# Generalization Regions in Hamming Negative Selection

Thomas Stibor[1], Jonathan Timmis[2], and Claudia Eckert[1]

[1] Darmstadt University of Technology
Department of Computer Science
Hochschulstr. 10, 64289 Darmstadt, Germany
[2] University of York
Department of Electronics and Department of Computer Science
Heslington, York, United Kingdom

**Abstract** Negative selection is an immune-inspired algorithm which is typically applied to anomaly detection problems. We present an empirical investigation of the generalization capability of the Hamming negative selection, when combined with the r-chunk affinity metric. Our investigations reveal that when using the r-chunk metric, the length $r$ is a crucial parameter and is inextricably linked to the input data being analyzed. Moreover, we propose that input data with different characteristics, i.e. different positional biases, can result in an incorrect generalization effect.

## 1 Introduction

Negative selection was one of the first immune inspired algorithms proposed, and is a commonly used technique in the field of artificial immune systems (AIS). Negative selection is typically applied to anomaly detection problems, which can be considered as a type of pattern classification problem, and is typically employed as a (network) intrusion detection technique.

The goal of (supervised) pattern classification, is to find a functional mapping between input data $X$ to a class label $Y$ so that $Y = f(X)$. The mapping function is the pattern classification algorithm which is trained (or learnt) with a given number of labeled data called *training data*. The aim is to find the mapping function, which gives the smallest possible error in the mapping, i.e. minimize the number of samples where $Y$ is the wrong label ( this is especially important for *test data* not used by the algorithm during the learning phase). In the simplest case there are only two different classes, with the task being to estimate a function $f : \mathbb{R}^N \to \{0, 1\} \ni Y$, using training data pairs generated i.i.d.[1] according to an unknown probability distribution $P(X, Y)$

$$(X_1, Y_1), \ldots, (X_n, Y_n) \in \mathbb{R}^N \times Y, \quad Y \in \{0, 1\}$$

such that $f$ will correctly classify unseen samples $(X, Y)$. If the training data consists *only* of samples from one class, and the test data contains samples from two or more classes, the classification task is called *anomaly detection*.

[1] independently drawn and identically distributed

Once a functional mapping (a model) is found, a fundamental question arises : *does the model predict unseen samples correctly with a high accuracy, or in other words, does the model generalize well ?*. This question is empirically explored for Hamming negative selection algorithm and the associated r-chunk matching rule.

## 2 Artificial Immune System

Artificial immune systems (AIS) [9] is a paradigm inspired by the immune system and are used for solving computational and information processing problems. An AIS can be described, and developed, using a framework which contains the following basic elements:

- A representation for the artificial immune elements.
- A set of functions, which quantifies the interactions of the artificial immune elements.
- A set of algorithms which based on observed immune principles and methods.

### 2.1 Hamming Shape-Space and R-chunk Matching

The notion of *shape-space* was introduced by Perelson and Oster [8] and allows a quantitative affinity description between immune components known as antibodies and antigens. More precisely, a shape-space is a metric space with an associated distance (affinity) function.

The Hamming shape-space $U_l^{\Sigma}$ is built from all elements of length $l$ over a finite alphabet $\Sigma$. A formal description of antigen-antibody interactions not only requires a representation, but also appropriate affinity functions.

The r-chunk matching rule is an affinity function for the Hamming shape-space and can be defined as follows :
Given a shape-space $U_l^{\Sigma}$, which contains all elements of length $l$ over an alphabet $\Sigma$ and a shape-space $D_r^{\Sigma}$, where $r \leq l$.

**Definition 1.** An element $e \in U_l^{\Sigma}$ with $e = e_1 e_2 \dots e_l$ and detector $d \in \mathbb{N} \times D_r^{\Sigma}$ with $d = (p, d_1 d_2 \dots d_r)$, for $r \leq l,\ p \leq l - r + 1$ *match with r-chunk rule* if $e_i = d_i$ *for* $i = p, \dots, p + r - 1$.

Informally, element $e$ and detector $d$ match if a position $p$ exists, where all characters of $e$ and $d$ are identical over a sequence length $r$.

## 3 Hamming Negative Selection

Forrest et al. [6] proposed a (generic) negative selection algorithm for detecting changes in data streams. Given a shape-space $U = S_{seen} \cup S_{unseen} \cup N$ which is partitioned into training data $S_{seen}$ and testing data $(S_{unseen} \cup N)$.

