Applicability Issues of the Real-Valued Negative Selection Algorithms

Zhou Ji
The University of Memphis, Memphis, TN 38152
and
St. Jude Children's Research Hospital
Memphis, TN 38138
zhou.ji@stjude.org

Dipankar Dasgupta
The University of Memphis
Memphis, TN 38152
dasgupta@memphis.edu

ABSTRACT

The paper examines various applicability issues of the negative selection algorithms (NSA). Recently, concerns were raised on the use of NSAs, especially those using real-valued representation. In this paper, we argued that many reported issues are either due to improper usage of the method or general difficulties which are not specific to negative selection algorithms. On the contrary, the experiments with synthetic data and well-known real-world data show that NSAs have great flexibility to balance between efficiency and robustness, and to accommodate domain-oriented elements in the method, e.g. various distance measures. It is to be noted that all methods are not suitable for all datasets and data representation plays a major role.

Categories and Subject Descriptors

I [**Computing Methodologies**]: Miscellaneous

General Terms

Algorithms

Keywords

Negative selection algorithms, One-class classification

1. INTRODUCTION

The paradigm of negative selection algorithms (NSA) is one of the earliest models developed in the area of Artificial Immune Systems (AIS) [3]. It has been used in various applications [13, 19] and different variations of this type of algorithms are also being proposed [1, 15]. However, along with similar thinking about AIS in general, researchers kept raising questions about whether this method provides anything unique and anything more powerful than alternatives, and when and whether it is appropriate to use it. It is believed that negative selection algorithms have distinct process compared with other algorithms and may be most suitable for certain applications [5]. It was pointed out that negative selection algorithms are not appropriate to be used as a general classification method because they use samples from one class in training [4].

Among the latest works reviewing negative selection algorithms, Stibor et al [16, 17] raised serious concern about negative selection algorithms' usability, taking *V-detector* [11, 12] as a specific case to compare. As one of the new variations, *V-detector* has some unique features to be more reliable and efficient than previous negative selection algorithms. It was originally discussed in real-valued representation, though its unique detector generation process is not limited to specific data representation.

This paper tries to tackle the issues discussed in those works [16, 17]. We will see that some issues are not really attributed to the main issue in question, namely negative selection algorithms as a distinct type of method. In many cases, they are not intrinsic to *V-detector* either. The concerns about negative selection algorithms' applicability and significance are not all answered, especially the question whether we can strongly conclude that negative selection algorithms are irreplaceable to solve certain problems. Such killer applications still have not been found. Nevertheless, this paper clarifies some confusions that may mislead readers about the applicability and weakness of negative selection algorithms, especially those in real-valued representation.

In the next section, we will go through several issues that have drawn attention. Then, in section 3, we illustrate the problems and properties of *V-detector* with several groups of experiments, sometimes with comparison with SVM (Support Vector Machine). Also, some new developments of *V-detector* are introduced. Features such as different distance measures serve as a good example showing that negative selection algorithms are not necessarily limited to some specifics of the elements of the algorithm. Instead, NSA are more a family of algorithms based on some minimal common framework but with diverse details. Lastly, section 4 summarizes the current status and potential research need of negative selection algorithms.

2. APPLICABILITY ISSUES

2.1 Curse of dimensionality

High-dimensional data is a general difficulty for classification and other machine learning problems. Many algorithms cannot scale very well with the number of dimensions. The

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

search space grows exponentially larger as the number of dimensions increases. The similar issue appears in different ways in different paradigms and the actual severity also depends on the specific application and the property of the data involved. Nevertheless, it is a problem more fundamental than choosing a specific technique.

On one hand, the problem is how to represent any region in high-dimensional space. In negative selection algorithms, the question is in the number of detectors that is considered enough to cover the abnormal space. Variable-sized detector proposed in *V-detector* is an easy but powerful solution. In *V-detector*, variable-sized detectors are in fact maximized, meaning that instead of concerning with overlap as in earlier models [7, 8], each detector takes the maximum size that is allowed by the self samples. This way, the large continuous nonself region has a good chance to be covered by fewer number of detectors.

On the other hand, a more fundamental difficulty is that when the dimensionality is high, the relatively smaller number of samples could not really catch the characteristics of the target concept. Consider a self region that is a n-dimensional hypersphere. If we have 1000 sample points, we are readily convinced that the points can represent the shape of the actual region in case of 1-dimension (circle) or 2-dimension (sphere). To the minimum, though, it is evident that we need at least 2 points to represent the extent in 1-dimensional case. Similarly, we need at least 4 points to have the roughest representation in 2-dimensional case. In 10-dimensional case, 1000 points are barely enough to have one single point for each dimension. That is similar to trying to use 3 points to represent a 3-dimensional shape. *V-detector* or other negative selection algorithms have not provided a direct remedy to this problem.

