Adaptive Positive-Negative Selection Approach

Piotr Honko
Faculty of Computer Science, Bialystok University of Technology,
Wiejska 45A, 15-351 Bialystok, Poland
E-mail: p.honko@pb.edu.pl

Abstract. Positive selection and negative selection are the main artificial immune approaches. They can find application in any task where an automatic identification of incoming cases as self or non-self is needed. Both the approaches build the so-called detectors for protecting self cells, i.e. positive class objects. In a positive selection approach, detectors are able to recognize self objects, whereas the other approach produces detectors that eliminate non-self objects. The difference in construction of detectors causes that the choice of the approach may depend on the characteristic of the data under consideration.

The goal of this paper is to provide a hybrid approach that is able to adapt to a given data set, producing thereby a classifier achieving the best performance. The adaptation is done twofold: (a) in the training phase, each candidate for a detector is automatically determined whether it will be used to build a self or non-self detector; (b) in the inner test phase, the best type of the classifier (i.e. pure positive selection, pure negative selection, positive-negative selection, negative-positive selection) is chosen.

Results of experiments conducted on databases from the UCI Repository show that the combination of positive and negative selection approaches may give a higher classification accuracy.

1. Introduction
The ability of a natural immune system to distinguish between pathogens and non-pathogens has inspired researchers to adapt this mechanism in artificial intelligence [1]. Such a system includes immune cells (detectors), called T cells, which have a component for recognizing self cells and an antigen receptor for locating and eliminating infected pathogens. The system generates a number of T cells, thereby any pathogen to infect the organism is very likely to be detected by at least one of cells. Before becoming a detector of the system a cell is verified to ensure that it is able to distinguish between self and non-self cells. The verification can be done using positive or negative selection. In positive selection T cell that recognizes any own cell is saved, otherwise it is removed. In negative selection, in turn, only cells that do not detect self cells are stored [2].

Both the approaches have successfully been adapted in the field of network security in order to automatically detect anomalies, intrusions, spams, or viruses (see e.g. [3, 4, 5, 6, 7, 8, 9, 10, 11]).

In spite of the fact that both selection approaches are mutually dual, thereby they can be complementary to each other, they have been usually used separately. Only a few studies on combining positive and negative selection approaches can be found.

A formal framework for analyzing different positive and negative detection schemes for data represented by binary strings was given in [12]. The framework enables to find the tradeoff
between positive and negative detection schemes in terms of the number of detectors needed to
correctly distinguish self objects from non-self ones.

An approach proposed in [13] is devoted to binary data representations and uses r-chunk
matching rule. The approach starts with building binary trees corresponding to positive r-chunk
detectors. To obtain a compact representation of the positive detector set, all complete subtrees
of these trees are removed. Finally, a positive tree is replaced with its negative counterpart if
the latter one is more compact.

The goal of this paper is to propose a combination of positive and negative selection
approaches that adapts to the data to be classified. In the training phase, points from real-
valued space (i.e. candidates for detectors) are selected randomly. Based on the distance of a
point to the nearest self object it is determined whether it will be a self or non-self detector.
In the inner test phase, objects are classified using four combinations: pure positive selection,
pure negative selection, positive-negative selection, negative-positive selection. In the first two
cases, only self or non-self detectors are used. In the third (fourth) case, if an object is not
recognized by any self (non-self) detector, then it is classified by its nearest detector. One of
the four combinations that gives the highest accuracy is chosen as the classifier for the data.

Experimental research shows that the positive-negative approach may increase the
classification accuracy and that a better performance is obtained when one of the four
combinations is defined individually for each dataset.

The rest of the paper is organized as follows. Section 2 proposes a combination of positive
and negative selection approaches. Section 3 describes experimental research conducted using
the proposed approach and six datasets taken from the UCI Repository. Concluding remarks
are provided in Section 4.

2. A Combination of Positive and Negative Selection Approaches

This section proposes an algorithm (Algorithm 1) that generates self and non-self detectors at
the same time. An algorithm (Algorithm 2) that classifies new objects according to one of four
versions (positive selection, negative selection, and their two combinations) is also introduced.

