# Improving SOSDM: Inspirations from the Danger Theory

Emma Hart and Peter Ross

Napier University, Scotland, UK
{e.hart,p.ross}@napier.ac.uk

**Abstract.** This paper presents improvements to SOSDM based on ideas gleaned from the Danger Theory of immunology. In the new model, antibodies emit a signal describing their current level of contentment – monitoring the total level of contentment in the system provides a mechanism for determining when an immune response should occur, i.e. when new antibodies should be produced. It also provides a method of detecting catastrophic changes in the environment, i.e. significant changes in input data, and thus provides a means of removing antibodies. The new system, *d*SOSDM, is shown to be more robust and better able to deal with dynamically changing databases than SOSDM.

## 1 Introduction

A growing body of literature has shown that the Artificial Immune System (AIS) paradigm is a viable metaphor for performing data-clustering, for example [4,5,8]. In particular, the most recent applications capitalize on the dynamic aspect of the immune metaphor to produce systems capable of clustering moving data. Such systems must necessarily be self-regulating, however experience has shown that they are often difficult to control. This paper introduces an extension to an existing system that allows it to self-regulate its own size, in response to a dynamically changing environment. The extensions are rooted in the relatively contentious Danger Theory.

Danger theory is a theory that has been proposed by Matzinger [7] as an alternative viewpoint to the classical self/non-self discrimination theory popular with a large faction of immunologists. The theory claims to counter certain objections to the classical viewpoint and takes the stance that the immune system does not discriminate between self and non-self, but 'some self from some non-self'. It is proposed that this occurs by the immune system responding to *danger* and not *non-self*. The theory is controversial – this paper does not defend or dispute its existence, nor does it attempt to *exactly* model any of the proposed biological mechanisms implicit in the theory. It is simply of interest because at a high level, some of its concepts can be adapted to improve the engineering of artificial systems.

The interested reader should refer to [7,2] for a detailed description of how she proposes the immune system utilises the notion of danger. The theory invokes many different types of immunological cell with a complex sequence of

J. Timmis et al. (Eds.): ICARIS 2003, LNCS 2787, pp. 194–203, 2003.

signals passing between them. However, a detailed analysis is unnecessary here in to order to understand how the theory can inspire the implementation of artificial systems. An extremely simplified explanation is as follows: if cells become damaged (for example due to attack by invading bacteria) then those cells emit a distress signal as they undergo lytic cell death. Antigen-presenting cells in the region of the cells emitting these signals capture any antigens in the neighbourhood, and then travel to the local lymph node where they present the captured antigen to lymphocytes. The immune system then responds, causing the stimulated antibodies within the danger zone to undergo clonal expansion. The nature of what exactly constitutes a danger signal is unclear and is the subject of much immunological research. It has been suggested that heat-shock proteins might be released as a cell dies, or that the sudden lack of synaptic contact with an antigen presenting cell might signal danger. However, in order to make use of the concept in artificial systems, it is sufficient to accept that such a signal can exist.

### 1.1 Danger Theory and AIS

The literature contains very few direct examples of the danger theory being utilised in Artificial Immune Systems. [3] implements a basic version of the idea in a computer immune system by using the signals generated by dying computer processes to indicate danger, and thus detect when a fault has occurred, though this is clearly only one of many possible danger signals that could be emitted by a computer immune system. [1] present a conceptual discussion on the use of danger theory in artificial systems, and attempt to ground the discussion by suggesting relevant application areas to which the theory could usefully be applied. However, their ideas tend to be directed towards systems which essentially implement some form of negative selection (for example to perform anomaly detection) and consider how danger theory can address weaknesses in such systems. In fact, they state that "*it is not obvious how the Danger Theory could be of use to data-mining problems . . . because the notions of self and non-self are not used*". However, in this paper we argue that the concepts embodied in the Danger Theory can be applied to models of the immune system in which there is no notion of self and non-self, so long as the AIS practitioner does not intend to faithfully replicate mechanisms observed in the biological system. This approach is of course common in probably all biologically inspired paradigms – neural networks cannot hope to faithfully model brains, nor ant-colony systems real colonies of ants, yet by taking inspiration from natural systems, successful artificial ones have been produced.

