Geometrical Insights into the Dendritic Cell Algorithm

Thomas Stibor
Fakultät für Informatik,
Technische Universität
München,
stibor@in.tum.de

Robert Oates
School of Computer Science
The University of Nottingham
Nottingham, NG8 1BB, UK
rxo@cs.nott.ac.uk

Graham Kendall
School of Computer Science
The University of Nottingham
Nottingham, NG8 1BB, UK
gxk@cs.nott.ac.uk

Jonathan M. Garibaldi
School of Computer Science
The University of Nottingham
Nottingham, NG8 1BB, UK
jmg@cs.nott.ac.uk

ABSTRACT

This work examines the dendritic cell algorithm (DCA) from a mathematical perspective. By representing the signal processing phase of the algorithm using the dot product it is shown that the signal processing element of the DCA is actually a collection of linear classifiers. It is further shown that the decision boundaries of these classifiers have the potentially serious drawback of being parallel, severely limiting the applications for which the existing algorithm can be potentially used on. These ideas are further explored using artificially generated data and a novel visualisation technique that allows an entire population of dendritic cells to be inspected as a single classifier. The paper concludes that the applicability of the DCA to more complex problems is highly limited.

Categories and Subject Descriptors

H.1 [**Models and Principles**]: [General]; I.6.4 [**Simulation and Modeling**]: Model Validation and Analysis

General Terms

Algorithms

1. INTRODUCTION

The dendritic cell algorithm (DCA) is a popular immune-inspired approach for solving anomaly detection problems. The DCA imitates the function of natural dendritic cells and has its inspiration rooted within the danger theory [10, 11]. In recent years the DCA has been investigated from a practical perspective. This has been in the form of benchmarking the DCA on data sets and comparing it to other classification techniques such as self-organising maps [6]. From a mathematical perspective however, the DCA has not been fully investigated and as a consequence, the underlying classification principle is poorly understood. In this paper we demonstrate that the DCA can be represented as a collection of linear classifiers. We justify and underpin this statement by showing that the underlying metric is based on the dot product. From this insight it consequently follows that the geometrical properties of the DCA results in a collection of linear thresholded classifiers.

For the sake of clarity we omit any biological explanation of dendritic cells and danger theory. Explanations of these elements can be found in [3]. The paper is structured as follows: in Section 2 the latest version of the DCA is presented. In Section 3 properties of the dot product are summarized. The linear classifier which is based on the dot product is explained in Section 4. The use of linear classifiers to model the operation of the DCA and a novel visualisation technique for exploring the DCA are explained in Section 5. In Section 6 the results of applying artificially generated data sets to the algorithm are presented. A final discussion and conclusions are provided in Section 7.

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

2. THE DCA

The dendritic cell algorithm (DCA) is an abstraction of the operation of biological dendritic cells. The DCA is a developing algorithm and, as a result, there are several variants. In this paper we will consider the deterministic DCA, presented in [4] as at the time of writing, it is the most recent incarnation of the algorithm and lends itself easily to analysis. The DCA is a population-based algorithm. The population is collection of linear classifiers with their own sample memory, termed 'cells'. The DCA accepts three time-varying streams of data as inputs: two real valued streams, (i.e. in $\mathbb{R}$) and one stream of integers, ($\in \mathbb{Z}$). These data streams are generated by application-specific heuristics. The 'Danger' signal is a real-valued, (0-100) signal which rises proportionally with the 'anomalousness' of the current situation. The 'Safe' signal is a real-valued, (0-100) signal which rises proportionally with the 'normality' of the current situation. The aim is that the Safe and Danger signals are not recipricol, so that the system can make decisions in the presence of conflicting data. The 'Antigen' stream is the integer-based signal. It produces integers that attempt to describe the problem state in a meaningful way. For classification purposes, the antigen stream is simply a list of

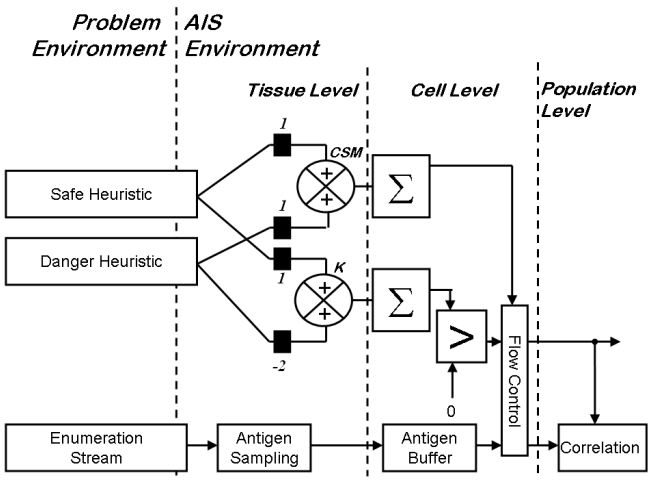


Figure 1: A block diagram representation of the DCA.