The basic idea is to generate a number of detectors for the complementary space $U \setminus S_{seen}$ and then to apply these detectors to classify new (unseen) data as self (no data manipulation) or non-self (data manipulation).

**Algorithm 1**: Generic Negative Selection Algorithm

**input** : $S_{seen}$ = set of self seen elements
**output**: $D$ = set of generated detectors
**begin**
1. Define self as a set $S_{seen}$ of elements in shape-space $U$
2. Generate a set $D$ of detectors, such that each fails to match any element in $S_{seen}$
3. Monitor (seen and unseen) data $\delta \subseteq U$ by continually matching the detectors in D against $\delta$.

**end**

The generic negative selection algorithm can be used with arbitrary shape-spaces and affinity functions. In this paper, we focus on Hamming negative selection, i.e. the negative selection algorithm which operates on Hamming shape-space and employs the r-chunk matching rule. More specifically, we explore the performance of how well Hamming negative selection can generalize when using the r-chunk affinity metric.

### 3.1 Holes as Generalization Regions

The r-chunk matching rule creates undetectable elements (termed holes). Holes are elements of $N$, or self elements, not seen during the training phase ($S_{unseen}$). For these elements, no detectors can be generated and therefore they cannot be recognized and classified as non-self elements. The term holes is not an accurate expression, as holes are *necessary* to generalize beyond the training set. A detector set which generalizes well, ensures that seen and unseen self elements are *not* recognized by any detector, whereas all other elements are recognized by detectors and classified as non-self. Hence, holes must represent unseen self elements; or in other words, holes must represent *generalization regions* in the shape-space $U_l^{\Sigma}$.

## 4 Generalization Regions Experiments

In [1] and [5] results are presented which show the coherence between the number of holes and the number of generable detectors under the assumption that the training set $S_{seen}$ is randomly drawn from $U_l^{\Sigma}$. More specifically, the coherence between the element length $l$, r-chunk length $r$, number of self elements $|S_{seen}|$ and the number of holes and generable detectors is shown [5]. However, these results provide no information *where* holes occur. Holes must occur in regions where most self elements are concentrated. Recall, as holes are not detectable by any detector, holes must represent unseen self elements,

or in other words, holes must represent *generalization regions*. In order to study the number and the occurrence of holes which are dependent on the r-chunk length, we have created a number of artificial self data sets (illustrated in figures 3,4,5; these can be found in the Appendix). The first self data set contains 1000 random points $p \in [0,1]^2$ which lie within a single ellipsoid cluster with centre $(0.5, 0.5)$, height 0.4 and width 0.2. Each point $p = (x, y)$ is mapped to a binary element $e_0, e_1, \ldots, e_{15}$, where the first 8 bits encode the integer x-value $\lceil 255 \cdot x + 0.5 \rceil$ and the last 8 bits the integer y-value $\lceil 255 \cdot y + 0.5 \rceil$, i.e. $[0,1]^2 \rightarrow (i_x, i_y) \in [1, \ldots, 256 \times 1, \ldots, 256] \rightarrow (b_x, b_y) \in U_8^{\{0,1\}} \times U_8^{\{0,1\}}$. This mapping was proposed in [4]. The second self data set contains 1000 random generated self elements which are lying within a rectangle. The third data set contains 1000 Gaussian ($\mu = 0.5, \sigma = 0.1$) generated points. It is not possible to generate all self elements [2] within the self region (ellipse, rectangle, Gaussian), therefore we explore where holes occur. Ideally, as stated before, holes should occur within the self region.

In figures 3,4,5, one can see that for $r < 8$, holes occur in regions which lie outside of the self region — or put another way, only a limited number of holes exist at all (see e.g. Fig. 3). Furthermore, it was observed that for $8 \leq r \leq 11$, holes occur in the generated self region (as they should), and a detector specificity of $r = 10$ provides the best generalization results. However, for $r > 11$ the detector specificity is too large, and as a result, the self region is covered by the detectors rather than by the holes. It is worth noting that a certain detector specificity *must* be reached to obtain holes within the generated self region.