We suggest that two basic ideas can be gleaned from the Danger Theory which can be useful to many kinds of artificial immune system implementations:

1. cells can emit distress signals, which reflect their relationship with the current state of their immediate environment
2. cells can die an unnatural unprogrammed death (*lytic* cell death as opposed to *apoptosis*, natural cell death) if severely stressed

The new version of SOSDM implements both these ideas. First however, a brief description of the limitations of the current SOSDM and other AIS models is given.

## 2 Limitations of SOSDM and Other AIS Models

SOSDM was first presented in [5,6] as system for clustering dynamically changing data-sets. It is a self-organising system based on analogies with immunology and sparse-distributed memories – an overview of the algorithm is given in figure 1. This paper concerns step (6) of the algorithm – the original model dynamically adjusted the number of antibodies in its system through the use of an antibody-addition mechanism and an antibody-deletion mechanism. The former mechanism added antibodies to the system whenever *stagnation* of the system was detected – this was defined to occur whenever the number *and* type of antibody in the system had remained static over some pre-defined and fixed number of iterations. The antibody-deletion mechanism deleted antibodies from the system whenever their stimulation-level (relative to other antibodies in the system) was less than another pre-defined threshold. These methods have some unappealing aspects:

- Stagnation could occur simply because the system had discovered the best set of antibodies to cluster the data, and not because it had become 'stuck' in a local optima
- If unnecessary antibodies were added due to incorrectly detecting stagnation, the delete mechanism would have to be invoked several iterations later to remove such antibodies
- Determining a suitable threshold below which to delete antibodies was difficult – if not chosen carefully, severe oscillations occurred in which antibodies would be deleted then re-added a few iterations later, etc.

The literature shows that controlling the growth of an artificial immune system can often be problematic. In many proposed immune-network algorithms, for example [9], new antibodies are added to systems via a cloning mechanism, which needs to be carefully checked in order to produce networks of bounded size. Thus, the problem for such systems is not how to *add* antibodies but how to *delete* them from the system in order to prevent exponential growth of the network. Timmis [9] tackled this by introducing a resource allocation mechanism, in which a network had a finite number of arbitrary resources which it had to allocate to its members. Antibodies that do not obtain sufficient resource are removed. This work was further improved on by Neal [8] who simplified the resource allocation mechanism in an algorithm named SSAIS such that resources were not allocated centrally (which is contrary to the distributed nature of the biological immune system) but were dealt with locally by each node in the network. In the next section, we propose a mechanism by which both antibody addition and deletion can be controlled by an algorithm inspired from danger theory.

1. begin with a fixed number of antibodies $A$, randomly initialised.
2. present a subset $s$ of the data-set visible at time $t$ to the SOSDM
3. distribute the data in the set $s$ to *each antibody* in the SOSDM, with a *strength* proportional to the *relative affinity* of the antibody for the data
4. compute the accumulated error at each antibody
5. update all antibody definitions depending on the total accumulated error at each antibody
6. add or delete antibodies if necessary
7. go back to step 2

**Fig. 1.** The SOSDM algorithm

## 3 Adding Danger to SOSDM

A key aspect of the SOSDM algorithm is that antigens become bound to an antibody if the affinity of the antibody for the antigen *relative* to all other antibodies in the system is greater than some threshold. Thus, it follows that all antigens bind to at least one antibody, even though the affinity between any given antigen-antibody pair may be very weak.

Inspired by the notions embodied in the danger-theory of immunology, we propose that an antibody accumulates a measure of the affinity between itself and all antigens it binds too. This quantity can be considered to be a measure of the current level of 'contentment' that the antibody is currently experiencing. The antibody transmits its current contentment level to the system – if the total level of contentment within the system consistently remains below a fixed threshold, then the system responds by producing a new antibody. Using the idea of contentment, we also model both natural and lytic cell death. If the system suddenly experiences a significant change in its overall level of contentment, cells undergo lytic death in response to the sudden 'shock'. This could happen for example if the data which the system is exposed to suddenly changes radically, for example if entire clusters disappear or clusters 'move'. If a cell has zero contentment, this indicates that it is not bound to any antigens and it undergoes natural cell death (i.e. the process is independent of the danger theory). The model is explained in detail below.