In section 3, we will present some empirical results of *V-detector*'s behavior when the number of dimensions increases. The difficulty in theoretical analysis partly lies in the fact that actual configuration of self and nonself regions is the deciding factor of a practical problem. For example, if the entire search space, for example, a unit hypercube, is nonself space, it is trivial to cover with one detector; if the self region is a single point at the origin, we can cover the nonself region with finite number of detectors and may be able to find the minimum possible number mathematically; if the nonself region is one connected region that has the same shape as the detectors, it is possible to be covered by one single detector.

2.2 Real-valued representation

Negative selection algorithms can be modeled in any data space. The most widely used data representation is binary or string representation. The real-valued representation is another common one. Gonzalez et al [6] showed that blindly using binary representation will break the data proximity in the problem space and cannot provide meaningful generalization. The same argument holds true for real-valued representation.

Real-valued representation only makes sense if the underlying problem is continuous and we have proper real numbers to represent some continuous properties in the problem. Let us consider the KDD Cup 1999 data [17]. Each record has 41 features, 9 of which are in fact discrete. In the reported experiments [17], these discrete fields are converted to a natural number arbitrarily and then normalized between minimum and maximum. Such conversion has two essential flaws:

- Because of those discrete fields, the converted points will be distributed on separated (parallel) planes in the real space. The distance within one plane should not be interpreted in the same way as the distance between the planes. The connotation of being closer or farther apart is not the same as in the original data space.

- The different values come from the assignment of natural numbers to the original discrete fields. In most cases, the original discrete values don't have an intrinsic order. It is only meaningful to talk about being different or being the same. The artificial order and the distance resulted in the converted values is thus totally arbitrary. For example, the difference between two items converted as 0 and 1 is not smaller in any way than the difference between two items converted as 0 and 2.

Because the entire process of a negative selection algorithm, or as a matter of fact, of any learning algorithms, is built on the concept of affinity or distance, the results based on such converted data may be fallacious. When it works, it is more likely to be merely good luck. For example, when the matching threshold is small enough, actually two points with different discrete values are never considered matching each other. In such cases, the converted real-valued fields not only fail to contribute to measure the distance or affinity between the two points, they also limit the reasonable choice of threshold for other real real-valued fields.

Freitas et al [4] strongly suggested different weights in distance formula. That idea has the same motivation as described above. In a more general way, we could construct customized distance measure/matching rule for a given hybrid data representation. If the data item is in the form of $(x_1, x_2, \cdots, x_I, y_1, y_2, \cdots, y_J)$, where x_i, $i = 1, \cdots, I$ are real numbers and y_j, $j = 1, \cdots, J$ are discrete values, we can define a customized distance measure comprised of two parts: one is Euclidean distance over $(x_1, x_2, \cdots, x_I)$; the other is Hamming distance over $(y_1, y_2, \cdots, y_J)$. The two parts can be combined with proper weight or used in a matching rule that also has two corresponding parts.

2.3 Apparent limitations

2.3.1 Limitation of specific matching rules

Following the above discussion, we notice that matching rule, which usually takes the form of a distance measure plus a matching threshold, plays a very important role. Visually, the same concept can be expressed as the geometric shape of the detectors. Hart [9] noticed that importance of choosing the proper recognition region, which refers to the similar idea as the shape of detectors. It should be pointed out that this is as important in any other AIS systems or any learning paradigms as in negative selection algorithms.

Sometimes, the apparent limitation of a negative selection algorithm is in fact the limitation of a specific matching rule or detector shape. For real-valued negative selection algorithms, Euclidean distance and therefore hyperspherical detectors are commonly used, but they are not the only possibility. Limitation of Euclidean distance or hyperspherical detectors is not the special problem of real-valued negative

selection algorithms. In fact, other matching rules and detector shapes were used in several works, for example, rectangular detectors [2], hyper-ellipsoid detectors [15], etc.

Negative selection algorithms are methods with great flexibility. First, the concept of negative selection can be realized in very different flavors of so-called negative selection algorithms. Second, even for a specific negative selection algorithm, e.g., *V-detector*, there are many elements in the model that are not inherently limited as it appears. For example, for real-valued representation, which is not necessarily the only choice in the first place, we could use very different distance measures or matching rules. In section 3, we will see that Euclidean distance can be easily extended to be L_m distance, which could result in different shapes of detectors.