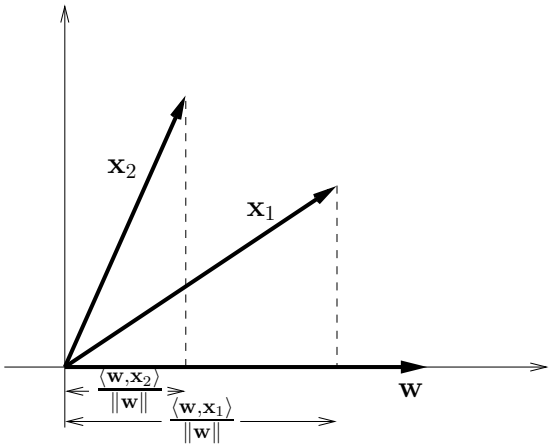


Figure 2: Geometrical interpretation of the dot product of w and the two vectors $\mathbf{x}_1, \mathbf{x}_2$.

identifiers that represent the items in the environment to be classified. In other words the enumeration uniquely identifies an item or state within the problem space which is to be classified as anomalous or normal. The operation of a single cell within the population is illustrated in Figure 1.

The signal inputs, (danger and safe) are combined using weighted sums to generate two intermediate signals CSM and K. The CSM signal is a measure of the overall signal magnitudes that the cell is being exposed to. The K signal is a measure of the instantaneous difference between the normality and anomalousness of the input data. A negative value of K indicates normality and a positive value indicates anomalousness. The values of K and CSM are accumulated over the cell's lifetime. When the cumulated CSM signal reaches a cell-specific threshold, the cell is said to "migrate". The term 'migrate' comes from the biological dendritic cell and indicates the point in its life-cycle where it moves from the tissue and into a lymph node. Algorithmically this means that the cell is ready to classify the antigen that it has sampled as being normal or anomalous, based on the signals that it has been exposed to. When the cell migrates all of the antigen samples that it has collected are given the same classification. Equations 1 and 2 define how these intermediate signals are generated

$$\mathrm{CSM}_n = S_n + D_n \tag{1}$$

$$\mathrm{K}_n = D_n - 2S_n \tag{2}$$

Where D_n is the current sample of the danger signal, S_n is the current sample of the safe signal, CSM_n is the current CSM value and K_n is the current value of the K signal.

The means of collating the outputs from a population of cells varies between versions. A comparison between two popular collation techniques for the DCA is performed in [1]. The most common technique is to use the "mean context antigen value" or MCAV [5]. This technique combines the votes cast by every cell for a specific antigen, using a value of 1 to indicate an anomalous classification and a value of 0 to indicate a normal classification. The MCAV can be calculated at set intervals to give real-time performance. As such the value of the MCAV can be seen as the probability that the sampled antigen are anomalous.

The population-based and signal-dependent nature of the DCA makes it difficult to analyse using conventional techniques. In [12] frequency analysis was used to characterise the behaviour of an individual cell. However, the cells within the population can migrate asynchronously, which makes analysing the algorithm as a whole in the frequency domain challenging [13].

This paper shall focus solely on the signal processing element of the DCA rather than the antigen sampling and classification. For some applications the antigen sampling phase is trivial, as there is a strong temporal link between the presentation of antigen to the system and the resulting signal. In other words, the effects of a state change in the problem space can be sensed almost immediately. It must be stressed that this is not a valid assumption for all problem types and more work is required to understand the relationship between the antigen sampling element of the algorithm and the signal processing layer for non-trivial temporal matching.

3. DOT PRODUCT

In this section we summarize well known and straightforward mathematical facts of the dot product. For further details see any textbook on linear algebra (e.g. [9]).

Given two vectors $\mathbf{w}, \mathbf{x} \in \mathbb{R}^d$, the dot product[1] is defined as follows:

$$\mathbf{w}^T \cdot \mathbf{x} = \sum_{i=1}^{d} w_i x_i = \langle \mathbf{w}, \mathbf{x} \rangle \tag{3}$$

The dot product has a straightforward geometrical interpretation, namely, the length of the projection of $\mathbf{x}$ onto the unit vector $\mathbf{w}/\|\mathbf{w}\|$ (see Figure 2).

Furthermore, the dot product can be used to calculate lengths, angles and distances.

[1]Also called inner product or scalar product.

