Can a negative selection detect an extremely few non-self among enormous amount of self cells?

Akira Imada

Brest State Technical University
Moskowskaja 267, 224017 Brest, Republic of Belarus
akira@bsty.by

Abstract. We have had lots of reports in which they asserted a negative selection algorithm successfully distinguished non-self cells from self cells, especially in a context of “network intrusion detection” where self patterns are assumed to represent normal transactions while non-self patterns represent anomaly. They further assert a negative selection gives us an advantage that we use only a set of self cells as training samples. This would be really an advantage since we usually don’t know what do anomaly patterns look like until they complete an intrusion. We, however, suspect its applicability more or less. This paper gives it a consideration to one of the latest such approaches.

1 Introduction

A sultan has granted a commoner a chance to marry one of his 100 daughters by presenting the daughters one at a time letting him know her dowry that had been defined previously. The commoner must immediately decide whether to accept or reject her and he is not allowed to return to an already rejected daughter. The sultan will allow the marriage only if the commoner picks the daughter with the highest dowry. — “Sultan’s Dowry Problem” [1]

In real world, we have many problems in which it is easy to access to any one of the many candidate solutions which could be the true solution but most likely not, which we don’t know in advance.

The ultimate extreme is called *a-needle-in-a-haystack* problem. The needle originally proposed by Hinton & Nowlan [1] was exactly the one configuration of 20 binary bits. In other words, the search space is made up of 2^{20} points and only one point is the target. No information such as how close is a currently searching point to the needle.

---

[1] According to the author(s) of the web-page of Cunningham & Cunningham, Inc. (http://c2.com) the problem was probably first stated in Martin Gardner’s Mathematical Recreations column in the February 1960 issue of The Scientific American. To explore the problem more in detail, see, e.g., http://mathworld.wolfram.com. We thank Mariusz Rybnik at University Paris XII for suggesting that the problem is reminiscent of our context.

Yet another problem, *a-tiny-flat-island-in-a-huge-lake* — this is a problem we came across when we had explored a fitness landscape defined on all the possible synaptic weight values of a fully-connected spiking neurons to give them a function of associative memory [2]. To simplify it we formalized the problem in more general form as follows.

Testfunction 1 (A tiny flat island in a huge lake) [2] *Find an algorithm to locate a point in the region A all of whose coordinates are in* $[-a, a]$ $(a \leq 1)$ *in an universe of the n-dimensional hypercube all of whose coordinate* x_i *lie in* $[-1, 1]$ $(i = 1, \ldots, n)$.

Many researchers in artificial immune system community have suggested us that the problem might be easy if we use the concept of negative selection. To simply put, the negative selection is an evolutionary selection mechanism by which immune system trains itself only using *self cells* as training samples, so that it can recognize *non-self cells* afterwards.

So, our idea is as follows. We test a set of samples one by one, as many as possible, to know whether each of those samples is the true solution or not. If we happen to find the island during this search procedure, then our goal is attained. This is a matter of if-we-have-good-luck or we should rather be more than lucky. Hence, we train the system in parallel using those samples during the procedure, regardless of whichever the real solution might be found or not as a result. Then at least we can expect that the system will recognize the island later after the training.

In this paper, we approach the problem from this view point. That is, we take it, more in general, just a pattern classification problem, but under the constraint that we have two classes one of which includes an extremely few patterns while the other includes an almost infinite number of patterns. Or, we might as well take it a task of discrimination of a few of non-self cells as anomaly patterns from enormous amount of self cells which represent normal patterns.

One of such latest approaches among others is by Zhou Ji and D. Dasgupata [3]. They wrote

> *The idea of negative selection was from T cell development process in the thymus. If a T cell recognizes self cells, it is eliminated before deployment for immune functionality. In an analogous manner, the negative selection algorithm generates the detector set by eliminating any detector candidates that match self samples. It is thus used as an anomaly detection mechanism with the advantage that only the negative (normal) training data are needed.*

[2] It is not necessarily to be said for the top of the island to be "flat", but the originally this was a test-bed for evolutionary computations, and the fitness of the island region is one and zero in a lake region, that is why.