2.3.2 Limitation of one-class classification

Generally speaking, performance of a classification algorithms or a learning method depends on the probability distribution of the data. Any serious analysis cannot be done without taking into consideration that distribution. One-class classification, however, is an effort to learn when no information of the second class is available [18]. That means that the probability distribution of the abnormal data (or nonself data) is never known according to the basic assumption. That is the main reason that Freitas et al [4] casted the doubt on negative selection algorithms. On the other hand, one-class learning is a valid need and has been studied from various aspects and used in many applications [14]. In summary, limitation does exists, but it is not specific to negative selection algorithms or *V-detector*.

It is noteworthy that the probability distribution of only self space could be taken into account in one-class classification, including negative selection algorithms.

2.4 Negative or positive selection?

Both positive selection and negative selection can be a reasonable choice according to the specifics of the problem. One obvious scenario when negative selection is preferred is when we have a large number of self samples and we can potentially generate a relatively small set of detectors.

The straightforward positive selection proposed a couple of times by Stibor et al [16, 17] is hardly a valid alternative in most cases. When the number of self samples are larger than the number of detectors by more than one order of magnitude [17], such positive selection method like Self-Detector is not a realistic solution. The choice between negative and positive selection can be based on various reasons. There are at least two factors that cannot be ignored. First, in many applications, a large amount of normal data is typical, e.g. in network intrusion detection problem. These data are not only too large to be used directly to find anomaly, but contains too much duplicated or similar information. Second, if a method does not provide any inductive ability, for example, clustering or rule extraction, from the raw data, it is not really a learning method, strictly speaking.

2.5 Coverage versus detection rate in NSA

Detector coverage and detection rate are two terms that may lead to misunderstanding when we discuss how well the detector set works. Failure to make clear distinction may muddle otherwise clear analysis.

Coverage is the proportion of nonself space that is covered by detectors. For a given instance, we usually don't know the actual value because the nonself space is the unknown we are seeking for. If we discuss coverage in terms of a number, we are making assumption about how the nonself space (or self space) can be induced from the self sample points we have, at least conceptually. Detection rate, on the other hand, refers to the percentage of nonself sample points that are detected by the detector set in a particular experiment. Thus, the difference is two-folded:

- Coverage depends on how we interpret the training data set. Even for a defined set of detectors, the value of coverage must be based on some assumption that cannot be verified. For example, Stibor et al [17] showed examples of coverage provided by *V-detector*[11] at termination. Nine self points were used to train the system. The discussion of coverage was based on the assumption that real self region is all the perfect circles around the training points. *V-detector*'s termination is decided by estimated coverage. Detection rate is influenced by the coverage as well as the validity of the assumption or interpretation we make about the training data.

- Coverage is the ratio of covered nonself space to the entire nonself space. The probability distribution is usually not considered to evaluate the coverage. Detection rate, on the other hand, depends on the actual frequency distribution of test data. The distribution is usually reflected in the real data. This exposes a weakness of *V-detector*'s termination criterion. The statistical estimate of coverage using random sampling does not take into consideration the probability distribution of the data to be detected. Thus the conclusion of enough coverage or not are always bias depending on how different the actual distribution is from uniform distribution. Logically, this cannot be totally solved because the self training data at best can only provide distribution of the self space.

2.6 Test with training data

Usually training data should not be used to test the system trained by them. When the data are lacking, sometime we could include training data as part of test data depending on the learning algorithm in question. For example, if we extract rules from the training data, it is still valid verification if we use some seen data to test the rules. As in the straightforward positive selection [16, 17], when the target concept is represented by the training data themselves, it does not reflect how the system works if we test with the training data. The even worse practice is when all the self data in test data are those already used to train the system. That means if we let the system be overfitted to the extreme, we're going to see the perfect results. For one-class classification like negative selection algorithms, training data are all self data. Detection rate won't be affected directly, but false alarm can be misleadingly perfectly 0.

2.7 Real challenges to negative selection algorithms

While the problems in certain algorithm variations or experiments were sometimes mistaken as the problem of negative selection algorithms, the real difficulties are often neglected. Other than the general difficulties in learning algo-

rithms, such as high dimensionality mentioned above, there are some real challenges for negative selection algorithms:

- The key role of matching threshold (or self radius, self threshold etc.) It is important to use it for striking a balance between being aggressive and conservative to raise alarm, but it is unknown generally. We could develop some automatic mechanism to choose the value, but it seems hardly possible to have a universally reliable way to decide.
- Mapping from problem space to data representation. It is probably more important than the choice of learning algorithms from a practical point of view. We need to represent the original problem space in proper feature space so we can differentiate between self and non-self. Many high dimensional feature space can be reduced to lower dimensionality by using other method.

Again, it is arguable whether those issues are specific to negative selection algorithms, but they are the difficulties that are currently beyond the reach of negative selection algorithms.

