d,u) \leq \delta$, where $\text{dist}$ is a distance and $\delta$ is a pre-specified threshold. Various matching function are discussed e.g. in [11] and [12].

The problem relies upon construction the set $\mathcal{D}$ in such a way that

$$\text{match}(d,u) = \begin{cases} \text{false} & \text{if } u \in \mathcal{S} \\ \text{true} & \text{if } u \in \mathcal{N} \end{cases}$$

for any detector $d \in \mathcal{D}$.

A naive solution to this problem, implied by the biological mechanism of negative selection, consists of five steps:

(a) initialize $\mathcal{D}$ as empty set, $\mathcal{D} = \emptyset$,
(b) generate randomly a detector $d$,
(c) if $\text{match}(d,s) = \text{false}$ for all $s \in \mathcal{S}$, add $d$ to the set $\mathcal{D}$,
(d) repeat steps (b) and (c) until the sufficient number of detectors is be generated.

3. V-Detector algorithm

The original V-Detector algorithm was proposed by Ji and Dasgupta in [3]. It operates on normalized vectors of real-valued attributes. Each vector can be viewed as a point in the $d$-dimensional unit hypercube, $\mathcal{U} = [0,1]^d$. Each self sample, $s_i \in \mathcal{S}$, is represented as a hypersphere $s_i = (c_i,r_s), i = 1,\ldots,l$, where $l$ is the number of self samples, $c_i \in \mathcal{U}$ is the center of $s_i$ and $r_s$ is its radius. It is assumed that $r_s$ is identical for all $s_i$’s. Each point $u \in \mathcal{U}$ inside any self hypersphere $s_i$ is considered as a self element.

For Euclidean affinity metric, the detectors $d_j$ are represented as hyperspheres: $d_j = (c_j,r_j), j = 1,\ldots,p$, where $p$ is the number of detectors. In contrast to self elements, the radius $r_j$ is not fixed but it is computed as the Euclidean distance from a randomly chosen center $c_j$ to the nearest self element (this distance must be greater than $r_s$, otherwise the detector is not created). Formally, we define $r_j$ as

$$r_j = \min_{1 \leq i \leq l} \text{dist}(c_j,c_i) - r_s.$$  \hspace{1cm} (2)

In its original version, the V-Detector algorithm employs Euclidean distance to measure proximity between a pair of samples. Therefore, self samples and the detectors are hyperspheres. Formally, Euclidean distance is a special case of Minkowski norm $L_m$ defined as:

$$L_m(x,y) = \left( \sum_{i=1}^{d} |x_i - y_i|^m \right)^{\frac{1}{m}},$$  \hspace{1cm} (3)
where \( x = (x_1, x_2, \ldots, x_d) \) and \( y = (y_1, y_2, \ldots, y_d) \) are points in \( \mathbb{R}^d \).

Particularly, \( L_2 \)-norm is Euclidean distance, \( L_1 \)-norm is Manhattan distance and \( L_\infty \) is Tchebyshev distance. Additionally, it was defined fractional distance metric (with \( 0 < m < 1 \)), which is more appropriate for high-dimensional spaces.

![Image](image_url)

**Figure 1.** (a) Example of performance \( V \)-Detector algorithm for 2-dimensional problem. Black and grey circles denotes self samples and \( v \)-detectors, respectively. (b) Unit spheres for selected \( L_m \) norms in 2D.

Another consequence of applying fractional metrics for \( V \)-Detector algorithm is modification of the shape of detectors. Figure 1(b) presents the unit spheres for selected \( L_m \)-norms in 2D with \( m = 2 \) (outer most), \( 1, 0.7, 0.5, 0.3 \) (inner most).

4. Tolerant \( V \)-Detector

\( V \)-Detector algorithm is sensitive on self samples which are outliers. They obstruct the \( N \) space coverage and make a larger number of detectors necessary. Tolerant \( V \)-Detector introduces an additional radius of tolerance \( r_j^t \) (\( r_j^t \geq r_j \)), which is a distance to the nearest \( k \)-self sample, instead of the nearest one. Hence, each tolerant \( v \)-detector (\( v_k^t \)-detector) in \( \mathbb{R}^d \) space is represented as a real-valued vector \( d_j^t = (c_j, r_j, r_j^t) \).