Let' see, recalling our universe is n-dimensional Euclidean space, two algorithms they proposed: one is to generate detectors of constant sized hyper-spheres and the other is to generate variable sized hyper-spheres. They concluded that detectors which detect anomaly patterns are successfully created just by training with normal patterns.

When we think of a network intrusion detection, we usually don't know what do anomaly patterns look like in advance. Hence this feature of training with only normal patterns is really advantageous. Our concern then is what if the number of non-self cells is extremely smaller than self cells, which is of usual cases when we think of a network intrusion detection. In order to explore this issue, we apply their algorithms to *a-tiny-island-in-a-huge-lake* mentioned above. We can control the difficulty of the task by changing the value of a, and we set it to a very small value. The ultimate case in which the pattern all of whose coordinates shrink to zero, is the problem called *a-needle-in-a-haystack*.

2 Algorithm

As described in the previous section, lots of algorithms to distinguish non-self patterns from self patterns have been proposed. The goal of these algorithms is to create detectors which cover non-self space as much as possible. Here, in this paper, we concentrate on the algorithm called *"Augmented Negative Selection Algorithm with Variable-Coverage Detectors"* proposed by Zhou Ji and D. Dasgupata (2004) [3], as well as its simpler version in which detector size is constant instead of variable, also proposed by the same authors in the same article. The followings are these two algorithms that we paraphrased the original ones with the semantics being intact. Firstly, the simpler version is:

Algorithm 1 (Constant-sized Detector Generation) *After setting (i) N_t, the number of training samples; (ii) r_d, the radius of detector; and (iii) N_d, the total number of detectors:*

1. *Create N_s samples of self cells at random.*
2. *Create a hyper-sphere which has the radius r_d and whose center locates at random in $[-1, 1]$. This is a candidate detector to detect non-self cells.*
3. *If this-hyper sphere does not contain any sample self cells, then put it as a detector in D, the detector's repertoire. Otherwise delete the hyper-sphere.*
4. *Repeat 2-3 until we find N_d detectors.*

This algorithm, in our humble opinion, does not contain the concept of negative selection or whatever in an immune system metaphor neither, if not at all. The second one is:

Algorithm 2 (Variable-sized Detector Generation) *After setting (i) N_t, the number of training samples; (ii) r_s, the radius of self cells; (iii) c_0, expected coverage, i.e., the degree to how much those created detectors cover non-self cells;*

(iv) $c_{\max}$*, the upper bound of self coverage; and (v)* N_d*, the maximum number of detectors:*

1. *Empty* D*, the detector's repertoire.*
2. *Try to find a point* $\mathbf{x} = (x_1, \cdots, x_n) \in [-1, 1]^n$ *which is not contained by any of the valid detectors so far created, unless the number of those trials exceeds* $1/(1 - c_0)$*. If no such* $\mathbf{x}$ *is found, then terminate the run.* [3]
3. *If* r*, the distance between* $\mathbf{x}$ *and its closest self sell in the training sample, is larger than the radius* r_s*, i.e., if the candidate doesn't include any of the sample self cells, then add the sphere whose center is* $\mathbf{x}$ *and radius is* r *to* D *as a new valid detector.*
4. *If such* $\mathbf{x}$ *cannot be found within the consecutive trials of* $1/(1 - c_{\max})$ *time, then terminate the run.* [4] *Otherwise repeat 2 and 3, until we find a total of* N_d *detectors.*

We do not think this algorithm strongly reflects an immune system either, despite the title of the original paper indicates it. However at least the title holds true in the sense that detectors are chosen by trying to match them to the self strings and if a detector matches then it is discarded, otherwise it is kept. This is, above all, what we call a natural selection algorithm.