By calculating the *entropy* [7] of the binary representation of $S$ for different r-chunk length $r$, it is possible to obtain an explanation for *why* a detector specificity $r \geq 8$ is required to obtain holes close or within the self region. Entropy is defined as

$$H(X) = \sum_{x \in \mathcal{A}_X} P(x) \log_2 \left( \frac{1}{P(x)} \right) \quad \text{[bits]} \tag{1}$$

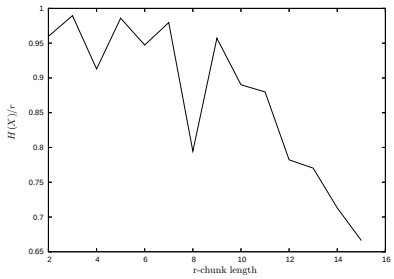
where the outcome $x$ is the value of a random variable which takes one of the possible values $\mathcal{A}_X = \{a_1, a_2, \ldots, a_n\}$, having probabilities $\{p_1, p_2, \ldots, p_n\}$ with $P(x = a_i) = p_i$. Roughly speaking, entropy is a measurement of randomness (uncertainty) in a sequence of outcomes. The entropy is maximal[3], when all outcomes have an equal probability.

In this entropy experiment, all 1000 generated self points are concatenated to one large bit string $L_S$ of length $16 \cdot 10^3$. The bit string $L_S$ is divided into $\lfloor 16 \cdot 10^3 / r \rfloor$ substrings (the outcomes $\mathcal{A}_X$). The entropy for $r = \{2, 3, \ldots, 15\}$ for each data set is calculated and the ratio $H(X)/r$ to the maximum possible entropy is calculated, and depicted in a graph (see Fig. 1). The maximum

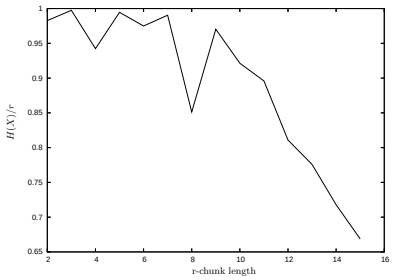
[2] Simulating $S_{seen}$

[3] largest uncertainty

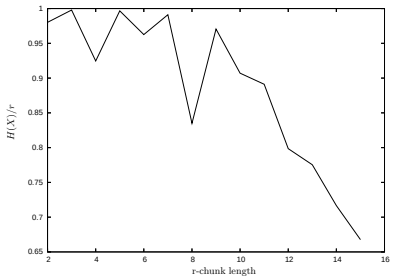
possible entropy for r-chunk length $r$ is $r$ bits (each $r$ bit sequence is equally likely). In figure 1, the coherence between $H(X)/r$ and $r$ for each data set is




(a) Entropy ratio of ellipse self set




(b) Entropy ratio of rectangle self set




(c) Entropy ratio of Gaussian self set

**Figure1.** Coherence between entropy ratio $H(X)/r$ of self set $S$ and r-chunk lengths $r = \{2, 3, \ldots, 15\}$.

presented. One can see that when the r-chunk length $r$ is increased towards $l$, the entropy decreases as the bit strings of length $r$ become more specific, rather than random. Of most interest is the value at $r = 8$. For this value, the entropy ratio $H(X)/r$ results in a spiky jump, when compared to the neighbor values $r = 7$ and $r = 9$. Through exploring the mapping function $[0, 1]^2 \rightarrow (i_x, i_y) \in [1, \ldots, 256 \times 1, \ldots, 256] \rightarrow (b_x, b_y) \in U_8^{\{0,1\}} \times U_8^{\{0,1\}}$, one can see that the bit string of length 16 is semantically composed of two bit strings of length 8 which represents the $(x, y)$ coordinates. A r-chunk length $r < 8$ destroys the mapping information — the semantic representation of the $(x, y)$ coordinates — and therefore the bit strings of length $r$ have a random character rather than a semantic representation of the $(x, y)$ coordinates. As a consequence, holes occur in regions, where actually *no* self regions should be (see Fig. 3(a)-3(f), 4(a)-4(f), 5(a)-5(f)).

It has been noted that a similar statement[4] was mentioned by Freitas and Timmis [2] with regard to the r-contiguous matching rule: "It is important to

[4] without empirical results

understand that r-contiguous bits rule have a *positional bias*". Our entropy experiments support and empirically confirm this statement.