## 4 The New Model

Assume that an SOSDM immune system $I$ contains $N$ antibodies, $a_i$, each consisting of a binary string of length $l$, and $X$ antigens (data-items), $x_i$, also of length $l$. Applying the SOSDM algorithm results in $n_{a_i}$ antigens binding to an antibody $a_i$.

First, we measure the average *distress* of each antibody bit, $d(a_{ij})$ which is simply a cumulative measure of the error between the actual value of the

antibody bit $j$ and the value of corresponding antigen bit $j$ of each antigen it binds.

$$d(a_{ij}) = \sum_{k=0}^{k=n_{a_i}} (|a_{ij} - x_{kj}|)/n_{a_i} \tag{1}$$

Then, once all data-items have been stored in the immune system, for each antibody $a_i$, the total number of bits whose average distress is greater than a threshold $T_b$ (the distress-threshold) is calculated – $D_{a_i}$

$$D_{a_i} = \sum_{j=1}^{j=l} \begin{cases} 1 & if \quad d(a_{ij}) > T_b \\ 0 & otherwise \end{cases} \tag{2}$$

The *contentment* of an antibody, $c_{a_i}$ is then defined as shown below, and reflects the percentage of contented bits in the antibody.

$$c_{a_i} = 1 - (D_{a_i}/l) \tag{3}$$

Thus, the average contentment of the whole immune system, $c_I$ can also be calculated, simply as

$$c_I = \sum_{i=1}^{i=N} c_{a_i}/N \tag{4}$$

$c_I$ is central to the new addition mechanism. The idea is to monitor the value of system contentment $c_I$ and use it to signal to the immune system that it needs to produce new antibodies. In order to ignore the effects of any instability in the system from one iteration to the next due to random effects, we monitor the moving-average of $c_I$ – $m(c_I)$ – and compare it to a threshold denoted $T_s$, the system-contentment threshold. If $m(c_I)$ is below this threshold, then a new antibody is added to the system, in order that it can try and increase its contentment level. The algorithm is shown in figure 2. Antibodies with contentment level equal to zero, i.e. that do not recognise any antigens, undergo the equivalent of natural cell death and are removed from the system. The algorithm allows the system to be in one of two states: *immature* or *matured*. Whenever a new antibody is added to the system, it enters a maturational phase to allow it to adjust to the new set of antibodies. This phase endures for a fixed number of iterations before the system is considered mature.

## 5 Experiments Using Static Data-Sets

An initial set of experiments was performed with the original SOSDM in order to analyse how the contentment parameter varied and hence determine suitable settings and sensitivities for the bit-threshold $T_b$ and the system-threshold $T_s$. Typical results are reported for an experiment on a data-set known as the quarter-set, containing 200 binary data-items, each of length 64 bits, in which the data is equally divided into 4 clusters (see [5,6] for a detailed description of

- At each iteration
  - Calculate moving-average of system contentment $c_I$ over previous $m$ iterations, $m(c_I)$
    - If $m(c_I) < T_s$ and the system is *not* immature, add a new antibody to the system
    - Set system-phase to *immature* if an new antibody added
  - If the system is *immature* phase
    - Update number of iterations *immature* counter
    - If immature counter equals maximum iterations allows in maturation, reset system phase to *matured*

**Fig. 2.** The new mechanism for antibody addition

this data-set). The original SOSDM algorithm was run in each from a starting point in which it contained two antibodies, and new antibodies were added at intervals of 20 iterations whenever stagnation was detected, up to a maximum of 4, as described in section 2. The deletion mechanism in SOSDM was turned off so that antibodies could only ever be added to the system for the purpose of these experiments. The moving average of the system contentment was calculated over 5 iterations of the algorithm.

