On AIRS and Clonal Selection for Machine Learning

Chris McEwan and Emma Hart

Napier University, Edinburgh, Scotland
{c.mcewan, e.hart}@napier.ac.uk

Abstract. AIRS is an immune-inspired supervised learning algorithm that has been shown to perform competitively on some common datasets. Previous analysis of the algorithm consists almost exclusively of empirical benchmarks and the reason for its success remains somewhat speculative. In this paper, we decouple the statistical and immunological aspects of AIRS and consider their merits individually. This perspective allows us to clarifying why AIRS performs as it does and identify deficiencies that leave AIRS lacking. A comparison with Radial Basis Functions suggests that each may have something to offer the other.

1 Introduction

The Artificial Immune Recognition System (AIRS) was proposed by Watkins [23, 22], extending a lineage of immune-inspired work on unsupervised learning to the supervised domain. Initial results were favourable and these results have been reproduced several times by different authors [14]. To this day, AIRS remains one of the most widely studied and applied AIS in pattern classification. This popularity is further encouraged by a publicly available plug-in[1] for the Weka Data Mining environment [24].

Theoretical insight into *why* AIRS performs as it does remains scant. Several hypotheses have been tentatively offered in the literature [7, 12], but did not reach any definite conclusions. These studies tend to lack the rigour typical of machine learning literature. It is from this perspective that we attempt to approach AIRS in this paper.

The paper unfolds as follows: In Sect. 2, after introducing AIRS, an experiment with a simplified derivative algorithm validates some concerns and allow us to work back towards the full AIRS algorithm, bringing its main functionality into focus. This then points to additional issues that we explore and verify experimentally, building a rather complete technical picture of AIRS as a learning algorithm. In Sect. 3, we propose that some of these issues can be rectified by exploiting aspects of a more classical approach, Radial Basis Functions. By comparison and experiment, we demonstrate that each may have something to offer the other. This leads to a more general comparison between clonal selection and classical iterative descent algorithms. We conclude in Sect. 4 by taking a broader view toward future work.

[1] `http://www.artificial-immune-systems.org/algorithms.shtml`

2 AIRS

AIRS is an unweighted k-nearest neighbour classifier. The immunological inspiration contributes to how the algorithm is trained to develop a repertoire of “memory cells” which is based, loosely, on Burnet’s Clonal Selection theory [9]. We outline this process in Algorithm (1): memory cells (prototypes) stimulated by antigen (data) proliferate and mutate; these stimulated cells and their progeny compete under selective pressure for continued stimulation, resulting in only the fittest being aggregated into the repertoire. It is this repertoire that is used for classification, in proxy of the full data set.

```
memory = initialiseRandomRepertoire()
for (x,y) in trainingData do
    best = memory.bestMatchingCell(x,y)
    pool = [best]
    while pool.avgStimulation() < threshold do
        for cell in pool do
            pool.add(cell.mutations(cell.stimulation(x)))
        end
        pool.cellsCompeteForResources()
    end
    fit = pool.fittestCell()
    if fit > best then
        memory.add(fit)
        if $||best - fit||_2 < \epsilon$ then
            memory.remove(best)
        end
    end
end
```

Algorithm 1: Pseudo-code for the AIRS training procedure

Variations of this general strategy abound in the Pattern Recognition literature, thus, the immunological component is the key point of distinction of AIRS as an algorithm. Typically, clonal selection is applied in a (black-box) optimisation setting: each cell represents a solution; their “stimulation” reflecting an objective function value. This variant on evolutionary algorithms has some practical benefits that result from the immune-system’s particular *hyper-mutation* process: there is no arbitrary parental cross-over in generating new solutions; and mutation in inverse proportion to stimulation promotes poor solutions to explore the space (few large mutations) and exploits better solutions which more conservatively approach their local optima (via many small mutations).

This population-based stochastic hill-climbing strategy has proven to be most effective in complex multimodel and multi-objective optimisation settings [6]. Certainly, finding the locations of prototypes in our data’s feature space can be cast as an optimisation problem. This is vaguely implied in the AIRS literature, but a simple argument shows that this implication is misleading.

2.1 Clonal selection in the learning context

Recall, AIRS takes a "one-shot" pass through the training set, responding to each datum individually. Each prototype μ_k receives stimulation as an inverse function of distance[2] from the datum x_i. This stimulation parameterises the quantity of mutants produced and the magnitude of mutation suffered.