3 Evaluation of How it Works

We use a measure originally proposed by Lopes et al. [4] in which four quantities, i.e., (i) true-positive, (ii) true-negative, (iii) false positive, and (iv) false negative are used. Here we assume positive sample is non-self and negative sample is self, since detectors is designed to detect non-self cells. Hence, these four terms are defined in a sence that (i) t_p (true positive) — true declaration of positive sample, i.e., non-self declared as non-self (ii) f_p (false positive) — false declaration of positive sample, i.e., self declared as non-self (iii) t_n (true negative) — true declaration of negative sample, i.e., self declared as self (iv) f_n (false negative) — false declaration of negative sample, i.e., non-self declared as self. Under these definitions $d_r = t_p/(t_p + f_n)$ implies detection rate, and $f_a = f_p/(t_n + f_p)$ implies false alarm rate.

4 Experiment, Results, and Discussion

We assume here the whole universe is n-dimensional hyper-cube $[0, 1]^n$ as mentioned already; any point all of whose coordinates lie in $[(0.5 - a), (0.5 + a)]$ where $0 < a < 0.5$ are a non-self cell, whilst other points in the universe are self cells [5] ; and all the self cells are hyper-sphere whose radius is r_s.

[3] This is because when we have sampled m points and only one point was not covered, the expected coverage is $1 - 1/m$. Hence the necessary number of tries to ensure expected coverage c_0 is $m = 1/(1 - c_0)$.

[4] See also the footnote above replacing c_0 with $c_{\max}$.

[5] We modify our Testfunction 1 for the sake of simplicity of coding in this way, which keeps the problem equivalent to the original one.

4.1 A 2-dimensional version of an-island-in-a-lake

First of all, in order for our eyes to be able to observe the behavior of the algorithms our experiment is performed on a 2-dimensional space, that is, we set $n = 2$. We employ a set of 500 randomly selected points in the self region as the training samples, and 1000 points randomly chosen from entire space is the test data. The reason of these settings is to enable us to compare our results with their's in the original proposition [3].

The location of the self points in the training sample and the created detectors when we set $r_s = 0.1$ which is the value recommended by the original proposition [3] are shown in Fig. 1. So far so good. However, our goal is to recognize

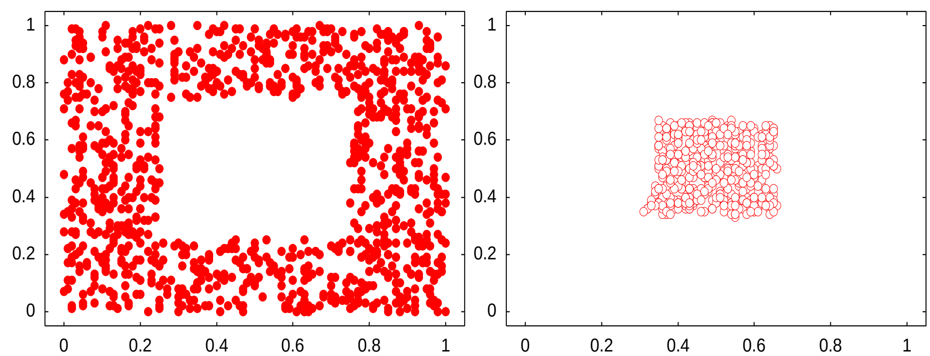

Fig. 1. A set of five hundred self-points employed as training samples (Left), and a set of five hundreds detectors created by the Algorithm-1 with $a = 0.25$ and $r_s = 0.1$ (Right) from an experiment in 2-dimensional space.

non-self patterns from extremely tiny region. Hence the next experiment is a dependency on the value of a. Fig. 2 shows the number of required trials to find the pre-defined number of detectors, which is 500 here, and the number of successes when those 500 detectors tried to detect the 500 non-self samples. Both are plotted as a function of value of a using the Algorithm-1 with $r_s = 0.1$ to create the detectors. As we can see in the Figure, the difficulty of the task becomes harder exponentially as a becomes smaller, and therefore we know this algorithm would not work if the region to be searched for is extremely tiny.