### 5.1 Parameter Settings

The first experiments investigated the variation in the contentment parameter as the original SOSDM system was running. Figure 3(a) shows the variation in contentment of both the whole system and individual antibodies as the threshold $T_b$ is varied. From these experiments, we concluded that a sensible value for the threshold parameter was 0.35 – if the parameter is set below this, it takes the system too long to achieve a stable level of contentment, and if it is much higher, the average contentment is always too high to make it a meaningful indicator of system performance. Figure 3(b) shows the variation in individual antibody contentment at the fixed threshold of 0.35. Further experiments showed that this choice of value was robust across the two other data-sets described in section 5.2.

Further experiments were performed using the quarter data-set, this time using *dSOSDM* with $T_b$ set to the chosen value of 0.35. Again, no deletion mechanism was included, to see if the addition mechanism alone could control the size of the systems produced. As above, the systems were initialised with 2 antibodies, however this time no limit was placed on the number of antibodies the mechanism could add. Figure 4 shows how the setting of the system-threshold parameter $T_s$ affects the number of antibodies present in the final system after *d*SOSDM had been applied for 200 iterations. The results are the average of 10 experiments for each value of $T_s$.

Figure 4 suggests that $0.5 < T_s < 0.6$ is a suitable value – the number of antibodies produced is then comparable to the known number of clusters (although

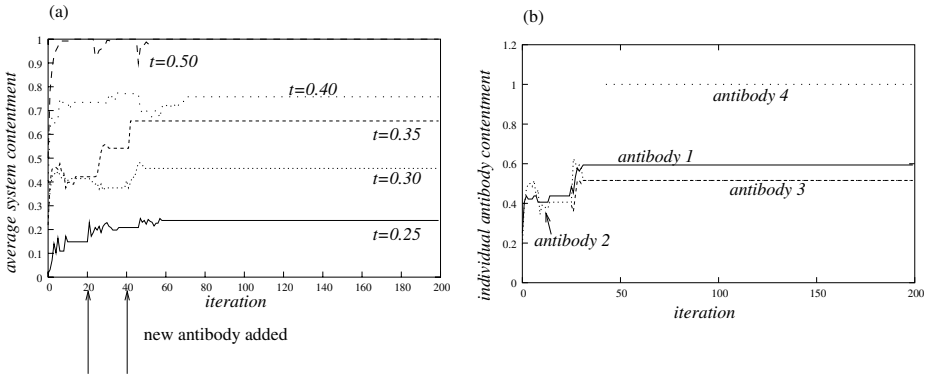


**Fig. 3.** (a) Variation in system contentment as the distress-threshold $T_b$ is varied (b) Variation in individual contentment when $t = 0.35$

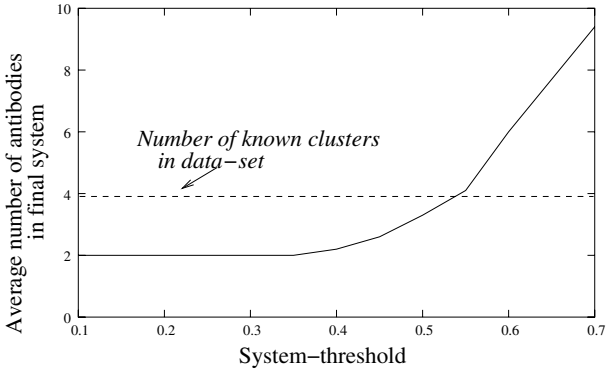


**Fig. 4.** Average number of antibodies obtained using *d*SOSDM as parameter $T_s$, the system-threshold, is varied

of course, due to the random method of generating the clusters, there may be more or less than exactly 4 clusters). Increasing the contentment threshold, i.e. requiring the average match between an antibody and its bound antigens to be higher, forces the system to find sub-clusters and hence add more antibodies. This parameter offers a simple way of tuning the specificity of the system – if specific trends wish to be identified in the data, the value can be increased, whereas a lower value identifies more general trends.