The source code and some documentation of the authors' implementation of *V-detector* is available at `http://umpeople.memphis.edu/zhouji/vdetector.html`. That will serve as a convenient resource to encourage more examination of the method and negative selection algorithms in general.

3. EXPERIMENTS

This section further presents some results either to support or to illustrate some of discussion in the previous section.

3.1 How V-detector failed for KDD data

Stibor et al [16] showed that *V-detector*'s performance is generally very close to the popular statistical learning algorithm SVM (Support Vector Machine). A positive selection algorithm named "Self-Detector" was proposed to compare with *V-detector*. It worked well with the relatively smaller number of self samples. As we discussed in the previous section, such a straightforward positive selection could not serve as a general replacement for negative selection algorithms.

As reported in Stibor et al's later work [17], the performance of *V-detector* on the KDD Cup 1999 data is unacceptably poor. The detection rate is around 1% depending on the control parameter. However, if we look more closely at the results, we will notice the direct reason that *V-detector* failed in these experiments: the number of detectors in the generated detector set is around 1. To put it in another way, termination condition failed and obscured all the other possible problems and the chance to succeed. In fact, the later development of *V-detector* [12, 10] that were published before those experiments has solved the weakness of termination condition to a large extent. Statistically more sophisticated estimate were introduced to decide when to stop adding more detectors [12]. A 'boundary-aware' version of *V-detector* [10] abandoned the implicit assumption that the self samples are all independent internal points, which were used by the earlier version of *V-detector* and other similar methods. It would detect anomaly more aggressively because the detectors can actually touch individual self sample point as long as there is no other self samples around to eliminate them.

Table 1 compares the difference between the algorithm considered in [17] and new variations of *V-detector*. For the purpose of comparison, 20 subsets were extracted from the enormous KDD data using a process described in [17]. Each of these subsets is independently drawn from the entire set and all keep the same proportion of self (normal) data as in the original set. The results of *V-detector* shown in table 1 are for one such subset, but the numbers are mean and standard deviation of 100 repeated runs with same control parameters. This is a little different from getting statistics of 20 different datasets [17]. Although those 20 subsets are drawn randomly from the entire set, it avoids unnecessary complication to treat them as independent datasets.

Figs. 1 and 2 show the results of all the 20 subsets. Fig. 1 is the results using self threshold $r_s = 0.05$. Fig. 2 shows similar results when self threshold r_s is 0.1. These results consistently show that the proper new variations of *V-detector* can change the outcome dramatically and obtain satisfactory detection.

3.2 More reliable experiments

The setting in the above experiments is not ideal. It was chosen that way for the purpose of comparison to highlight the difference between the variations of the algorithm. One of the factors is the process to convert from raw data to a real-valued vector. The other is that testing only with seen training data, which may generate misleading results, even though the abnormal data are unseen in this case because it is a one-class classification. To verify *V-detector* is indeed able to detect well, we used different data to test the detector set. We took one subset to train the system and tested with all other subsets. The results using self threshold $r_s = 0.1$ are summarized in Fig. 3. *V-detector*'s results are consistent with different testing data sets. Similar tests using different self thresholds ($r_s = 0.01$ and $r_s = 0.05$) were also carried out. The results are very close to what is presented here.

3.3 SVM is not always the preferred solution

SVM did very well in KDD data even when we redid the test with different subsets. However, such good results do not guarantee that it can replace alternative methods like *V-detector* under any conditions. As a simple example, let us consider a scenario when *V-detector* is much easier to use than SVM. Two cases were designed so that the self region is a disconnected region. (1) Fig. 4(a) is a self region that is a circle partially cut by a cross, which we will call "intersection". This is one of the synthetic data sets tested in earlier work [12]. (2) Fig. 4(b) is a self region made of five small circles. Both are over the unit square 2-dimensional search space.

Tables 2 and 3 show clearly that SVM does not work as well as negative selection algorithm when default kernel function is used as in previous experiments. That means at least we need to choose proper kernel function to make SVM work. The correct choice depends on extra knowledge of the problem. *V-detector* got significantly better results without the need to refine the control parameters.