4.2 A 20-dimensional version of an-island-in-a-lake

Here experiments are performed on a higher dimensional version of our test-bed *a-tiny-flat-island-in-a-huge-lake.*

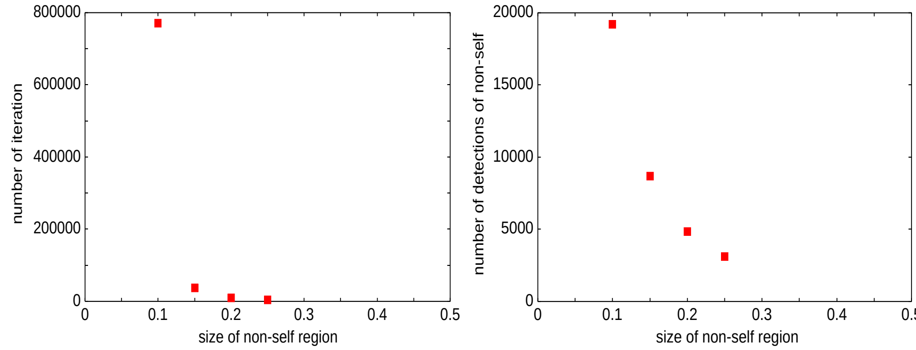

Fig. 2. The number of iteration required to find 500 successful detectors (Left), and the number of successes when 500 detectors tried to detect 500 non-self samples, that is, the number of successes out of 25000 events (Right). Both are as a function of value of a when we experimented with the Algorithm-1 with $r_s = 0.1$ in 2-dimensional space.

In a preliminary experiment, we found it much more difficult than in the case $n = 2$. What we found, for example, is even if we increase the number of training sample of self patterns from 1000 to 10000, the distribution of the sample is very sparse under the condition of $n = 20$. If the algorithm worked well, the detector would be supposed to locate in the non-self region, such as Fig. 3 (Right) which is from a result of 2-dimensional experiment for comparison purpose, while the result in 20-dimensional experiment, as shown in Fig. 3 (Left), was not in that way. The coordinates of whole detectors are almost uniformly distributed, which

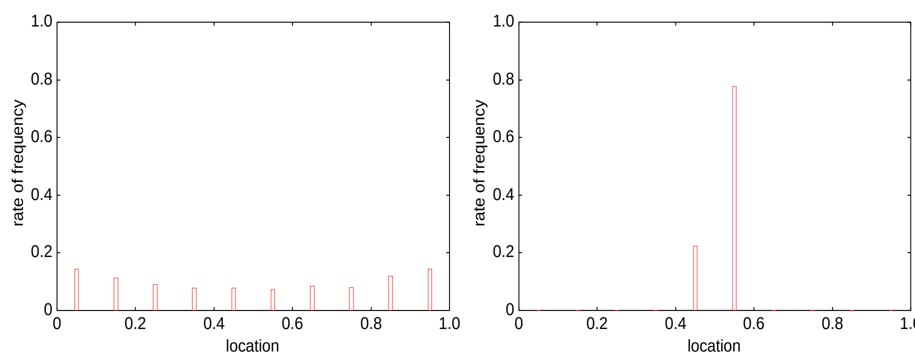

Fig. 3. The distribution of all the coordinates of the detectors for an experiment with $n = 20$ (Left), and the distribution when $n = 2$ for the purpose of comparison (Right).

means a failure to find a set of successful detectors.

Then, we give it a consideration of how will the Algorithm-2 (Variable Sized Detector) improve the situation. In an experiment in 20-dimensional space where the Algorithm-2 creates certain number of detectors with 1000 training samples of self patterns. Non-self region in this experiment was $[0.495, 0.505]^{20}$ and radius of self was set to 0.1. As a result of a run under $c_0 = 0.99$, a total of 96 detectors are created. We ran the algorithm for $n = 5$, 15, 20, and 25 to study a dependency of the degree to how successfully the detector will be created on the dimension of search space. The number of detectors created is somehow similar in each dimension, ranging from 91 to 96. In Fig. 4, we show detection-rate and false-alarm-rate as a function of dimensionality. Though not satisfactorily, we see somewhat of a succeeded result, at least as for detection-rate.