We question the validity of applying black-box stochastic optimisation in a unimodal setting where stimulation and location have a monotonic relationship. Quite simply, the algorithm "knows" the direction and distance from its current optimum – this is used directly in calculating stimulation – and so random search appears to serve no valid purpose. As illustrated in Fig. (1) each best matching prototype μ_t has a surrounding region of potential mutations (solid circle) with an obvious optimal step μ_{t+1}. Over half of the potential mutations (shaded region) will be *a priori* less fit than the parent.

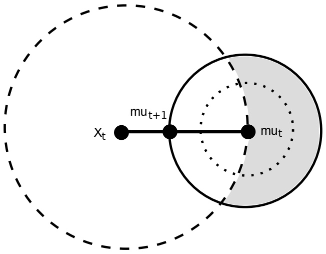

Fig. 1. A schematic representation of the stochastic search procedure for AIRS. There is a trivial (one generation) optimum μ_{t+1} easily derived from the same information used to calculate stimulation. Further, over half the potential mutations of a prototype μ_t will necessarily be less fit.

One might reason that the benefit of this stochasticity may be to overcome AIRS' necessarily myopic nature: rather than directly chasing immediate short-term optima, some random noise allows the algorithm to average out movements without averaging across the data (which should be inaccessible in batch form). While attractive, this justification is heavily contradicted by the implementation. The algorithm has an overly elitist selection criteria: only the best matching memory cell initiates a response, and only the best matching mutant becomes a candidate memory cell. Further, the generation of separate mutation pools

[2] Specifically $1 - ||\widehat{\mu_k - x_i}||_2$ where the hat represents a "normalised" Euclidean distance. We will have more to say on this later.

per datum, from which only the best candidate can survive, does not favour retaining mutants that may still prove beneficial in hindsight.

A simple experiment clarifies. We completely remove the immunological component from AIRS, replacing it with a trivial, deterministic update which we dub AIRS$^-$ (see Alg. 2). Here, we simply choose a single candidate memory cell exactly halfway between the datum and the best matching memory cell[3].

```
memory = initialiseRandomRepertoire()
for (x,y) in trainingData do
    best = memory.bestMatchingCell(x,y)
    fit = 0.5 * (best + x)
    memory.add(fit)
    if $||fit - best||_2 < \epsilon$ then
        memory.remove(best)
    end
end
```

Algorithm 2: Pseudo-code for AIRS$^-$. The optimal (one step) candidate is chosen deterministically, rather than via AIRS' many rounds of mutation and resource competition.

The performance differences for several datasets are reported in Table 1. The figures validate our concern: *the clonal selection phase of AIRS has almost no positive effect on classifiers performance.* Not only is the stochastic search unnecessary, it can be detrimental. AIRS$^-$ performs significantly better on all high-dimensional datasets. Indeed, on the *newsgroup* dataset AIRS performs no better than random guessing. For comparison, on the same task 3-nearest neighbour achieves 75% accuracy, linear regression 80% and Multinomial Naive Bayes 97%. This suggests that the degrees of freedom in high-dimensional space seriously impede the stochastic search procedure; which is intuitive, but contrary to previous claims.

2.2 From AIRS$^-$ back to AIRS

In deriving the deterministic update rule for AIRS$^-$ we simply performed the logical extreme of what AIRS was indirectly attempting by blind search. We can improve our understanding of AIRS if we pursue this idea some more. Recall,

[3] In an earlier experiment, rather than replace the evolutionary search phase we allowed the deterministically chosen candidate to compete with AIRS' mutants, but did not allow the candidate to mutate new solutions. In this regime, it is possible that the stochastic search could mutate past the optimal (one step) midpoint, getting even closer to the antigen. In fact, this almost never occurred – our deterministically selected midpoint was, almost without exception, selected as the fittest candidate for each training instance.

	dimension	AIRS	AIRS$^-$
elements	2	74.35 ±7.29	71.95 ±7.72
iris	4	94.67 ±5.36	94.47 ±6.34
balance	5	80.93 ±4.11*	77.36 ±4.83
diabetes	8	71.60 ±4.40*	69.45 ±4.98
breastcancer	9	96.28 ±2.35	96.35 ±2.19
heart-statlog	13	78.15 ±8.63	77.11 ±7.34
vehicle	18	62.05 ±4.89*	57.06 ±6.04
segment	19	88.21 ±2.48*	83.79 ±2.91
ionosphere	34	84.44 ±5.18	89.66 ±5.39 *
sonar	60	67.03 ±11.60	84.58 ±7.86 *
newsgroup	3783	51.35 ±4.60	78.87 ±14.05 *
* significant at p-value of 0.001			