### 5.2 Results: Growing the AIS Using *d*SOSDM

Experiments were repeated using *half* and *eighth* data-sets, [5]. These binary data-sets are again artificially generated to contain two and eight known clusters respectively (although again, each cluster could contain sub-clusters due to the random generation process). *d*SOSDM was applied to both data-sets, and initialised with 2 random antibodies in each case. $T_s$ was set to 0.55 and $T_b$ to

**Table 1.** Comparison of the average number of antibodies used by SOSDM and *d*SOSDM to cluster data-sets, and average accuracy of recall ($RA$) of data (standard deviations in brackets)

| Data | *Expected No. clusters* | # Clusters SOSDM | # Clusters *d*SOSDM | $RA$ of string generalist | $RA$ *d*SOSDM |
|---|---|---|---|---|---|
| half | 2 | 2.29 | 2.30 (0.58) | 48 | 49.38 (0.49) |
| quarter | 4 | 6.75 | 3.99 (1.41) | 40 | 43.38 (0.76) |
| eighth | 8 | 10.06 | 10.18 (0.96) | 36 | 42.85 (0.29) |

0.35, the values found from experiments with the quarter data-set. The length of the maturational period was set to 10 iterations. Again, *d*SOSDM did not use any deletion mechanism and no limit was placed on the number of antibodies that could be added to the system. Table 1 shows the average number of antibodies created by *d*SOSDM in each case (averaged over 100 runs of 200 iterations each). The table compares these results to those found using SOSDM on the same data-sets, originally reported in [5]. Note that SOSDM required the use of both addition and deletion mechanisms in order to control its size and stability, but the simpler *d*SOSDM is either comparable or better in performance in each case. The table also shows the average recall accuracy of each item in the dataset which is indicative of the accuracy of clustering (see [5] for further explanation).

## 6 Clustering Dynamically Changing Data

The original motivation behind SOSDM was to produce a system that could dynamically cluster data, and thus react to changes in the environment. If new clusters appear in the data-set, or clusters suddenly disappear, then the system should be able to detect these changes and react accordingly, by adding new antibodies or removing existing ones. Using the danger theory analogy, such extreme changes in environment would cause severe stress or trauma to cells which were previously content, causing them to undergo lytic cell death. Thus, we model this in *d*SOSDM by monitoring the average change in cell contentment, $\Delta c_I$ between iteration $(t)$ and iteration $(t+1)$ for the system, according to equation 5.

$$\Delta c_I = \frac{1}{N} \sum_{i=0}^{i=N} |c_{a_i}(t+1) - c_{a_i}(t)| \tag{5}$$

Note the use of the absolute value of the change in equation 5 as environmental changes could have a positive or negative effect on contentment, however, even sudden positive changes are traumatic to the cells. If $\Delta c_I$ changes by more than some fixed parameter $z$ then the environment that the system has been exposed is considered to have changed radically and *all* cells in the system undergo lytic death. The consequence of this is that the system is effectively re-initialised and therefore produces new, random, antibodies (the minimum number of antibodies required to be present in the system is always fixed at 2). The only caveat

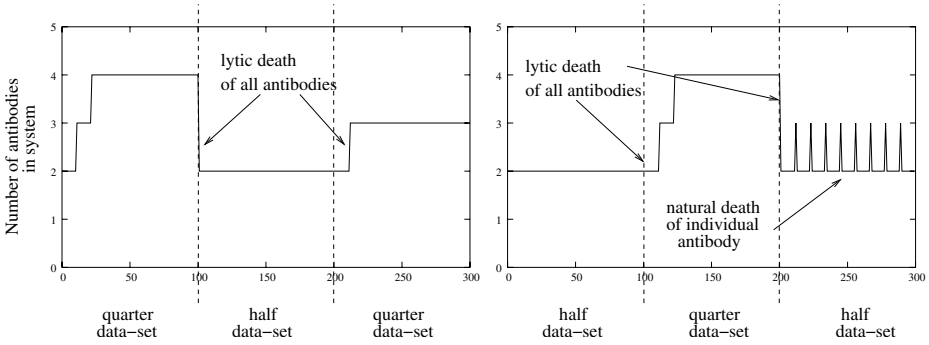


**Fig. 5.** Variation in numberof antibodies with dynamically changing data

to this is that if the system is still in a maturational phase, then catastrophic changes are ignored.

