Theoretical Aspects of the Dendritic Cell Algorithm

Feng Gu, Julie Greensmith, Wei Chen and Uwe Aickelin

Intelligent Modelling & Analysis Group
School of Computer Science
University of Nottingham, UK

November 17th, 2009

Motivation of the work

- The lack of a formal definition of the DCA
  - **Ambiguities** for understanding the algorithm
  - **Incorrect** applications and implementations
- Theoretical analysis of the DCA
  - **Other algorithms in AIS** were theoretically analysed
  - Theoretical analysis of its **runtime complexity**
  - Induce and reveal its **mathematical properties**
  - Formally investigate its **classification performance**

Scope of the work

- Formal definition of the **data structures**
- Formal definition of the **procedural operations**
- Theoretical analysis of the **runtime complexity**
- Formulation of two **runtime variables**
- Analysis of **temporal correlation**

What is the DCA

- Based on a behavioural model of **dendritic cells**
- Incorporated with the **danger theory**
- **Immune inspired** and **population based**
- A correlation algorithm based on **time series**
- Designed for **anomaly detection**
- The dDCA developed for **ease of analysis**

Data structures

- **Input data** consist of time-dependent data instances
- **Weight matrix** for the operation of signal transformation
- **DC population** consists of a group DC objects
- **Output list** for recording the output of each DC object

Data structures

Definition (input data)

Let a set Signal $\subset \mathbb{R}^2$ and a finite set Antigen $\subset \mathbb{N}$ be the two types of input data instances.

$$S : \text{Time} \longrightarrow \text{Signal} + \text{Antigen}$$

Time	1	2	3	...	n
Instance	$S(1)$	$S(2)$	$S(3)$	...	$S(n)$

Definition (weight matrix)

Let the predefined weights matrix for signal transformation be defined as follows.

$$W : 2 \times 2 \text{ Matrix}, \text{ where } w_{ij} \in \mathbb{N}$$

w_{11}	w_{12}
w_{21}	w_{22}

Data structures

Definition (DC population)

Let Population be a group of objects with a size $N = |\text{Population}|$, each object contains the following data structures.

- $i \in \mathbb{N}$ is the index of a DC object
- $l(i) \in \mathbb{R}$ is the initial lifespan of a DC object
- $K(i) = 0$ is the initial signal profile of a DC object
- List Antigen is a list for storing the antigen profile

Index	1	2	3	...	N
Object	$DC(1)$	$DC(2)$	$DC(3)$	...	$DC(N)$

Data structures

Definition (output list)

Let lst : List $\mathbb{N} \times \mathbb{R}$ be a list for recording the output of each DC object, where List is a data structure in which elements are arranged in a linear order, and $lst(j)$ ($j \in \mathbb{N}$) is an element of lst.

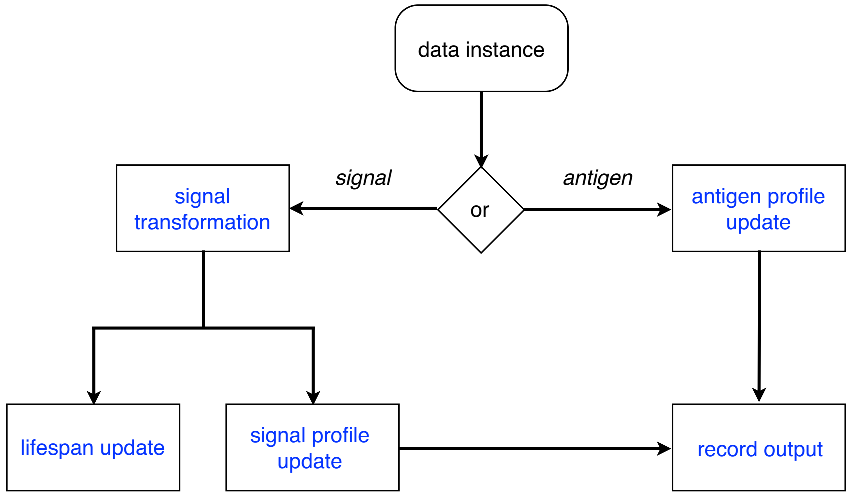
$$\varphi_1 : \mathbb{N} \times \mathbb{R} \rightarrow \mathbb{N} \text{ (first dimension)}$$
$$\varphi_2 : \mathbb{N} \times \mathbb{R} \rightarrow \mathbb{R} \text{ (second dimension)}$$

Index	1	2	3	...	n
Element	$lst(1)$	$lst(2)$	$lst(3)$	...	$lst(n)$

Procedural operations

- **Initialisation phase** for weight matrix and DC population
- **Detection phase** for input data and DC population
  - signal transformation
  - lifespan update
  - signal profile update
  - antigen profile update
  - record output
- **Analysis phase** for output list
  - analysis calculation

Flowchart of detection phase



Detection phase

Definition (signal transformation)

Let the function of transferring input signal instances to output signal instances be as follows.

$$O : \text{Time} \to \mathbb{R} \times \mathbb{R}$$

$$O(t) = \begin{cases} S(t) \times W^T & \text{if } S(t) \in \text{Signal} \\ \text{Null} & \text{otherwise} \end{cases}$$

$$\pi_1 : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \text{ (first dimension)}$$

$$\pi_2 : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \text{ (second dimension)}$$

Detection phase

Definition (lifespan update)

Let the function of updating the lifespan of a DC object during runtime be as follows.

$$F : \text{Time} \times \text{Population} \to \mathbb{R}$$

$$F(t,i) = \begin{cases} I(i) & \text{if } t = 0 \\ I(i) - \pi_1(O(t)) & \text{if } F(t-1,i) \leq 0 \\ F(t-1,i) - \pi_1(O(t)) & \text{otherwise} \end{cases}$$

