Artificial Immune System

Magnus Erik Hvass Pedersen (971055)
Daimi, University of Aarhus, May 2003

1 Introduction

The purpose of this document is to verify attendance of the author to the *Swarm Intelligence* course at DAIMI, University of Aarhus.

First the basic artificial immune system is surveyed with emphasis on anomaly detection followed by a number of applications, and finally some thoughts on soft computing and evolution in general are given.

The reader is assumed to be familiar with swarm intelligence and related topics. Furthermore, since this is a document of limited span, the essential results and ideas from the source material are presented *as is* - without thorough verification. The source material is used throughout without explicit reference.

2 Artificial Immune System (AIS)

Different aspects of the immune system (IS) have been modelled for solving different problems including anomaly detection, clustering, and function optimization. Since the interest is on the computational model, the use of biological terms is limited and simplified in the following.

2.1 Anomaly Detection

The obvious feature of the IS is its ability to protect an organism from harmful agents known as *pathogens*, such as bacteria and vira. The concept is simple: Find the pathogen, identify it as harmful, and destroy it. The cell responsible for this is the *lymphocyte*[1]. Assuming the pathogen has already been found, the distinguishing between harmful and harmless is the focus of our attention, and the destruction of harmful pathogens is replaced in an implementation by a context-appropriate response.

The objective of AIS in anomaly detection is to minimize damage while maximizing usability. But being completely usable, the system would have no protection, being completely safe the system would not be usable. Once again it is a matter of balancing requirements.

2.1.1 Self, Non-self

Although the optimal classification of pathogens is either as harmful or harmless, the IS works slightly different. Normally the lymphocytes do not know what a harmful pathogen looks like, because this information is not encoded into

[1] There are different kinds in the IS but this model combines their features.

the organism, it only knows what itself looks like. So the organism trains the lymphocytes to look for pathogens that do *not* look like anything it knows, i.e. *non-self*.

2.1.2 Negative Selection

When a new lymphocyte is created, this training is achieved by a process known as *negative selection*. The lymphocyte is exposed for a certain amount of time to *self*, if it recognizes any of these it is destroyed. This ensures that a lymphocyte will only recognize *non-self*, and this is a quality that is desired in many protection systems since it avoids the explicit definition of abnormal behaviour.

2.1.3 Costimulation

In the IS it is possible for lymphocytes to bind to *self*. Such suicidal behaviour is of course unwanted, and the solution is known as *costimulation* in which the lymphocyte will die if it does not receive a second signal. When tissue is damaged this second signal will be chemical, but in the AIS costimulation is not always easy to model, as damage to a hardware or software system may be hard to assess.

If costimulation is required then the matching of lymphocytes and pathogens can be even less rigid as costimulation provides the ultimate gate before a response is launched.

2.1.4 Lymphocyte

When a lymphocyte has survived negative selection, it is said to be mature. The life of a mature lymphocyte is relatively short though, and it is necessary for it to bind with a pathogen a number of times to get activated. If costimulation is required but is absent, the lymphocyte will also die.

Such a hard life serves a number of purposes. First off if the lymphocyte makes it all the way, it will become a memory cell and will be rewarded with longevity (possibly indefinite), and it will only require one binding with a pathogen to become activated in the future. This corresponds to primary and secondary response of the IS, in which the primary response is not as effective until the IS catches up, but the secondary is almost immediate.

The second purpose is to create diversity. A lymphocyte recognizes a limited number of different pathogens. Continuously replacing useless lymphocytes with mutated (or random) ones, increases the number of recognizable pathogens over time.

2.1.5 Pathogen Binding

The binding of a lymphocyte to a pathogen in the IS is done by protein matching. Because there are so many lymphocytes, this matching can be rather strict, whereas it makes sense to use a less rigid match in the AIS with its (comparatively) small amount of lymphocytes.

Implementation-wise the AIS can use binary strings for the matching (binding) of lymphocytes and pathogens, with the matching criterion being a number of equal contiguous bits, Hamming distance, etc. It is however useful to abstract from this and consider any *object* that can be compared for similarity and is capable of mutation or random generation, as a possible representation.

2.1.6 Mutation

When a lymphocyte dies, it is replaced with either a randomly generated or a mutation of one of the best lymphocytes (as measured by their success in matching pathogens). Different schemes are possible for the mutation, but it is necessary to choose the scheme and its parameters depending on the need for convergence, and adjust the life span of a matured lymphocyte accordingly.

If the matching function is expensive to compute for less rigid matchings, fast mutation can in some cases be used as a cheap alternative.

The mutation scheme can be selected so that AIS resembles genetic algorithms (GA) without *crossover*, so the lymphocytes perform optimization when converging on their betters.

2.1.7 Self-adaption

A number of parameters and topologies are possible for different parts of the AIS. Making these self-adaptive will push the empirical decision-making one or more levels, but may cause less transparent cause-effect relationships that could ultimately lead to erroneous - instead of improved - classification of pathogens.