Furthermore, the observations implicate an additional "positional bias" problem. When elements of different lengths are concatenated to a data chunk, and the r-chunk length is too large (too specific) for some small length elements and also too small (too generic) for some large length elements, then holes occur in the wrong regions (see Fig. 2). Figure 2 shows elements $e_1, e_2$

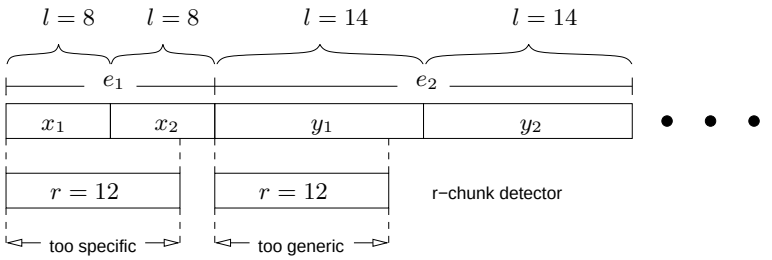



**Figure2.** Concatenating elements $e_1, e_2$ of different length, can result in wrong generalization, as no suitable r-chunk detector length exists which capture the representations of $e_1$ and $e_2$.

— which represent coordinates $(x_1, x_2)$ and $(y_1, y_2)$ — of different lengths and a r-chunk detector of length $r = 12$. This r-chunk length is too specific for length $l = 16$ of $e_1$, but likewise too generic for length $l = 28$ of $e_2$. As a consequence, no suitable r-chunk detector length for this example in figure 2 exists. We emphasize this "positional bias" problem here, as in many Hamming negative selection approaches when applied as a network intrusion detection technique, elements[5] of different lengths are concatenated: the implications are clear — for an overview of this approach see [3].

## 5 Conclusion

Hamming negative selection is an immune-inspired technique, which can be applied to anomaly detection problems. In this paper we have empirically explored the generalization capability of the Hamming negative selection when using the r-chunk length $r$. The generalization ability in Hamming negative selection is caused by undetectable elements termed "holes". Holes are undetectable elements which must represent unseen self data. Moreover, holes must occur in regions where most self data is concentrated. Our results have revealed that the r-chunk length must be of a certain length to achieve a correct generalization. The r-chunk length can not be chosen arbitrary, as much depends on the semantic representation of the input data. An r-chunk

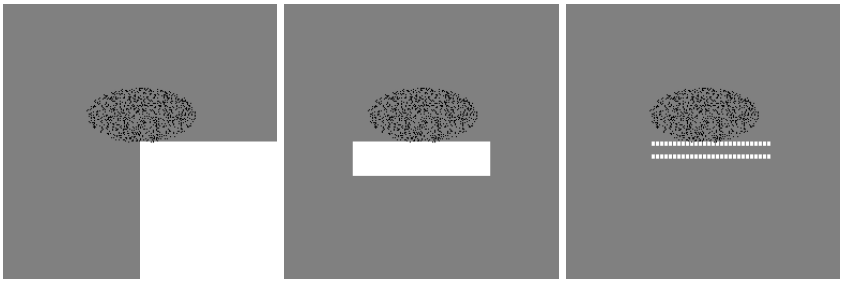
[5] IP-Addresses, Ports, etc.

length which does not properly capture the semantic representation of the input data, will result in an incorrect generalization. Furthermore, we conclude that input data which is composed of elements of different lengths, can itself result in an incorrect generalization, as a suitable r-chunk length does not exist for each different length.