Introducing the two different radii for a single detector \( d_j \) was inspired by RST, where \( r_j \) is responsible for the classification of all nonself samples with full certainty (lower approximation), while \( r_j^t \) (upper approximation) is used for determining objects which may possibly be nonself. Base on this assumptions, during the classification stage, censorship sample is classified as:

- **nonself** - when was covered by at least one hypersphere with lower approximation \( r_j \);
- **self** - when was not covered by any hypersphere with upper approximation \( r_j^t \);
- **uncertain** - otherwise (when was covered by at least one hypersphere with upper approximation \( r_j^t \)).

Due to during the learning stage, the distance to all samples have to be calculated, \( r_j^t \) can be determined without increasing the time and computational complexity, in comparing to original \( V \)-Detector algorithm. While the censoring stage, additional verification should be performed only for not recognized samples. However, this operation does not have to be done immediately, since it only applies to uncertainty objects (for example suspicious network connections) and can be done in the idle time of the processor. Figure 2 presents an example of \( v_k \)-detector in 2D space for \( k = 2 \) and Euclidean affinity metric.
5. Experiments

Experiments were performed on two very popular datasets, mainly used for testing intrusion detection systems, namely: KDD Cup 1999 and Kyoto 2006+. Both of them are high dimensional and numerous. To decrease the computation complexity, all tests were performed on samples with specified protocol: ICMP, TCP and UDP. In this paper, only results for ICMP are presented and detailed information about used subsets are presented in Table 1.

| Dataset         | Protocol | No. of attributes | No. of samples | No. of self samples |
|-----------------|----------|-------------------|----------------|---------------------|
| KDD Cup 1999    | ICMP     | 22                | 11911          | 4428                |
| Kyoto+          | ICMP     | 21                | 4069           | 319                 |

Moreover, from both subsets, some attributes were excluded (for example, source and destination MAC addresses in case Kyoto+) due to a met problems with normalization (too many unique symbolic values). In case of KDD Cup 1999, all attributes not related with ICMP protocol were removed. Was obtained a subset consisting of 22 attributes, instead of the original 42.

Tables 2 and 3 shows the obtained results for Kyoto+ and KDD 1999 Cup, respectively. For both testing datasets, it can be observed an increase in length of self radius ($\Delta r$) with increasing the value $k$. As a result, each detector were able to detect more nonself samples at the expense of FAR. However, the influence of $k$ value on increase of DR for KDD 1999 Cup was more significant. For $k = 30$, DR has increased, more than twice, up to 78% from basic 33.9% at the very low FAR (less than 5%). For Kyoto+, for the same $k$ value, DR has increased from 12% to 93%. However, in this case, over 41% self sample were not correctly classified. Hence, the lower $k$ values should be chosen to achieve good balance between DR and FAR.

The column $|V_t|$ contains the percentage of samples classified as uncertain. As you might expect, this value increases with $k$ up to 80% for $k = 30$. Hence, $k$ should be carefully selected to minimize the number of samples, which further should be additionally censored by other algorithms.

The results presented in Tables 2 and 3 were achieved for $L_{0.5}$ affinity metric because dimensionality of testing datasets were to high for Euclidean and Manhattan distances.
Table 2. Results for Kyoto+ dataset.

| $k$ | $\Delta r$ [%] | $DR$ [%] | $FAR$ [%] | $V_t$ [%] |
|-----|----------------|----------|-----------|-----------|
| 0   | -              | 12.8     | -         | -         |
| 1   | 11.8           | 30.9     | 4.2       | 16.1      |
| 2   | 17.3           | 39.7     | 7.9       | 24.2      |
| 3   | 19.3           | 43.5     | 8.8       | 26.6      |
| 5   | 26.5           | 59.8     | 13.9      | 45.1      |
| 10  | 36.9           | 73.0     | 23.8      | 54.3      |
| 20  | 51.6           | 87.4     | 34.1      | 67.8      |
| 30  | 58.2           | 93.6     | 41.5      | 73.6      |

Table 3. Results for KDD 1999 Cup dataset.