3.4 Different distance measures

As mentioned in the previous section, distance measure is an important element in negative selection algorithms. In Euclidean space R^n, commonly used Euclidean distance, or 2-norm distance, can be generalized to Minkowski dis-

Table 1: Comparison of results using different variations of V-detector

	detection rate	SD	false alarm rate	SD
Reported result [17] $r_s = 0.05$	1.21	4.59	0	0
Reported result [17] $r_s = 0.1$	0.65	3.46	0	0
Using the same setting to compare $r_s = 0.05$	0.46	2.16	0	0
Using the same setting to compare $r_s = 0.1$	0.52	3.13	0	0
Statistically more reliable termination [12] $r_s = 0.05$	27.93	0.35	0	0
Statistically more reliable termination [12] $r_s = 0.1$	26.79	2.15	0	0
Boundary-aware algorithm [10] $r_s = 0.05$	81.19	30.49	1.46	0.07
Boundary-aware algorithm [10] $r_s = 0.1$	83.92	28.96	1.45	0.07

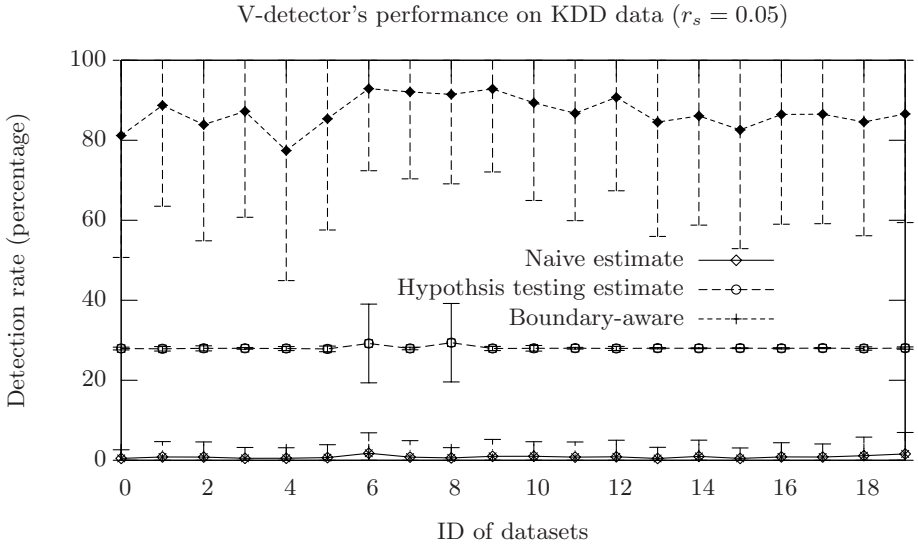


Figure 1: Results from different versions of *V-detector* $r_s = 0.05$

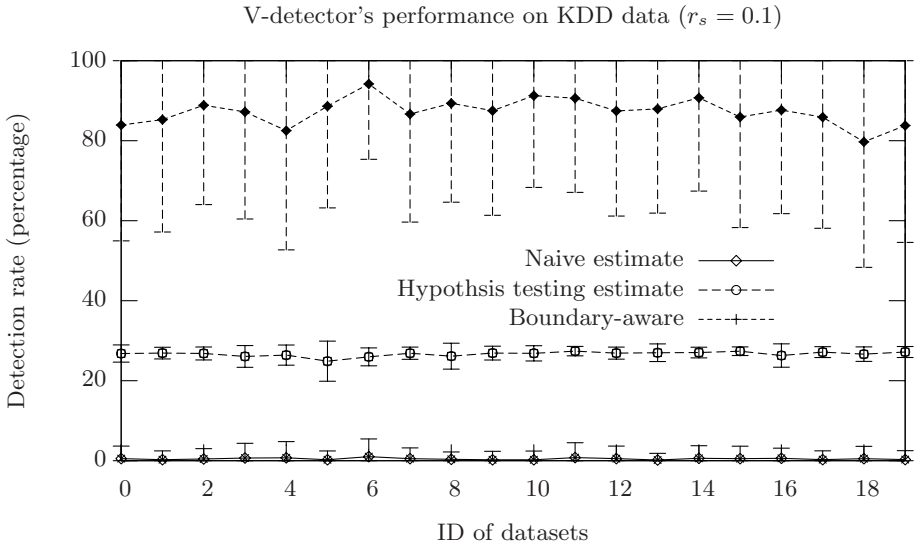


Figure 2: Results from different versions of *V-detector* $r_s = 0.1$

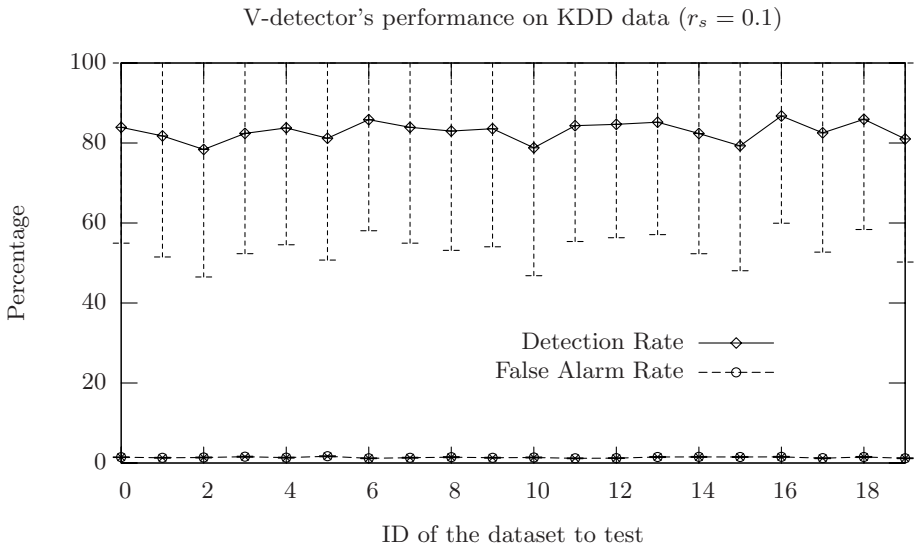
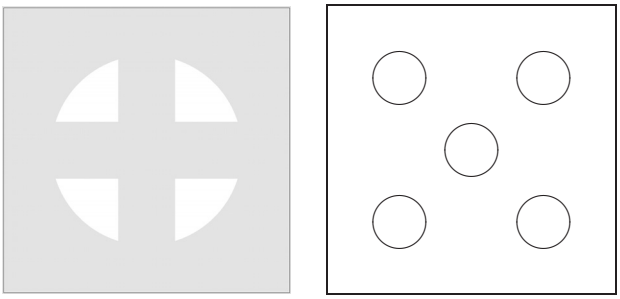


Figure 3: Detection results that were tested with totally unseen data



(a) 'Intersection'-shaped (b) '5 circles'-shaped

Figure 4: Two shapes of self region

Table 2: Results over Intersection self region

	detection rate	false alarm rate
SVM $\nu = 0.05$	77.67	6.25
V-detector $r_s = 0.05$	99.82	11.44
SVM $\nu = 0.1$	81.84	54.69
V-detector $r = 0.1$	96.58	9.69

Table 3: Results over 5-circles self region

	detection rate	false alarm rate
SVM $\nu = 0.05$	47.51	4.38
V-detector $r_s = 0.05$	99.96	11.21
SVM $\nu = 0.1$	65.58	49.64
V-detector $r = 0.1$	97.63	8.33

tance of order m, or L_m distance, for any arbitrary m. For a point $(x_1, x_2, \cdots, x_n)$ and a point $(y_1, y_2, \cdots, y_n)$ in n-dimensional space the 1-norm distance is Manhattan distance

$$\sum_{i=1}^{n} |x_i - y_i|.$$

The m-norm distance is defined as

$$\left(\sum_{i=1}^{n} |x_i - y_i|^m \right)^{\frac{1}{m}}.$$

The infinity norm distance is defined as

$$\lim_{m \to \infty} \left(\sum_{i=1}^{n} |x_i - y_i|^m \right)^{\frac{1}{m}} = \max(|x_i - y_i|, i = 1, 2, \cdots, n)$$