It is noted that some of these empirical problems are known from other aspects of computer science, for example the choice of what memory-lymphocyte to throw away when a new one is created (e.g. least-recently-used), is very similar to the problem of cache policies that try to limit thrashing[2].

3 Anomaly Detection Applications

3.1 Network Intrusion Detection

Using AIS for detecting intrusion attempts in a network is described in [1]. Their implementation uses information from TCP packets, including the addresses and ports of source and destination. Their system relies on human interaction for costimulation and response, but was able to detect simulated intrusion attempts (mainly probing) based on real occurences.

3.2 Surface Defect Inspection

Surface defects in textures are recognized quite successfully in [4] by the use of AIS to optimize parameters in socalled *Gabor* filters. An iterative training

[2]Recurring cache usage with frequency lower than that accomodated by the length of the cache, causing it to always just miss.

algorithm is used before the filters are applied to the textures, resulting in selection of the best fit lymphocytes (filter parameters).

3.3 Hardware Failure

A finite state machine (FSM) is monitored by an AIS for invalid transitions and states in [3]. A *greedy* generation of lymphocytes is used to create optimal diversity, enabling good coverage of the *non-self* space when using shorter match lengths. Although the basic idea is intruiging for important real-time applications such as medical equipment, the article notes that the AIS is seemingly more complicated than the FSM that it monitors.

3.4 Copy Protection

A potentially powerful usage of AIS is for copy protection of commercial software. It is however rather different from the other applications depicted here, in that they are of a *monitory* nature, in which the lymphocytes are more or less isolated from the pathogens. In copy protection, the malicious user has access to the inner workings of the lymphocytes themselves. The expected power arises from the fact that the copy protection is then complex and adapts to the behaviour of the malicious user, who will (perhaps) find this *feedback* akin to a Gordian knot.

Technically yet another interesting problem arises, because a database of normal behaviour (*self*) should be compiled into the program, and is therefore again subject to manipulation.

3.5 Spectrum Analysis

A chicken-farmer who has a problem with foxes could train his AIS while guarding the chickens himself. The AIS would be specialized to monitor the frequency spectrum of a real-time sound-recording. The matching of lymphocyte with pathogen would then recognize abnormal sounds - such as a fox causing turmoil.

Perhaps a more plausible use would be in automated chemical analysis for industrial and medical applications.

3.6 Target Identification

Friend or foe, *self* or *non-self*. Negative selection seems tailored to military purposes: First it is able to recognize the enemy without having seen it before (unless of course the enemy looks and behaves just like friends), and once recognized as a foe it will be memorized as such, furthermore *friendly fire* could be limited because of the classification.

Costimulation could be a network of sensors signaling whether damage has been sustained - for an entire nation this is similar to the triggering mechanism of the doomsday-device in *Dr. Strangelove*.

4 Thoughts On Soft Computing

The viability of soft computing stems from its ability to solve hard (e.g. NP-complete) problems quickly, and by providing simple, generalized algorithmic frameworks for complex decision making, or from extracting meaning from noisy data, etc. The success however, seems to depend on tricks and empirical magic. Eventually, it will be useful to identify the weakness precisely and improve upon it in a scientific way.

A soft computing algorithm is strictly better than its deterministic counterpart if:

- The deterministic algorithm can solve the problem optimally in t timeunits, then so must the soft computing algorithm.
- The probability of the soft computing algorithm failing to find the optimal solution must rival the probability of hardware failure.
- The probability that an optimal solution will be found by the soft computing algorithm in less than t timeunits is greater than 0.

Perhaps the *dire* $P = NP$ question becomes less interesting the more soft computing bridges these probabilistic gaps.

5 Thoughts On Evolution

What defines evolution? Is it strictly biological, chemical, divine or primal? What is the *threshold* when some object can be said to be *evolved*? If such things as language, fashion in clothing, tools, machinery, and computers are not considered part of evolution, are they then part of an isolated and parallel evolution? If this parallel evolution is *boot-strapped* by humans, would it then not be plausible that our evolved gadgets eventually will evolve their own sub-gadgets? And considering the effect our own inventions have on our lives and indeed the entire globe, this tendency of *backpropagation* or *feedback* would surely also emerge from the sub-sub-layers of evolutions.

It is then hard to distinguish whether computers (or whatever the stepping stone to the next evolutionary twist will be) are part of the same evolution as we are. One could curiously ask: If evolution had an immune system, would it characterize computers as *self* or *non-self*? Harmless or harmful?

References

[1] Architecture for an Artificial Immune System
Steven A. Hofmeyer, Stephanie Forrest
Dept. Computer Science, University of New Mexico

[2] Function Optimization By The Immune Metaphor
Slawomir T. Wierzchon
Inst. Computer Science, Polish Academy of Sciences

[3] The Architecture for a Hardware Immune System
D.W. Bradley, A. M. Tyrrell
Dept. of Electronics, University of York

[4] Surface Defect Inspect: an Artificial Immune Approach
Hong Zheng, Saeid Nahavandi
School of Engineering and Technology, Daekin University