Each point (i.e. m-ary vector of real numbers) to be used for building a detector in Algorithm 1
is selected randomly from the data space that is assumed to be normalized to the hypersphere.
The selection is done from either the whole space, i.e. $R_{SN}$ or the space of self objects, i.e.
$R_S$. The choice of the space is done at random. This procedure enables to avoid an undesirable
disproportion between self and non-self detectors sets in terms of their cardinalities. Namely,
selecting points from the whole space might cause that the construction of a self detector is
very unlikely (i.e. a candidate for a detector is usually too far from the nearest self object). On
the other hand, when we limit the domain to the self objects space, the number of constructed
non-self detectors might be very small (i.e. a candidate for a detector is usually too close to the
nearest self object).

The radius of a self detector is constant and is defined by the user. The radius of a non-self
detector is defined individually and equals to the distance of the detector to nearest self objects.

The algorithm stops searching for new detectors if each of $k$ points selected in a row is within
the scope of any detector already created.

Besides the pure positive or negative selection, Algorithm 2 is able to use its combination
for objects that are not included in the sphere of any detector. The class of the object (that is
not inside any detector already generated) is defined based on the type of the detector (self or
non-self) that is the nearest to the object. The distance is measured between the object and
the detector center.
Algorithm 1: GenerateDetectors

Data: $S$ – a set of self cells; $r$ – the self detector radius; $k$ – the maximal number of failed attempts in a row;

Result: $D_S$, $D_{NS}$ – sets of self and non-self detectors, resp.;

begin
  $R_S := \{[\min S_x^i, \max S_x^i] : 1 \leq i \leq m\};$
  /* The set of ranges of variable domains from $S$ */
  $R_{NS} := [0,1]^m$;
  $D_S = \emptyset$; $D_{NS} = \emptyset$; $k' = k$;

  while $k > 0$ do
    $b = $ getRandomValue$([0,1])$;
    /* returns a random value from the set $\{0,1\}$ */
    if $b = 1$ then $R = R_S$ else $R = R_{NS}$;
    $p =$ getRandomPoint$($R$)$;
    /* returns a random point from the space defined by $R$ */
    $b := 1$;
    if isInsideAnyDet$(p, D_S)$ then $b := 0$;
    /* returns true if a point is inside any detector of a give set */
    else if isInsideAnyDet$(p, D_{NS})$ then $b := 0$;
    if $b = 1$ then
      $v =$ distToNearestSelf$(p, S)$;
      /* returns the distance of a point to the nearest self object */
      if $v \leq r$ then $D_S = D_S \cup \{p\}$;
      else $D_{NS} = D_{NS} \cup \{(p,v)\}$;
      $k := k'$;
    end
    else $k := k - 1$;
  end
end

3. Experiments

This section describes experimental research that concerns verification of effectiveness of the proposed positive-negative selection approach.

Six datasets (Table 1) taken from the UCI Repository (http://archive.ics.uci.edu/ml) were used in the experimental research.

| short name | full name                      | no. attr. | no. obj. | class distribution |
|------------|-------------------------------|-----------|----------|-------------------|
| Iris       | Iris                          | 4         | 150      | 50:50:50          |
| Wine       | Wine                          | 13        | 178      | 59:71:48          |
| Letters    | Letter Recognition            | 16        | 20000    | 789:19:211        |
| Shuttle    | Statlog (Shuttle)             | 10        | 43500    | 34108:37:132:6748:2458:6:11 |
| Skin       | Skin Segmentation             | 3         | 245057   | 50 857:194 200    |
| Record     | Record Linkage Comparison     | 12        | 574913   | 2093:572 820      |
Algorithm 2: ClassifyObjects

Data: $O$ – a set of objects; $r$ – the self detector radius; $D_s, D_{ns}$ – sets of self and non-self detectors, resp.; $C$ – the combination ($PS$ – positive selection, $NS$ – negative selection, $PNS$ – positive-negative selection, $NPS$ – negative-positive selection);