Squared lengths of $\mathbf{w}$:

$$\|\mathbf{w}\|^2 = w_1 w_1 + w_2 w_2 + \ldots + w_d w_d = \langle \mathbf{w}, \mathbf{w} \rangle \quad (4)$$

Squared Euclidean distance between $\mathbf{w}$ and $\mathbf{x}$:

$$\|\mathbf{w} - \mathbf{x}\|^2 = \langle \mathbf{w}, \mathbf{w} \rangle + \langle \mathbf{x}, \mathbf{x} \rangle - 2\langle \mathbf{w}, \mathbf{x} \rangle \quad (5)$$

Cosine angle between $\mathbf{w}$ and $\mathbf{x}$:

$$\cos\phi = \frac{\langle \mathbf{w}, \mathbf{x} \rangle}{\|\mathbf{w}\| \, \|\mathbf{x}\|} \quad (6)$$

$$= \frac{w_1 x_1 + w_2 x_2 + \ldots + w_d x_d}{\sqrt{w_1^2 + w_2^2 + \ldots + w_d^2}\sqrt{x_1^2 + x_2^2 + \ldots + x_d^2}} \quad (7)$$

If $\mathbf{w}$ and $\mathbf{x}$ are perpendicular, then the dot product $\langle \mathbf{w}, \mathbf{x} \rangle$ is zero.

4. LINEAR CLASSIFIERS

Linear classifiers are well known and investigated in the field of neural networks. We present linear classifiers from this perspective and show that the DCA can be modelled as a collection of linear classifiers.

A single layer neural network (also called perceptron) is pictured in Figure 3(a) and corresponds to the linear discrimination function

$$y(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + w_0 \quad (8)$$

where $\mathbf{x}$ is a d-dimensional input vector, $\mathbf{w}$ is d-dimensional weight vector and w_0 the bias (threshold).

The decision boundary $y(\mathbf{x}) = 0$ corresponds to a $(d-1)$-dimensional hyperplane in d-dimensional $\mathbf{x}$-space[2]. The weight vector $\mathbf{w}$ defines the orientation of the decision hyperplane, the bias w_0 the position in terms of its perpendicular distance from the origin. If w_0 is zero, then the hyperplane goes through the origin. The distance of $\langle \mathbf{w}, \mathbf{x} \rangle + w_0$ to the origin is $|w_0|/\|\mathbf{w}\|$, the distance of $\langle \mathbf{w}, \mathbf{x} \rangle + w_0$ to an arbitrary point $\mathbf{b} \in \mathbb{R}^d$ is $|y(\mathbf{b})|/\|\mathbf{w}\|$. Moreover, weight vector $\mathbf{w}$ is perpendicular to $\langle \mathbf{w}, \mathbf{x} \rangle + w_0$ and this consequently implies that $y(\mathbf{a}) = 0$ for any point $\mathbf{a}$ which lies on the hyperplane (see Figure 3(b)).

Due to the fact that the dot product can also be interpreted as the cosine angle ϕ between $\mathbf{w}$ and $\mathbf{x}$, it follows that (8) separates the $\mathbf{x}$-space in the two half-spaces $y(\mathbf{x}) < 0$ and $y(\mathbf{x}) > 0$ (see Figure 3(b)).

4.1 Linear Data Separation

Given sample $\mathcal{X} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$ from $\mathbb{R}^d \times \{-1, +1\}$. If $\mathcal{X}$ is linearly separable, then there exist vector $\mathbf{w}$ and bias w_0 such that

$$y^{(n)}(\langle \mathbf{w}, \mathbf{x}^{(n)} \rangle + w_0) \geq 0, \quad n = 1, 2, \ldots, N. \quad (9)$$

More specifically, the two parameters $\mathbf{w}$ and w_0 have to be inferred from $\mathcal{X}$ such that (9) is satisfied. This inference problem can be solved for instance with the perceptron update rule, the delta rule or by the pseudoinverse method which gives an analytical solution [2]. If $\mathcal{X}$ is not linearly separable, then the inference task can be formulated as linear Support Vector Machine separation problem

$$\min_{\mathbf{w}, w_0, \boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 \quad (10)$$

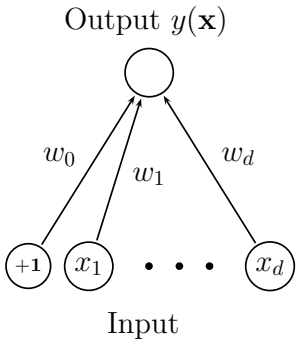
$$\text{subject to} \quad y^{(n)}(\langle \mathbf{w}, \mathbf{x}^{(n)} \rangle + w_0) \geq 1 - \xi_n, \quad (11)$$

$$\xi_n \geq 0, \, n = 1, 2, \ldots, N,$$

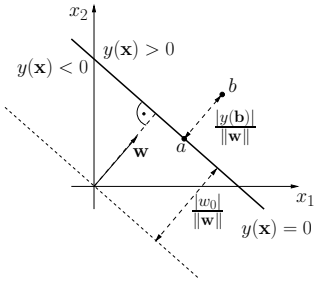
where ξ_n are slack variables to relax the hard separation constrains.

To summarize, parameters in learning algorithms for data separation problems have to be inferred from the data, that is, from sample $\mathcal{X}$. We will see in the subsequent sections, that parameters in the DCA are inferred in terms of speculative user generated heuristics.