For different norm, the detector (or recognition region) will take different geometric shapes and have different covering area. Fig. 5 illustrates the different shapes in 2-dimensional space. They are shown with the same radius. If we use radius r to indicate the size, r can be interpreted as the radius of the circle in the case of 2-norm distance. For Manhattan distance, the detector is a 45°-turned square whose edge is $\sqrt{2}r$; for infinity norm, the detector has the shape of a square whose edge is $2r$; for any norm between 2 and ∞, the shape is evidently between the radius r circle and the edge $2r$ square.

Tables 4 and 5 are the results obtained using different distance measures, for the "intersection" self region and the "5-circles" self region, respectively. There are two different implementations of Euclidean distance. One is the default setting of *V-detector*, in which the distance measure and matching process are actually implemented using the square of Euclidean distance for better performance in speed. The other Euclidean distance is implemented as L_2 distance in the general way. In term of detection results, there seems to be little difference between different distance measures for

Table 4: Effects of different distance measure: 'Intersection' shape

Distance measure	detection rate	SD	false alarm rate	SD
Euclidean (default efficient implementation)	96.36	1.49	9.77	1.58
Manhattan	97.05	0.83	10.53	2.12
Euclidean	96.23	1.57	9.59	1.38
3-norm	94.69	1.79	9.55	1.56
infinity norm	89.62	3.01	11	1.5

Table 5: Effects of different distance measure: 'Five circles' shape

Distance measure	detection rate	SD	false alarm rate	SD
Euclidean (default efficient implementation)	97.64	0.58	8.2	0.85
Manhattan	99.38	0.62	9.84	1.55
Euclidean	97.63	0.57	8.25	0.8
3-norm	98.16	0.23	8.25	0.6
infinity norm	98.68	0.1	9.28	0.79

these two examples, except that the Manhattan distance is slightly more aggressive to raise alarm of anomaly. However, the running time of the algorithm is noticeably different with different distance measures. The ∞ norm distance is the fastest. For general L_m distance, the algorithm runs slower for higher m.