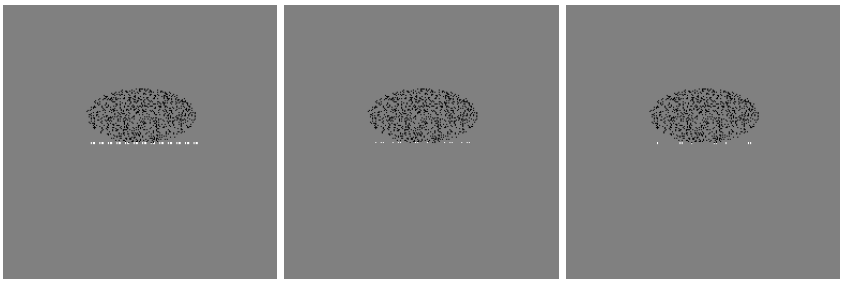
## References


1. Esponda F., Forrest S., Helman P. A formal framework for positive and negative detection schemes. *IEEE Transactions on Systems, Man and Cybernetics Part B: Cybernetics*, 34(1):357–373, 2004.
2. Freitas A., Timmis J. Revisiting the Foundations of Artificial Immune Systems: A Problem Oriented Perspective. In *Proceedings of the 2nd International Conference on Artificial Immune Systems (ICARIS)*, volume 2787 of *Lecture Notes in Computer Science*, pages 229–241. Springer, September 2003.
3. Aickelin U., Greensmith J., Twycross J.", Immune System Approaches to Intrusion Detection – A Review. In *Proceedings of the 3nd International Conference on Artificial Immune Systems (ICARIS)*, volume 3239 of *Lecture Notes in Computer Science*, pages 316–329. Springer, 2004.
4. González F., Dasgupta D., Gomez G. The effect of binary matching rules in negative selection. In *Genetic and Evolutionary Computation (GECCO)*, volume 2723 of *Lecture Notes in Computer Science*, pages 195–206, Chicago, 12-16 July 2003. Springer-Verlag.
5. Stibor T., Timmis J., Eckert C. On the Appropriateness of Negative Selection defined over Hamming Shape-Space as a Network Intrusion Detection System. *Congress On Evolutionary Computation (CEC)*, pages 995–1002, IEEE Press, 2005.
6. Forrest S., Perelson A. S., Allen L., Cherukuri R. Self-Nonself Discrimination in a Computer. In *Proc. of the 1994 IEEE Symposium on Research in Security and Privacy*. IEEE Computer Society Press, 1994.
7. MacKay D. J. C. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
8. Perelson A. S., Oster G. Theoretical studies of clonal selection: minimal antibody repertoire size and reliability of self-nonself discrimination. In *J. Theor. Biol.*, volume 81, pages 645–670, 1979.
9. de Castro L. N., Timmis J.. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer-Verlag, 2002.