[2]For $d = 2$ it is a straight line, for $d = 3$ a plane, and so on.



(a) Representation of a single layer neural network. The output is $y(\mathbf{x}) = w_0 + x_1 w_1 + x_2 w_2 + \ldots + x_d w_d$.



(b) The decision boundary in the space $\mathbb{R}^2$ is a separating line.

Figure 3: Geometrical interpretation of a single layer network and the corresponding linear discrimination function $y(\mathbf{x})$ which separates the space in the two half-spaces $y(\mathbf{x}) < 0$ and $y(\mathbf{x}) > 0$.

5. REPRESENTING THE DCA AS A COLLECTION OF LINEAR CLASSIFIERS

As discussed in Section 4, for typical classifier analysis it can be instructive to view the positions of the classification hyperplanes with respect to the input signal space. However, the DCA does not operate on the input signals directly, rather the sum of the signals experienced over a given cell's lifetime. This presents a challenge as the population is asynchronous and a cell can migrate at any time relative to the rest of the population depending on its input history. Conventional techniques for visualising linear

classifiers that use their input history (as opposed to their instantaneous inputs) usually treat each sample as a new input dimension. However, the number of samples a dendritic cell analyses varies depending on signal magnitude via the CSM gating mechanism, so the resulting space would have varying dimensionality between cells and input magnitudes. To overcome this challenge, rather than using the instantaneous input signals as axes, we can use the cumulated input signals. In this space the algorithm becomes easier to explore.

Each cell generates two planes in this space. The first is the decision boundary. The cell will not make a decision, (i.e. migrate) until its current decision boundary is breached. At the first instance where the cell's decision boundary is breached, the point is compared to the second plane, the classification boundary, to decide if the cell is voting for normality or anomalousness.

The inequality defining when a cell migrates is given in equation 12

$$M_i \leq \sum_{n=0}^{T} \mathrm{CSM}_n \tag{12}$$

Where M_i is the migration threshold of cell i and T is the index of the current sample. Substituting the definition for CSM_n from equation 1 into equation 12 and rearranging for $\sum$ safe gives equation 13.

$$\sum_{n=0}^{T} S_n = -1 \times \sum_{n=0}^{T} D_n + M_i \tag{13}$$

Where $\sum S_n$ is used as the y axis and $\sum D_n$ is used as the x axis. However, this only holds in the individual cell's frame of reference. To express equation 13 in the global co-ordinate frame, the line must be offset by the starting position of the cell in signal space, (x_P, y_P). Equation 13 shows that all decision boundaries are parallel going from $(x_P + M_i, 0)$, to $(0, y_P + M_i)$ with a constant gradient of -1. The classification boundary can be defined similarly, using the inequality in equation 14, which defines 'normality'.

$$\sum_{n=0}^{T} K_n > 0 \tag{14}$$

Substituting equation 2 and rearranging as before provides equation 15, the classification boundary.

$$\sum_{n=0}^{T} S_n = 0.5 \times \sum_{n=0}^{T} D_n \tag{15}$$

This also gives a constant gradient, of 0.5 in this case, implying that all of the classification boundaries are also parallel. This demonstrates that not only can an individual cell be modelled as a linear classifier, but a population of cells is limited to only produce those classification boundaries that can be expressed as combinations of parallel planes.

Rearranging equations 13 and 15 into the dot product representation for a linear classifier yields equations 16 and 17 respectively.

$$y_D(\mathbf{x}) = \langle \mathbf{W_D}, \mathbf{x} \rangle - M_i \tag{16}$$

Where y_D is the decision boundary, W_D is a vector representing the weightings of the safe and danger signals towards the CSM signal and M_i is the migration threshold of the cell being considered. For the current version of the DCA $\mathbf{W_D}$ is $[1, 1, 1, 1, ...]$ and $\mathbf{x}$ is $[D_n, S_n, D_{n-1}, S_{n-1}...]$ where the length of the vectors is defined by the number of samples required to satisfy equation 12.

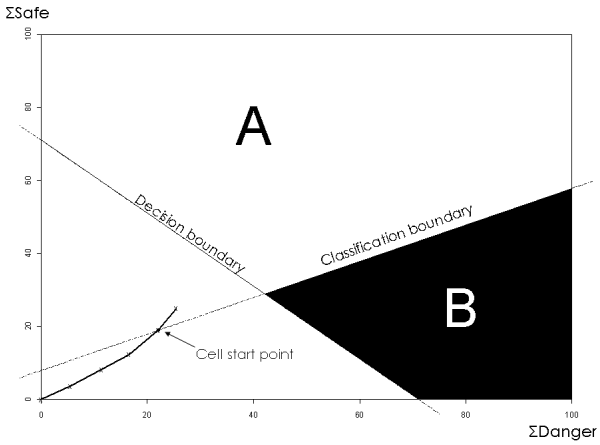


Figure 4: A single dendritic cell's cumulative input signal space.

$$y_C(\mathbf{x}) = \langle \mathbf{W_C}, \mathbf{x} \rangle \tag{17}$$

Where y_C is the classification boundary and W_C is a vector representing the weightings of the safe and danger signals towards the K signal. For the current version of the DCA $\mathbf{W_C}$ is $[1, -2, 1, -2, ...]$ and $\mathbf{x}$ is the same varying length vector described in equation 16.