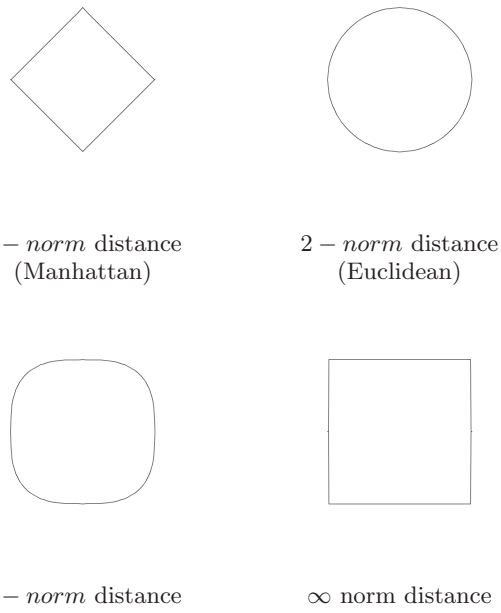


Figure 5: Various geometric shapes of detector (recognition region) corresponding to different m-norm distance

3.5 V-detector's behavior at high dimensions

The analysis of the behavior at high dimensionality is limited by the actual configuration of self space. Here we consider a simple experiment to get some empirical idea about *V-detector*'s performance when the number of dimensions increases. The searching space is a n-dimensional unit hypercube $[0,1]^n$. Assume we have 1000 training points randomly distributed over the real self region, a hypersphere centered at the center of the hypercube. We generated the detector set using those points as self samples and then tried to classify 1000 test points that were randomly drawn from the entire hypercube. Table 6 shows the results from $n = 3$ through $n = 12$. The self threshold is chosen as 0.1.

We need to keep in mind that 1000 training points are not adequate to represent a high-dimensional space. It is definitely not enough when $n > 5$, as we discussed earlier. Besides that, there are a few interesting facts that are noteworthy. First, the number of detectors didnt go up dramatically. That indicates that *V-detector* may be able to deal with the problem of representing a high-dimensional nonself space. Second, detection rate stays well. That in some sense confirms the reliability of detector set generated. Whats wrong with the false alarm? It is not too high, but starts to have large standard deviation even for not too high dimensionality ($n = 5$). The reason is that the uniformly distributed test data have much fewer normal data in it when dimensionality goes higher. When $n \geq 7$, all the testing data turned out to be abnormal data.

Another fact needs to be pointed out. When we use *V-detector* in a real application, we usually normalize the data based on the range of the normal data available. Consequently, it is less likely, though still possible, to have a situation that the normal data only occupies a small fraction of the entire search space as in the above experiment.

4. CONCLUSIONS

There are only a few core elements in negative selection algorithms, mainly representation in negative space and usage of detector sets. Consequently, this family of algorithms is very flexible to accommodate different strategies in it. As one of the latest variations, *V-detector* demonstrates such flexibility very well.

Just like any other machine learning methods, negative selection algorithms, e.g. *V-detector* cannot be used blindly to a problem without proper data representation and reasonable choice of control parameters or algorithm variations. Negative selection algorithms are good for some applications and may adapt to suit some others. Currently ongoing researches on this topic support the potential of negative selection algorithms. There are also some problems that negative selection algorithms are not meant for, for example, the problem with a small number of self samples, or general

Table 6: Behavior of *V-detector* at high dimensionality

dimensionality	detection rate	SD	false alarm rate	SD	number of detectors	SD
3	100	0.01	7.7	1.47	500	0
4	99.94	0.07	13.11	6.92	500	0
5	99.96	0.05	25.67	14.03	500	0
6	99.98	0.04	36	48	500	0
7	99.95	0.08	N/A	N/A	500.04	0.4
8	99.79	0.14	N/A	N/A	500.25	1.22
9	99.6	0.22	N/A	N/A	502.32	3.36
10	99.4	0.23	N/A	N/A	511.11	8.06
11	99.26	0.35	N/A	N/A	534.62	11.29
12	99.16	0.38	N/A	N/A	567.88	15.65

classification problem where probability distribution plays a crucial role.