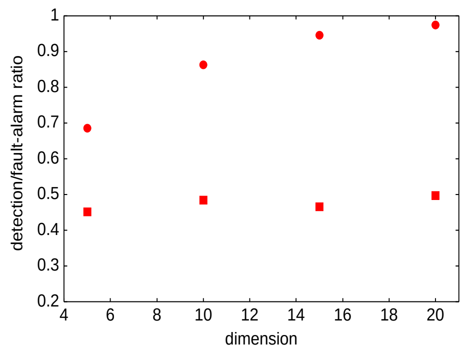

Fig. 4. Detection-rate (circles) and false-alarm-rate (rectangles) as a function of dimensionality in a series of experiments where the Algorithm-2 creates certain number of detectors (from 91 to 96) with 1000 training samples of self patterns.

Further, we will explore different parameter values with the goal being to learn the limit of how much small non-self region allows the algorithm to detect non-self points successfully. Then we will experiment by lowering the value c_0 which is 99.99% and 99% in the original version, although results are not shown here since our experiments have sometimes reversed our expectations so far.

5 Conclusion

We have obtained the similar results with the experiments by Zhou Ji and D. Dasgupata [3] when the domain of non-sell is not so small.

Usually, however, in the real world problem, anomaly patterns are extremely fewer than the normal ones. As such, our concern is on an extreme case. Unfortunately, we have not so far observed any satisfactory results under this extreme situation. In fact, Zhou Ji and D. Dasgupata [3] wrote

> *As an exception, the algorithm may also terminate when it fails to sample any non-self point after many repetitions. That implies that the self region covers almost the entire space. It may happen when the self samples are randomly distributed over the space, or the chosen self-radius is too big.*

And as they went on to write concerning another experiment in the same paper [3] *"One of the three types of IRIS data is considered as normal data, while the other two are considered abnormal,"* the number of normal and abnormal is usually comparable in such experiments.

We are exploring a number of other different approaches to the same target, that is, *a-tiny-flat-island-in-a-huge-lake* or its binary version *a-needle-in-a-haystack*. What we have tried so far are experiments by means of (1) a negative selection of binary detector using r-contiguous matching; (2) evolution of a set of fuzzy rules (3) a data-mining techniques and so on..., to detect a *tiny-island* or a *needle*.

Though still a lot of experiments have been resistant to be analyzed this series of works is not to show a counter example for an assertion but to call for a challenge. The objective is to detect anomaly phenomena which take place only occasionally and hence we don't know what does it look like, while we have enormous amount of daily normal phenomena. As far as we know this is still an open issue and we try to find some approaches, or at least to evoke interests in this problem in our community. The challenge is awaiting us.

References

1. G. E. Hinton and S. J. Nowlan (1987) *How Learning can Guide Evolution.* Complex Systems, 1, pp. 495–502.
2. A. Imada (2004) "How a Peak on a Completely-flatland-elsewhere can be Searched for? — A Fitness Landscape of Associative Memory by Spiking Neurons." Proceedings of Advanced Computer Systems (ACS) and Computer Information Systems and Industrial Management Applications (CISIM), Vol.2, pp. 171–150.
3. Zhou Ji and D. Dasgupata (2004) "Augmented Negative Selection Algorithm with Variable-Coverage Detectors." Proceedings of the Congress on Evolutionary Computation. pp. 1081–1088.
4. H. S. Lopes, M. S. Coutinho, W. C. Lima (1997) "An Evolutionary Approach to Simulate Cognitive Feedback Learning in Medical Domain." Genetic Algorithms and Fuzzy Logic Systems. World Scientific, pp. 193–207.