5.1 Visualising a Single Dendritic Cell

Figure 4 illustrates the visualisation of the decision boundary and classification boundary for a single dendritic cell.

The x axis is the cumulated danger signal that the cell has been exposed to and the y axis is the cumulated safe signal that the cell has been exposed to. Each cross represents a change in input signal experienced by the cell and the connecting solid black line represents the path through the signal space that the algorithm's previous inputs have described. Note that as this is cumulative space and the inputs are constrained to be positive, that the path is constrained to never move to the left or down. The black spot represents the point at which the cell last migrated and had its cumulated CSM and K signal reset to 0. The thin, solid line indicates the decision boundary, the first input signal that breaches that boundary will lead to a classification. If the input signal causes the path to enter the white region, (marked 'A') the cell will classify all collected antigen as being "safe". Conversely, if the input signal enters the black region, (marked 'B') the cell will classify all collected antigen as being "anomalous". These two regions are divided by the thin, dashed line, which illustrates the classification boundary for the cell. If the next signal change fails to breach the decision boundary, no classification shall be returned by the cell and it shall simply wait until sufficient data has been sampled.

5.2 Visualising a Population of Dendritic Cells

In order to visualise a population of cells, it is necessary to define how the output of the cells will be combined. For this paper it shall be assumed that the MCAV is calculated after every iteration. To simplify analysis, it will also be assumed that there is only one type of antigen being classified. The first assumption is reasonable for a real-time implementation of the algorithm. The second assumption is not realistic in an application sense. However, it is instructive for demonstrating the different classification region shapes that the cell population can construct. For the multi-cell visualisation, the same key can be used to identify the positions of the classification and decision boundaries, but instead of a black and white region, a grey-scale map will be produced that illustrates the possible outcomes for the next signal input. As before, black illustrates an anomalous region, (i.e. the probability that the data is normal approaches 0) and white illustrates a normal region, (i.e. the probability that the data is normal approaches 1). To illustrate a population of cells, random data shall be presented to the algorithm, with a Gaussian distribution. By using Gaussian distributions for both signal sources, it is hoped that the generated data will exemplify 'normality' for the most part, but will also produce situations where 'anomalous' signal characteristics will be generated at the extremes of the Gaussian curves. Two sets of data shall be used, both with a standard deviation of 1, but the first shall use a mean of 5 and the second shall use a smaller mean of 2. In each case a population of 100 cells will be used, with a uniformly distributed set of migration thresholds, ranging from 15 to 45, (a standard range for the DCA.)

6. RESULTS

Figure 5 illustrates four steps through the algorithm, using randomly generated input data with a mean of 5. Step five was chosen as a starting position as it allows the algorithm to settle into its usual operation. At step 0 all cells have been exposed to exactly the same amount of signal (i.e. 0), so the classification boundaries are all exactly the same.

In Figure 5(a) there are only three distinct classification boundaries, (the dashed lines.) This is because the cells have formed into three groups of cells. This separation is also observable by inspecting the decision boundaries, which have clearly split into three clusters. The gaps between the decision boundary groups are formed by the magnitude of the input data to the algorithm. Large steps cause many cells to migrate at once. The black circles on the data path mark the steps at which the cells that are part of the current population were last reset. Only three circles exist, showing that this population has no cells older than three iterations. The shape of the classification regions are all constructed from an averaging of linear classification boundaries. The shading demonstrates that in certain regions the classification is highly sensitive to change. As a result, in the boundary between normal and anomalous, small changes in the input signal can result in drastically different classification outputs. In this case it is even possible for a counter-intuitive situation to arise, where increasing the danger signal slightly can cause the classification to become normal. This is because cells that have been exposed to more normal signal in their overall lifetime can be forced to migrate by either signal, and can outnumber the cells voting for anomalousness.

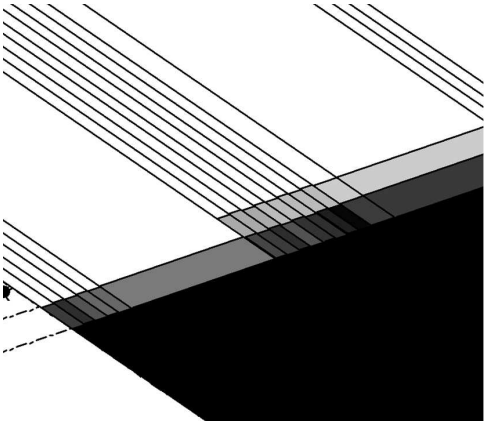


Figure 6: An expanded view of an example boundary region. Here we can see that the classification distribution is not gradiated, but varies significantly for minor changes in input signal.