### 6.1 Experimental Results

Two set of experiments were performed with dynamically changing data: in experiment 1, the environment consisted of the quarter data-set for 100 iterations, this was replaced by the half data-set at iteration 101, and then again by the quarter data-set again at iteration 201. Experiment 2 reversed the order of presentation of the data-sets, i.e., first the half set, then the quarter set, etc. Initial experimentation showed that $z = 0.05$ was sufficient to cause the desired behaviour in the system. Figure 5 shows typical behaviour from both experiments over 300 iterations. Points at which lytic cell death occurred are marked. The left-hand figure shows antibodies being added to the system during the first 100 iterations to accurately cluster the quarter data-set with 4 antibodies – lytic death occurs when the data changes and the system is re-initialised with 2 new antibodies, and no further ones need to be added in order to cluster the half-set. At iteration 200, lytic death again occurs and 2 new antibodies are produced (therefore the total number in the system appears in the figure to remain constant), and a further antibody is added to more accurately cluster the quarter set. The right hand-figure basically shows the reverse – it is interesting to note in this figure that during exposure to the final half data-set, the system contentment is consistently just below the threshold $T_s$, resulting in the periodic addition of a new antibody. The new antibody does not succeed in recognising any antigens however, so is always removed due to natural cell death. *d*SOSDM is thus able to dynamically adjust to the correct number of clusters present in the data in each experiment.

## 7 Conclusion

Using ideas from Danger Theory, we have improved the original SOSDM algorithm to make it more able to deal with dynamically changing environments.

Simple experiments have shown that the immune systems produced do suffer uncontrollable expansion and can react to changes in the environment. However, there is still much room for improvement – in particular we plan to modify the mechanism given in section 6 so that antibodies react *individually* to changes in their own contentment induced by their local environment, rather than transmitting values to a central controller. This is a simple modification to the system. This is more in keeping with the principles of the Danger Theory. Furthermore, more complex data-sets will be tested, and comparisons to conventional clustering algorithms given.

## References

1. U. Aickelin and S. Cayzer. The danger theory and its application to artifical immune systems. In J. Timmis and P Bentley, editors, *in Proceedings of the 1st International Conference on Artificial Immune Syste*, pages 141–148, 2002.
2. C.C. Anderson and P. Matzinger. Danger: The view from the bottom of the cliff. *Seminars in Immunology*, 12:231–238, 2000.
3. M. Burgess. Computer immunology. In *in Proceedings of LISA XII*, pages 283–297, 1998.
4. L.N. De Castro and F.J. Von Zuben. An evolutionary immune network for data clustering. In *Proceedings of the IEEE Brazilian Symposium on Artificial Neural Networks*, pages 84–89, 2000.
5. E. Hart and P. Ross. Exploiting the analogy between immunology and sparse distributed memories: a system for clustering non-stationary data. In J. Timmis and P. Bentley, editors, *in Proceedings of the 1st International Conference on Artificial Immune Systems*, pages 49–58, 2002.
6. Emma Hart. *Immunology as a Metaphor for Computational Information Processing: Fact or Fiction*. PhD thesis, University of Edinburgh, 2002.
7. Polly Matzinger. Tolerance, danger and the extended family. *Annual Review Immunology*, pages 991–1045, 1994.
8. M. Neal. An artificial immune system for continuous analysis of time-varying data. In J. Timmis and P. Bentley, editors, *Proceedings of 1st International Conference on Artificial Immune SYste,s (ICARIS)*, pages 76–85, 2002.
9. Jon Timmis and Mark Neal. A resource limited artificial immune system for data analysis. *Knowledge Based Systems*, 14(3-4):121–130, June 2001.