| $k$ | $\Delta r$ [%] | $DR$ [%] | $FAR$ [%] | $V_t$ [%] |
|-----|----------------|----------|-----------|-----------|
| 0   | -              | 33.84    | -         | -         |
| 1   | 3.2            | 48.2     | 0.2       | 6.1       |
| 2   | 6.4            | 61.5     | 0.4       | 16.6      |
| 3   | 9.1            | 48.9     | 0.6       | 32.2      |
| 5   | 10.0           | 58.0     | 0.8       | 34.1      |
| 10  | 18.1           | 59.7     | 1.2       | 34.9      |
| 20  | 18.5           | 77.9     | 3.7       | 42.3      |
| 30  | 21.5           | 78.2     | 4.7       | 80.0      |

6. Conclusions

Application of rough sets theory (RST) to the NSA allow for determining the lower and upper approximation spaces in $U$. In this way, an additional class of objects can be defined, except the standard $S$ and $N$, used by default in the immune systems. The uncertain class, defined as a bounding region of rough sets, is used for storing the samples for which the affinity metric to both classes is very similar. For correct classification, an additional algorithms could be applied and then the overall detection rate can significantly increase. This may be particularly important e.g. for security systems where slight deviations from normal connections may be unnoticed. Application of RST is one of the way to deal with this problem.

It is worth to emphasizing, high ratio of $DR$ was obtained without significant increasing the computational complexity. While the learning stage, the process of generating detectors does not require any additional operations as distances to all samples have to be calculated even in the basic version of $V$-Detector algorithm. During the censoring samples, only those not recognized are additionally compared with the radius representing the upper approximation.

Future works should focus on applying the RST to $b$-$v$ model [4] as well as to other popular affinity metrics.

Acknowledgment

This research was partially supported by the grant S/WI/1/2018 of the Polish Ministry of Science and Higher Education.
References

[1] Forrest S., Perelson A., Allen L., Cherukuri R.: Self-nonself discrimination in a computer. In Proc. of the IEEE Symposium on Research in Security and Privacy, Los Alamitos, 1994, pp. 202–212.

[2] Forrest S., Hofmeyr S. A., Somayaji A., Longstaff T. A.: A sense of Self for Unix Processes. Proc. of the 1996 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, 1996, pp. 120–128.

[3] Ji Z., Dasgupta D.: Real-valued negative selection algorithm with variable-sized detectors. Genetic and Evolutionary Computation GECCO-2004, Part I, LNCS Vol. 3102, Seattle, WA, USA, Springer-Verlag, 2004, pp. 287–298.

[4] Chmielewski A., Wierchoń S. T.: Hybrid negative selection approach for anomaly detection. In: A. Cortesi et al. (Eds.) Computer Information Systems and Industrial Management, LNCS 7564, Springer, 2012, pp. 242-253.

[5] Fernandes D.A.B., Freire M.M., Fazendeiro P.A.P., Incio P.R.M.: Applications of Artificial Immune Systems to Computer Security: A Survey. Journal of Information Security and Applications, Volume 35, August 2017, pp. 138–159.

[6] Pawlak Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic, Dordrecht, 1991.

[7] Honko P., Compound approximation spaces for relational data. International Journal of Approximate Reasoning Volume 71, April 2016, pp. 89–111.

[8] Félix R., Ushio T.: Rough Lymphocytes for Approximate Binding in Artificial Immune Systems. 15th International Conference on Electronics, Communications and Computers (CONIELECOMP 2005), Puebla, Mexico, pp. 272–277.

[9] Shen J., Wang J., Ai H.: An Improved Artificial Immune System-Based Network Intrusion Detection by Using Rough Set. Communications and Network, 2012, 4, pp. 41–47.

[10] Chmielewski A. (2017) Application of Rough Sets to Negative Selection Algorithms. In: Dang T., Wagner R., Kng J., Thoai N., Takizawa M., Neuhold E. (eds) Future Data and Security Engineering. FDSE 2017. Lecture Notes in Computer Science, vol 10646. Springer, Cham.

[11] Harmer P. K., Williams P. D., Gunsch G. H., Lamont G. B.: Artificial immune system architecture for computer security applications. IEEE Trans. on Evolutionary Computation, Vol. 6, 2002, pp. 252–280.

[12] Ji Z., Dasgupta D.: Revisiting negative selection algorithms. Evolutionary Computation, vol. 15(2), 2007, 223-251.