This is a potential source of error for the classification, as it cannot be pre-trained or controlled, it is simply a function of the input data. Figure 6 is an expanded view of an example boundary region between $\sum \text{safe} = 20$ and $\sum \text{safe} = 40$. Here it is possible to discern that the shading is not gradiated, but pseudo-randomly distributed, according to the density of the overlapping decision boundaries and the previous inputs.

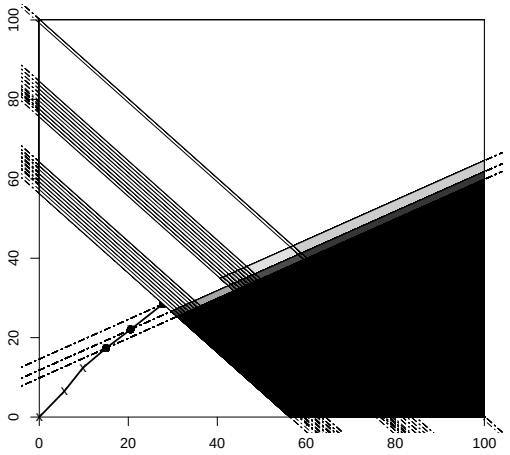
In Figure 5(b) many of the cells are forced to migrate as the input signal path passes through their decision boundaries. All of the cells in this case classify their collected data as normal, though this is unsurprising given that the random input signals have equal means and that the safe signal is weighted more heavily than the danger in the decision making process. Note how the first and second set of cells are pushed closer together by the migration process. This highlights a further potential problem with the algorithm, as the number of cells that migrate for a given input is going to be exceptionally hard to predict, as the density of decision boundaries is going to be highly dependant on the input data.

In Figure 5(c) the input signal fails to breach any decision boundaries. This results in the classification boundaries staying as they are. In practical terms, this would represent the algorithm not generating any output, and waiting to receive more data.

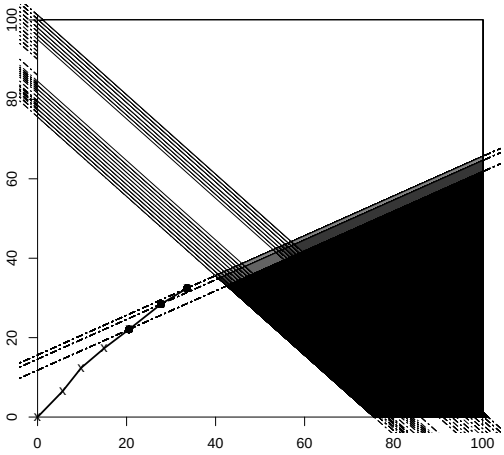
Finally, in Figure 5(d), several cells migrate and widen the 'grey area' between normal and anomalous. The wider region is indicative of the size of the signal that caused the migration to occur. In other words, the width of the classification boundaries is representative of the diversity in accumulated CSM for the cells in the population.

By using a Gaussian distribution with a smaller mean the effects of smaller input signals, relative to the selected migration thresholds can be observed. Figure 7 shows steps 5 through 8 for the algorithm, using randomly generated data with a mean of 2.

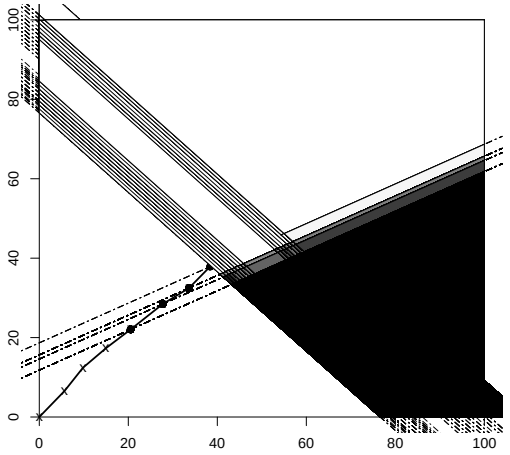
In Figure 7(a) there are only two distinct classification boundaries. The black spot on the origin is a sign that many of the cells are yet to migrate. The light colour of the region at the anomalous/normal border indicates that in fact,