Although the extent of negative selection algorithms' applicability is still an open question, many difficulties reported in recent years are not attributed to the algorithm itself. The real challenges for negative selection algorithms do exist but are often more general than a certain algorithm. For example, the the difficulty of high dimensionality, decision on optimal control parameters, and a good data model of the application domain are all important issues.

There are many aspects that are worth further exploration for negative selection algorithms. The effect of data distribution is a factor that needs careful investigation considering negative selection algorithms' general assumption that the abnormal samples are not available.

5. ACKNOWLEDGMENTS

This work was supported in part by NIH Cancer Center Support Core Grant CA-21765 and the American Lebanese Syrian Associated Charities (ALSAC).

6. REFERENCES

[1] H. T. Ceong, Y.-I. Kim, D. Lee, and K.-H. Lee. Complementary dual detectors for effective classification. In *Proceedings of Second International Conference on Artificial Immune System (ICARIS 2003)*, 2003.

[2] D. Dasgupta and F. Gonzalez. An immunity-based technique to characterize intrusion in computer networks. *IEEE Transactions on Evolutionary Computation*, 6(3):1081–1088, June 2002.

[3] S. Forrest, A. Perelson, L. Allen, R., and Cherukuri. Self-nonself discrimination in a computer. In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, Los Alamitos, CA, 1994. IEEE Computer Society Press.

[4] A. A. Freitas and J. Timmis. Revisiting the foundation of artificial immune systems: A problem-oriented perspective. In *Proceedings of Second International Conference on Artificial Immune System (ICARIS 2003)*, 2003.

[5] S. M. Garrett. How do we evaluate artificial immune systems? *Evolutionary Computation*, 13(2):145–178, 2005.

[6] F. González, D. Dasgupta, and J. Gómez. The effect of binary matching rules in negative selection. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2003)*, LNCS 2723, pages 195–206, Chicago, IL, July 2003. Springer.

[7] F. Gonzalez, D. Dasgupta, and L. F. Nino. A randomized real-value negative selection algorithm. In *Proceedings of Second International Conference on Artificial Immune System (ICARIS 2003)*, September 2003.

[8] F. A. Gonzalez and D. Dasgupta. Anomaly detection using real-valued negative selection. *Genetic Programming and Evolvable Machines*, 4:383–403, 2003.

[9] E. Hart. Not all balls are round: An investigation of alternative recognition-region shapes. In *ICARIS*, pages 29–42, 2005.

[10] Z. Ji. A boundary-aware negative selection algorithm. In *Proceedings of IASTED International Conference of Artificial Intelligence and Soft Coomputing (ASC 2005)*, Spain, September 2005.

[11] Z. Ji and D. Dasgupta. Real-valued negative selection algorithm with variable-sized detectors. In *LNCS 3102, Proceedings of GECCO*, pages 287–298, 2004.

[12] Z. Ji and D. Dasgupta. Estimating the detector coverage in a negative selection algorithm. In H.-G. B. et al, editor, *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, volume 1, pages 281–288, Washington DC, USA, 25-29 June 2005. ACM Press.

[13] D.-W. Lee and K.-B. Sim. Negative selection for DNA sequence classification. In *Proceedings of Joint 2nd International Conference on Soft Computing and Intelligent Systems and 5th International Symposium on Advanced Intelligent Systems (SCIS & ISIS 2004)*, Yokohama, Japan, Sept 2004.

[14] R. E. Sanchez-Yanez, E. V. Kurmyshev, and A. Fernandez. One-class texture classifier in the CCR feature space. *Pattern Recognition Letters*, 24:1503–1511, 2003.

[15] J. M. Shapiro, G. B. Lamont, and G. L. Peterson. An evolutionary algorithm to generate hyper-ellipsoid detectors for negative selection. In H.-G. B. et al, editor, *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, volume 1, pages 337–344, Washington DC, USA, 25-29 June 2005. ACM Press.

[16] T. Stibor, P. Mohr, J. Timmis, and C. Eckert. Is negative selection appropriate for anomaly detection? In H.-G. B. et al, editor, *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, volume 1, pages 321–328, Washington DC, USA, 25-29 June 2005. ACM Press.

[17] T. Stibor, J. Timmis, and C. Eckert. A comparative study of real-valued negative selection to statistical anomaly detection techniques. In *ICARIS*, pages 262–275, 2005.

[18] D. M. J. Tax. *One-class classification*. PhD thesis, Technische Universiteit Delft, 2001.

[19] D. W. Taylor and D. W. Corne. An investation of the negative selection algorithm for fault detection in refrigeration system. In *Proceedings of Second International Conference on Artificial Immune System (ICARIS 2003)*, 2003.