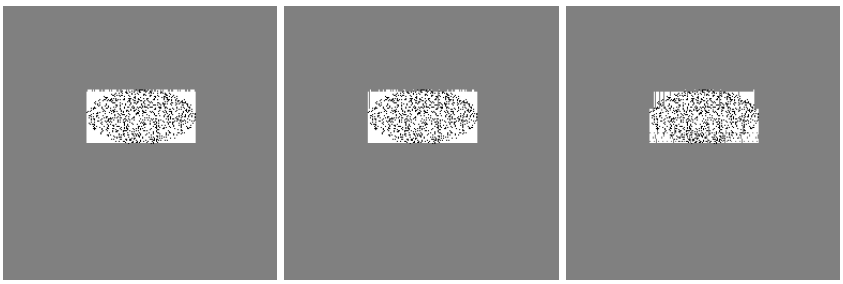

## 6 Appendix



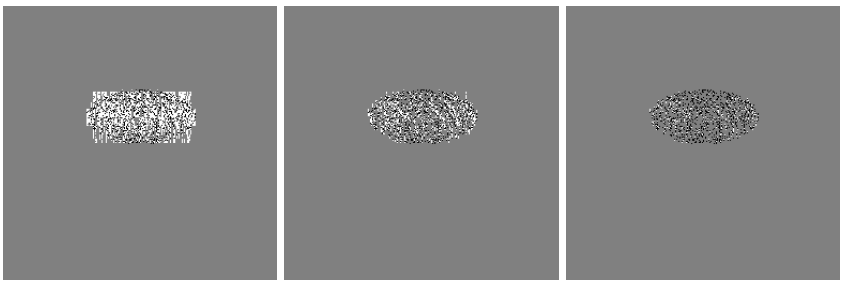
(a) $r = 2$ (b) $r = 3$ (c) $r = 4$



(d) $r = 5$ (e) $r = 6$ (f) $r = 7$

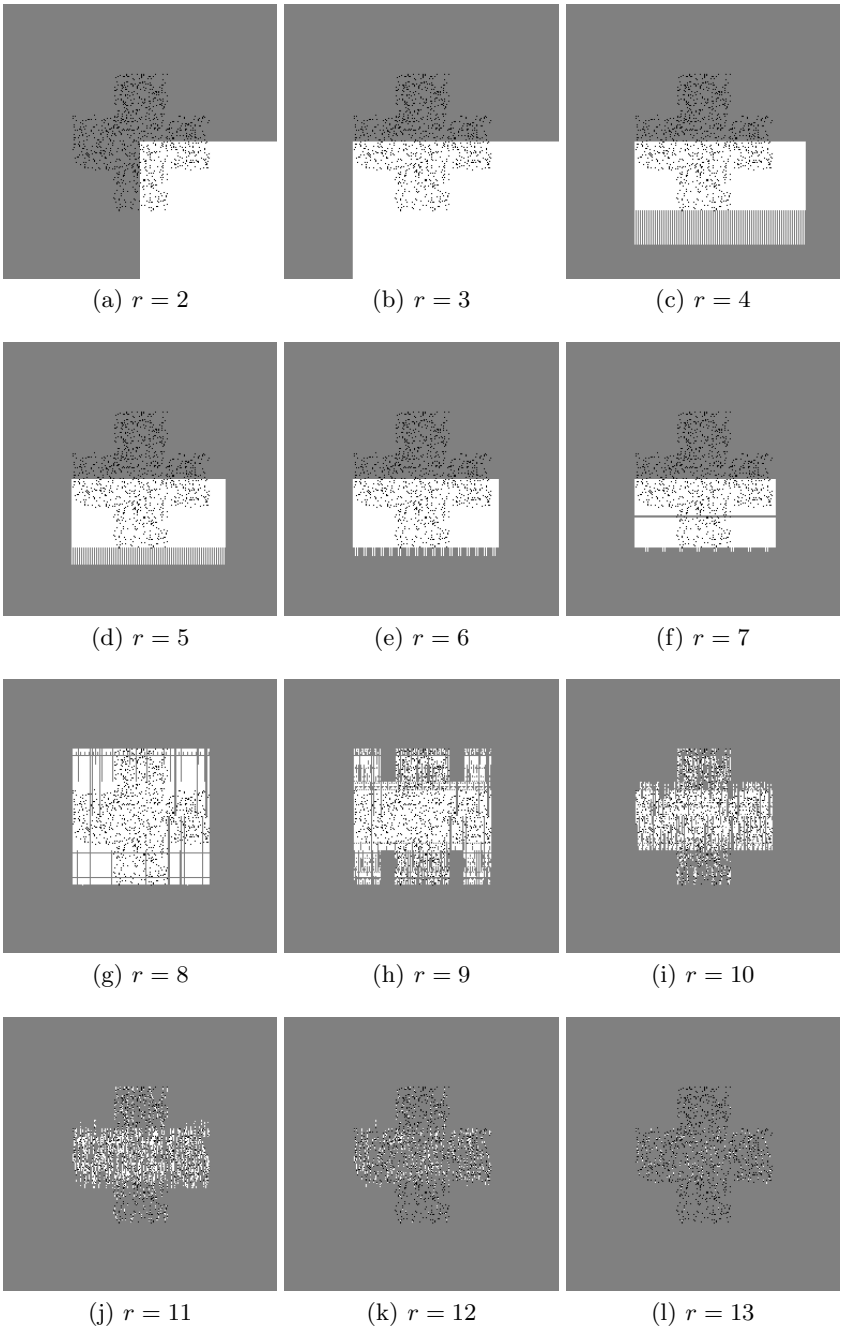


(g) $r = 8$ (h) $r = 9$ (i) $r = 10$



(j) $r = 11$ (k) $r = 12$ (l) $r = 13$

**Figure3.** 1000 random (self) points distributed inside an ellipse with center $(0.5, 0.5)$, height 0.4 and width 0.2. The grey shaded area is covered by the generated r-chunk detectors, the white area are holes. The black points are self elements.