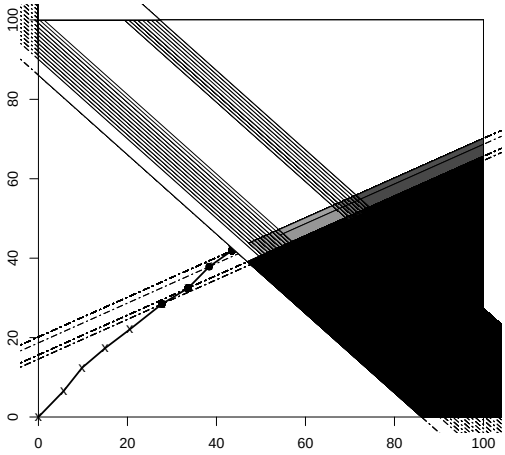
(a) Step 5



(b) Step 6



(c) Step 7



(d) Step 8

Figure 5: Watching the DCA decision boundaries move. These figures illustrate the 5th, 6th, 7th and 8th iterations of the algorithm responding to randomly generated input data. The population size is 100, and the migration thresholds are between 15 and 45. The input data is generated using a Gaussian probability distribution, with a mean of 5 and a standard deviation of 1.

very few cells have migrated. Again, the large width of that region is indicative of the large distribution of accumulated CSM within the population.

Figure 7(b) shows the input signal path moving with a gradient approximately equal to the gradient of the classification boundaries. This is the equivalent of receiving sufficient volumes of conflicting data in the short term to cause the certainty of the classification to drop. However, in this case, the majority of the cells are yet to migrate, so the longer-term view of the input signals causes the population to be heavily biased towards a normal classification. Contrasting the positions of the decision boundaries between Figure 7(b) and Figure 5(b) demonstrates some of the effects of smaller magnitudes of input data. The decision boundaries are more clustered together, indicating that if the signal strength remains at this level, the population will continue to output classifications at a quite steady rate, rather than the more erratic output from a larger input signal magnitude. This is further supported by Figures 7(c) and 7(c) where each subsequent presentation of input data causes an additional set of cell migrations.

The large number of black spots in Figure 7(d) indicates that the population has a diverse number of classification boundaries, though they are extremely close together.

7. CONCLUSIONS

The equations defining the decision and classification boundaries for the DCA raise many issues with the algorithm. Primarily, as a collection of linear classifiers, there are severe limitations on the data sets that the algorithm will be able to assess. This is made worse by the fact that the gradients of the boundaries are constant, applying still further restrictions onto the regions in signal space that the algorithm can discriminate between. It is likely that for the performance of the DCA on complex data sets to be competitive with other classification-based techniques that it will become necessary to introduce non-linearity into the classification boundaries. In contrast to learning machines where the model is inferred from the data set, the DCA requires the user to construct a model apriori to fit the hard coded weights. However, as with all expert system solutions, the optimal separation for the problem is unlikely to be found by the user. An alternative to this is discussed in [7]. In this work, the author suggests that a kernel method could be introduced to the DCA to expand the possible regions in signal space that could be discriminated between. This would in itself be challenging, as parameterising the appropriate kernel would be a non-trivial task. This overcomes the limitation of the achievable classification shapes, but not the issues associated with requiring the user to generate the model. In fact, this task is made more difficult through the introduction of non-linearity.

In this paper a novel visualisation technique for exploring the DCA was suggested. The technique has made it possible to make predictions about how the algorithm will react to input data. It also makes it possible for inferences about the flow of data and the shape of the classification boundaries to be made. For the first time it is possible to predict signal combinations that will result in the algorithm's behaviour becoming highly sensitive to subtle changes in signal magnitude, and in the future may facilitate better tuning of the algorithm based on test data. The visualisation technique also makes it possible to comment on the diversity of the population and how that impacts on the classification for a given sample of input data. In short, the technique allows a deeper insight into the families of problems for which the DCA is an appropriate tool.

More insight into the algorithm could potentially be gained by applying the visualisation technique to real data from an application where the DCA has had success. Comparing this with a modified, non-linear version of the algorithm could potentially yield a more powerful addition to the field of artificial immune systems. However, it would be important to benchmark this solution against an established non-linear classifier, such as the kernel perceptron (for details see [8]). Such a comparison would have to be highly favourable to justify the increased effort associated with hand tuning the model.

8. REFERENCES

[1] Y. Al-Hammadi, U. Aickelin, and J. Greensmith. Dca for bot detection. In *Proceedings of the IEEE World Congress on Computational Intelligence (WCCI) Hong Kong*, pages 1807–1816, 2008.

[2] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] J. Greensmith. *The Dendritic Cell Algorithm*. PhD thesis, The University of Nottingham, Computer Science, Jubilee Campus, Wollaton Road, Nottingham, NG8 1BB, 2007.

[4] J. Greensmith and U. Aickelin. The deterministic dendritic cell algorithm. In *7th International Conference on Artificial Immune Systems (ICARIS2008)*, pages 291–303, 2008.

[5] J. Greensmith, U. Aickelin, and J. Twycross. Articulation and clarification of the dendric cell algorithm. In *Proceedings of the 5th International Conference on Artificial Immune Systems (ICARIS 2006)*, pages 404–417, 2006.

[6] J. Greensmith, J. Feyereisl, and U. Aickelin. The dca: Some comparison: a comparative study between two biologically-inspired algorithms. *Evolutionary Intelligence*, 1(2):85–112, June 2008.