Result: $O_s, O_{ns}$ – sets of self and non-self objects, resp.;

begin
  foreach $o \in O$ do
    if $C \neq NS$ then $v = isInsideAnyDet_2(o, D_s)$;
    /* returns -1 if a point is inside any detector of a given set, or the distance to the nearest detector otherwise; */
    if $C \neq PS$ then $v' = isInsideAnyDet_2(o, D_{ns})$;
    switch $C$ do
      case $PS$ do
        if $v = -1$ then $O_s := O_s \cup \{o\}$; else $O_{ns} := O_{ns} \cup \{o\}$;
        end
      case $NS$ do
        if $v' = -1$ then $O_{ns} := O_{ns} \cup \{o\}$; else $O_s := O_s \cup \{o\}$;
        end
      case $PNS$ do
        if $v = -1$ then $O_s := O_s \cup \{o\}$;
        else if $v \leq v'$ then $O_s := O_s \cup \{o\}; else O_{ns} := O_{ns} \cup \{o\}$;
        end
      case $NPS$ do
        if $v = -1$ then $O_{ns} := O_{ns} \cup \{o\}$;
        else if $v \leq v'$ then $O_s := O_s \cup \{o\}; else O_{ns} := O_{ns} \cup \{o\}$;
        end
      end
  end
end

The following testing procedure was applied. First of all, to avoid possible influence of different ranges of attribute domains, all databases were normalized into a unitary hypercube $[0, 1]^n$ where $n$ is the cardinality of the attribute set. One class (the one whose cardinality is underlined in Table 1) of a database was taken as self data, the remaining ones as non-self data. Each data set was divided into five equinumerous sets. Four of them was used in 4-cross-validation (inner test), and the remaining one as an additional testing set (outer test). The non-self objects were only used in the classification phase. To eliminate the influence of randomness each data set was tested five times and the average results were taken. The radius of self objects (0.15 of the maximal possible distance) was defined in a preliminary experimentation. The distance between an object and detector was measured using the Euclidean metric. The quality measure used in the experiments was accuracy.

Table 2 shows the classification accuracy for the training data ($TR$), test data from 4-cross-validation ($TS_1$), and additional test data ($TS_2$). The negative-positive selection combination ($NPS$) had the best performance for five ($TR$), three ($TS_1$), and three ($TS_2$) out of six datasets. However, its results for other datasets are not always close to the best ones, see e.g. Wine ($TS_1$, $TS_2$) or Record ($TR$, $TS_1$, $TS_2$). It leads to the conclusion that the choice of the combination should be done for each database individually based on the training data ($TR$) result or first
Table 2. Accuracy of classification.

| App. | Iris TR | TS1 | TS2 | Wine TR | TS1 | TS2 | Letters TR | TS1 | TS2 |
|------|---------|-----|-----|---------|-----|-----|------------|-----|-----|
| PS   | 99.93   | 99.00 | 99.30 | 97.51   | 96.33 | 95.99 | 67.41 | 67.29 | 67.16 |
| NS   | 99.96   | 93.1 | 94.40 | 99.89   | 52.78 | 52.75 | 93.50 | 86.38 | 86.54 |
| PNS  | 99.93   | 99.60 | 99.90 | 97.96   | 96.33 | 95.99 | 64.82 | 64.68 | 64.64 |
| NPS  | 100     | 93.20 | 94.40 | 99.89   | 52.87 | 52.91 | 93.77 | 86.66 | 86.78 |

| App. | Shuttle TR | TS1 | TS2 | Skin TR | TS1 | TS2 | Record TR | TS1 | TS2 |
|------|------------|-----|-----|---------|-----|-----|-----------|-----|-----|
| PS   | 55.37      | 55.30 | 55.34 | 73.65   | 73.66 | 73.66 | 57.48 | 57.53 | 57.42 |
| NS   | 99.75      | 99.47 | 99.44 | 95.74   | 95.73 | 95.74 | 92.07 | 91.93 | 91.90 |
| PNS  | 56.15      | 56.13 | 56.12 | 72.35   | 72.36 | 72.36 | 70.97 | 71.00 | 70.93 |
| NPS  | 99.75      | 99.47 | 99.44 | 96.51   | 96.50 | 96.50 | 70.97 | 71.00 | 70.93 |

Test data (TS1) result. One can see that the best TR result translates into the best TS1 result for five data sets, whereas the highest TS1 result always implies the highest TS2 result. To investigate this issue more deeply, all the results, i.e. ones that were used to calculate the average results, were taken into account.