Detection phase

Definition (signal profile update)

Let the function of updating the internal signal profile of a DC object be as follows.

$$G : \text{Time} \times \text{Population} \to \mathbb{R}$$

$$G(t,i) = \begin{cases} K(i) & \text{if } t = 0 \\ K(i) + \pi_2(O(t)) & \text{if } F(t-1,i) \leq 0 \\ G(t-1,i) + \pi_2(O(t)) & \text{otherwise} \end{cases}$$

Detection phase

Definition (antigen profile update)

Let the function of updating the internal antigen profile of a DC object be as follows.

$$H : \text{Time} \times \text{Population} \to \text{List Antigen}$$

$$H(t,i) = \begin{cases} \text{Null} & \text{if } t = 0 \\ Insert(S(t), H(t-1,i)) & \text{if } S(t) \in \text{Antigen} \end{cases}$$

Detection phase

Definition (record output)

Let $R(t,i) = \{(r,a) | r = G(t,i) \text{ and } a \in H(t,i)\}$ be the output of a DC object, and the function of recording output is defined as follows.

$$(F(t,i) \leq 0) \wedge (t > 0)$$
$$\Longrightarrow Insert(R(t,i), lst)$$

Outline
Introduction
The Dendritic Cell Algorithm
Theoretical Analysis of the DCA
Conclusions and Future Work
Algorithm overview
Formal definition

Analysis phase

Definition (analysis calculation)

Let the function of analysis calculation performed per antigen type α is defined as follows.

$$C_\alpha : \text{List } \mathbb{N} \times \mathbb{R} \rightarrow \{0,1\}$$
$$C_\alpha(j) = \begin{cases} 1 & \text{if } \varphi_1(lst(j)) = \alpha \\ 0 & \text{otherwise} \end{cases}$$
$$R_\alpha : \text{List } \mathbb{N} \times \mathbb{R} \rightarrow \mathbb{R}$$
$$R_\alpha(j) = \begin{cases} \varphi_2(lst(j)) & \text{if } \varphi_1(lst(j)) = \alpha \\ 0 & \text{otherwise} \end{cases}$$

φ_1	φ_2
1	0.5
1	1.2
2	2.8

$\alpha = 1$

C_α	R_α
1	0.5
1	1.2
0	0

Runtime complexity of algorithms

- If the input size is n and **relatively large**
- The number of **primitive steps** executed
- Upper bounds for **worse case** analysis
- Lower bounds for **best case** analysis
- **Order of growth** to identify the leading term
- **Asymptotic notations** (O, o, Ω, and ω)

Outline
Introduction
The Dendritic Cell Algorithm
Theoretical Analysis of the DCA
Conclusions and Future Work
Analysis of runtime complexity
Runtime complex of the DCA

Detection phase

Code

```
while (data instance) do                  (n)
    if (antigen) then                     (a)
        antigen profile update;
    else if (signal) then                 (n-a)
        for (each DC object) do           (N)
            lifespan update;
            signal profile update;
            if (termination condition) then
                record output;
                reset DC object;
```

Running time of detection phase

Running time $T_1(n)$

The **while** loop is executed in n steps. The first **if** statement is executed in a steps, and the second **if** statement is executed in $n-a$ steps. The **for** loop involves $N = |\text{Population}|$ steps, which forms a double-nested loop, so this **if** statement includes $(n-a)\times N$ steps.

$$T_1(n) = a + (n-a)\times N$$
$$\{N \text{ is a constant}\}$$
$$\Longrightarrow \lim_{n\to+\infty} T_1(n) = \lim_{n\to+\infty} a + (n-a)\times N = a + (n-a) = n$$

Analysis phase

Code

```
while (output list) do                        (a)
   for (each antigen type) do                 (b)
      analysis calculation;
```


Running time of analysis phase

Running time $T_2(n)$

The **while** loop depends on the size of antigen instances a, and the **for** loop depends on the number of antigen types $b = |\text{Antigen}|$. The number of steps required for this phase is $a \times b$, and $1 \le b \le a \le n$.

$$T_2(n) = a \times b$$
$$\{1 \le b \le a \le n\}$$
$$\Longrightarrow a \le T_2(n) \le a^2$$
$$\{a \approx n \text{ if } n \to +\infty\}$$
$$\Longrightarrow n \le T_2(n) \le n^2$$

Runtime complexity of the DCA

Theorem (standard DCA)

The runtime complexity of the DCA is $\Omega(n)$ and $O(n^2)$.

Proof.

$$T(n) = 1 + T_1(n) + T_2(n)$$
$$\Longrightarrow 1 + n + n \le T(n) \le 1 + n + n^2$$
$$\{\text{the order of growth}\}$$
$$\Longrightarrow n \le T(n) \le n^2$$

The bound of Ω-notation or O-notation is asymptotically tight. □

Conclusions

- A **formal definition** of the DCA is given to prevent misunderstandings and ambiguities
- The DCA achieves the runtime complexity of a **lower bound** of $\Omega(n)$ and an **upper bound** of $O(n^2)$
- The runtime complexity of each analysis can be reduced by **segmentation** to $o(n \log n)$ even in the worst case scenario
- Two **runtime variables** of the algorithm are formulated from the input to predict the algorithm's runtime behaviours
- Formal induction of the conditions for achieving **high classification performance**

Future work

- Various types of **distributions of input data** should be tested using our formulas
- **Theoretical analysis** of other properties of the algorithm
- The relationship between the **weight matrix** and the **detection performance** should be investigated

Thank you very much! Any questions?