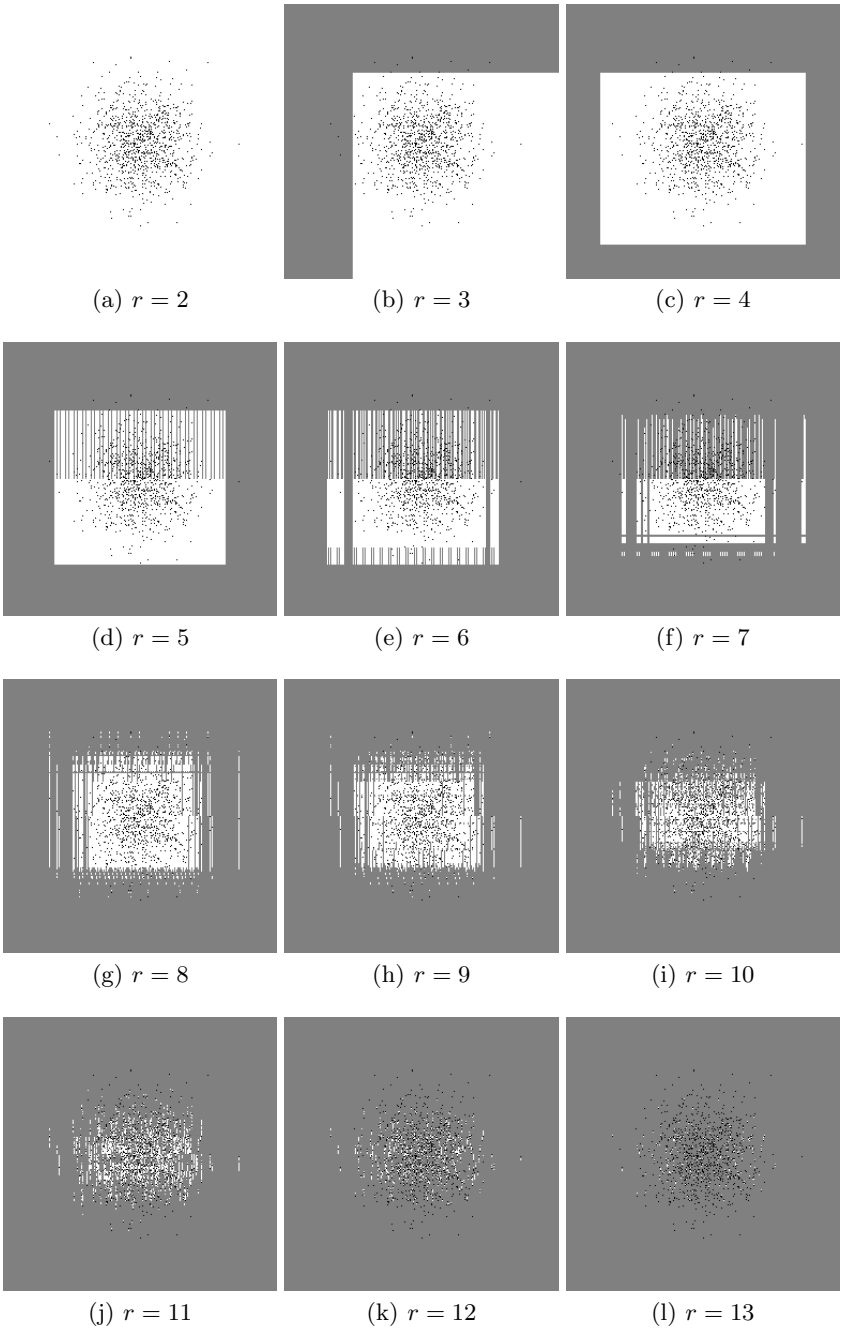
(a) $r = 2$ (b) $r = 3$ (c) $r = 4$

(d) $r = 5$ (e) $r = 6$ (f) $r = 7$

(g) $r = 8$ (h) $r = 9$ (i) $r = 10$

(j) $r = 11$ (k) $r = 12$ (l) $r = 13$

**Figure4.** 1000 random (self) points distributed inside two rectangles with $x, y$ coordinates $(0.4, 0.25)$, height 0.2, width 0.5 and coordinates $(0.25, 0.4)$, height 0.5, width 0.2. The grey shaded area is covered by the generated r-chunk detectors, the white area are holes. The black points are self elements.



(a) $r = 2$ (b) $r = 3$ (c) $r = 4$

(d) $r = 5$ (e) $r = 6$ (f) $r = 7$

(g) $r = 8$ (h) $r = 9$ (i) $r = 10$

(j) $r = 11$ (k) $r = 12$ (l) $r = 13$

**Figure5.** 1000 random (self) points generated by a Gaussian distribution with mean $\mu = 0.5$ and variance $\sigma = 0.1$. The grey shaded area is covered by the generated r-chunk detectors, the white area are holes. The black points are self elements.