[7] T. S. Guzella, T. A. Mota-Santos, and W. M. Caminhas. Artificial immune systems and kernel methods. In *ICARIS*, pages 303–315, 2008.

[8] R. Herbrich. *Learning Kernel Classifiers: Theory and Algorithms*. MIT Press, 2001.

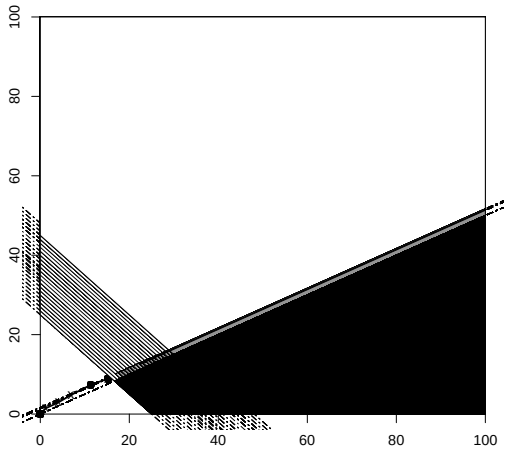
[9] G. James. *Modern Engineering Mathematics*. Pearson Education Limited, 4th edition, 2008.

[10] P. Matzinger. Tolerance, danger, and the extended family. *Annual Review of Immunology*, 12(1):991–1045, 1994.

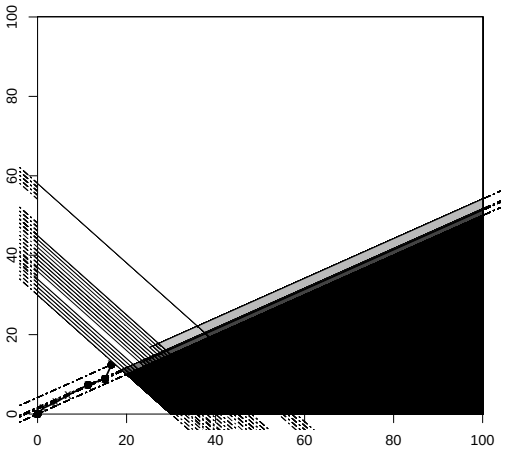
[11] P. Matzinger. Friendly and dangerous signals: is the tissue in control? *Nature Immunology*, 8(1):11–13, 2007.

[12] R. Oates, G. Kendall, and J. Garibaldi. Frequency analysis for dendritic cell population tuning. *Evolutionary Intelligence*, 1(2):145–157, June 2008.

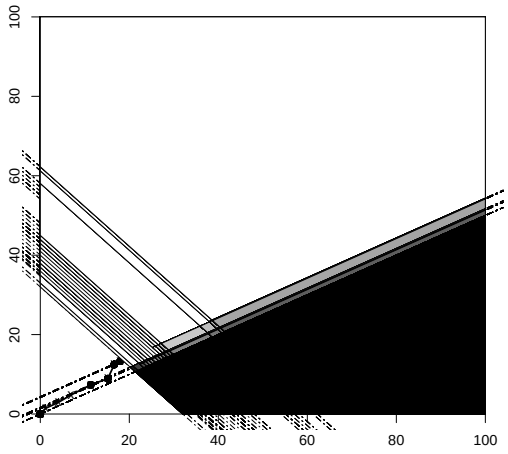
[13] R. Oates, G. Kendall, and J. M. Garibaldi. The limitations of frequency analysis for dendritic cell population modelling. In *ICARIS '08: Proceedings of the 7th international conference on Artificial Immune Systems*, pages 328–339, Berlin, Heidelberg, 2008. Springer-Verlag.



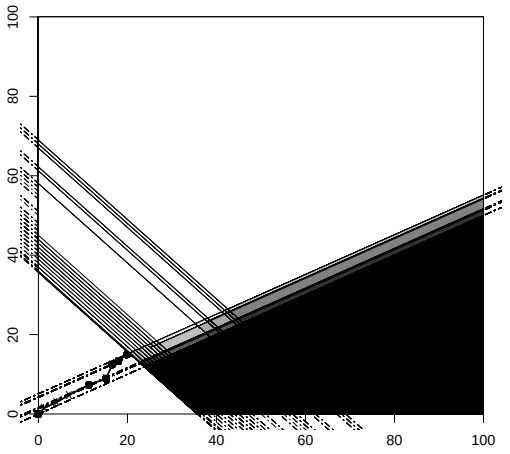
(a) Step 5



(b) Step 6



(c) Step 7



(d) Step 8

Figure 7: Watching the DCA decision boundaries move. These figures illustrate the 5th, 6th, 7th and 8th iterations of the algorithm responding to randomly generated input data. The population size is 100, and the migration thresholds are between 15 and 45. The input data is generated using a Gaussian probability distribution, with a mean of 2 and a standard deviation of 1.