Table 3 shows in percentage how often the best result for TR (TS1) is also the best one for TS1 (TS2). The experimentation reveals that the choice of the combination is more accurate if it is done based on the test data (TS1). Namely, for five databases the result is better or the same, and the average performance is clearly higher.

Table 3. Convergence of the best results for the train and first test data and for the train and second test data.

|                | Iris TR | Wine TR | Letters TR | Shuttle TS1 | Skin TR | Record TR | avg    |
|----------------|---------|---------|------------|------------|---------|-----------|--------|
| TR->TS1        | 60.00   | 4.00    | 100        | 100        | 100     | 96.00     | 76.67  |
| TS1->TS2       | 81.33   | 100     | 98.00      | 100        | 100     | 96.00     | 95.89  |

4. Conclusion

This paper has proposed a combination of positive and negative selection approaches devoted to real-valued data. The approach controls the process of detector generation, by determining whether a new detector will be self or non-self. This is also able to find the best combination (i.e. pure positive selection, pure negative selection, positive-negative selection, negative-positive selection) for a given dataset.

Based on the experimental research one can conclude that:

(i) A combination of both the approaches may give a higher classification accuracy than each of them used separately.

(ii) Finding the best combination for a given dataset is more likely when it is done based on not the training data but the test one.

The proposed approach is planned to be applied in future works as a tool for automatic detection of network anomalies.
Acknowledgments
This work was supported by the grant S/WI/1/2018 of the Polish Ministry of Science and Higher Education.

References
[1] Read M, Andrews P S and Timmis J 2012 Handbook of Natural Computing ed Rozenberg G, Bäck T and Kok J N (Berlin, Heidelberg: Springer Berlin Heidelberg) pp 1575–1597
[2] Dasgupta D 1999 Artificial Immune Systems and Their Applications ed Dasgupta D (Berlin, Heidelberg: Springer) pp 3–21
[3] Kim J and Bentley P J 2001 Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) (Morgan Kaufmann) pp 1330–1337
[4] Stibor T, Mohr P, Timmis J and Eckert C 2005 Proceedings of the ACM SIGEVO Genetic and Evolutionary Computation Conference (GECCO-2005) (Washington, D.C.: ACM Press)
[5] Stibor T, Timmis J and Eckert C 2005 Proceedings of the Congress on Evolutionary Computation (CEC-2005) (Edinburgh, UK: IEEE Press)
[6] Zhang F and Qi D 2011 J. Comp. Vir. 7 267–277
[7] Chmielewski A and Wierzchon S T 2012 Computer Information Systems and Industrial Management - 11th IFIP, TC 8 International Conference, CISIM 2012, Venice, Italy, September 26-28, 2012. Proceedings pp 242–253
[8] Mostardinha P, Faria B F, Zúquete A and Vistulo de Abreu F 2012 Artificial Immune Systems: 11th International Conference, ICARIS 2012, Taormina, Italy, August 28-31, 2012. Proceedings ed Coello Coello C A, Greensmith J, Krasnogor N, Liò P, Nicosia G and Pavone M (Berlin, Heidelberg: Springer Berlin Heidelberg) pp 178–190
[9] Chikhi S and Ramdane C 2014 Int. J. Inf. Sec. Priv. 8 1–25
[10] Peng L and Chen Y 2014 2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, CyberC 2014, Shanghai, China, October 13-15, 2014 (IEEE Computer Society) pp 56–59
[11] Idris I and Selamat A 2014 Appl. Soft Comput. 22 11–27
[12] Esponda F, Forrest S and Helman P 2004 IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34 357–373
[13] Nguyen V T, Nguyen X H and Luong C M 2013 The 2013 RIVF International Conference on Computing Communication Technologies - Research, Innovation, and Vision for Future (RIVF) (IEEE Computer Society) pp 6–11