Table 1. Performance comparison of AIRS and our deterministic derivative. Experiments were performed in Weka using the default algorithm parameters, 10-fold stratified cross-validation and a paired T-test. Most datasets are standard UCI benchmark problems. *Elements* is a synthetic mixture of gaussians taken from [8] which is designed, for pedagogical reasons, to favour neither local nor global learning methods. *Newsgroups* is a two-class classification of determining `comp.graphics` from `alt.atheism` posts using a subset of the 20 Newsgroup dataset.

that for every datum the evolved candidate lies (somewhat) between the datum and the previously closest prototype. In AIRS$^-$ we used the update rule

$$\mu_{t+1} = \gamma(x_t + \mu_t) \tag{1}$$

where μ_{t+1} is the candidate, μ_t was the previous best matching prototype and $\gamma = 0.5$ was the distance to the boundary of the mutation region. Some trivial manipulation allows us to express (1) as

$$\mu_{t+1} = \mu_t + \gamma(x_t - \mu_t) \tag{2}$$

which is also the formula for an exponentially weighted moving average. Rather than holding γ fixed, we can incorporate AIRS' mutation as a function of stimulation, by allowing γ to decrease as stimulation increases

$$\mu_{t+1} = (1 - \gamma)\mu_t + \gamma x_t \tag{3}$$

which is simply a linear interpolation between prototype and datum. The only significant difference between this and AIRS is that AIRS will take many indirect steps over several generations, before selecting the "best" found.

Now, Eq. (2) and (3) are *exactly* the update rule for MacQueen's 1967 online k-means algorithm [11]. But whereas K-means explicitly moves μ_t to μ_{t+1}, AIRS keeps one or both depending on their mutual pairwise distance. Also, k-means will monotonically decrease γ over time, ensuring convergence of centroid locations; in contrast, AIRS employs a datum-specific value of γ based on pairwise distance. We now address any contribution of these differences in AIRS.

2.3 Representational power of the AIRS repertoire

Given the previous analysis, we can see that the repertoire of memory cells in AIRS are a distorted snapshot of the trajectory of a moving average – distorted, because the direction and magnitude of movements are stochastic, undirected and unconstrained.

Based on this observation, we hypothesise that, though smaller in size, the AIRS repertoire does *not* compress or otherwise extract meaningful structure from the original dataset. We validate this claim by comparing the sum of squared distances between data and their closest memory cell, against that of k-means with the same number of centroids as AIRS memory cells (see Table 2). For non-trivial datasets, AIRS is far from the local optima found by k-means. Indeed, we can find the value $\hat{k}$ for k-means that produces the same performance as AIRS. It is apparent that a significantly larger amount of compression is possible than is achieved by AIRS.

	k (memory)	AIRS	k-means	$\hat{k}$
iris	47	1.10	0.768	20
balance	295	16.93	13.5	225
diabetes	407	22.81	8.028	125
breastcancer	209	55.22	28.0	100
heart-statlog	209	108.46	9.036	20
vehicle	336	92.50	23.284	25
segment	219	135.81	51.81	45
ionosphere	145	410.66	94.86	12
sonar	143	420.04	38.679	3

Table 2. The within-cluster squared distances for AIRS and k-means using the same number of centroids as AIRS' memory cells. The value $\hat{k}$ is the number of k-means required to produce the same performance as AIRS. This tends to be dramatically lower than the number of AIRS memory cells, reinforcing that AIRS' repertoire, though smaller than the dataset, has not extracted meaningful structure.

In Fig. (2) we illustrate this effect for the 2-dimensional *elements* dataset, on which AIRS performed reasonably. As Table (2) shows, the effect is less pronounced in low dimensions, but we are limited by what can be visualised. It is still clear from inspection that the density of the repertoire does not reflect the density of the data. Indeed, the repertoire appears to be uniformly spread within the same bounding region as the data. A similar result has been demonstrated by Timmis and Stibor for the algorithm *ai-Net* [19]. Although the details of both algorithms are quite different, AIRS also suffers from the same problem: when deciding if the candidate should replace the original prototype, inflicting a fixed threshold (based on mean pairwise distance) makes it is impossible for the algorithm to represent densities at a finer granularity than that threshold. Further, the mutual exclusion between prototype and candidate precludes any compromise by selecting, say, an intermediate representation such as an average between both.

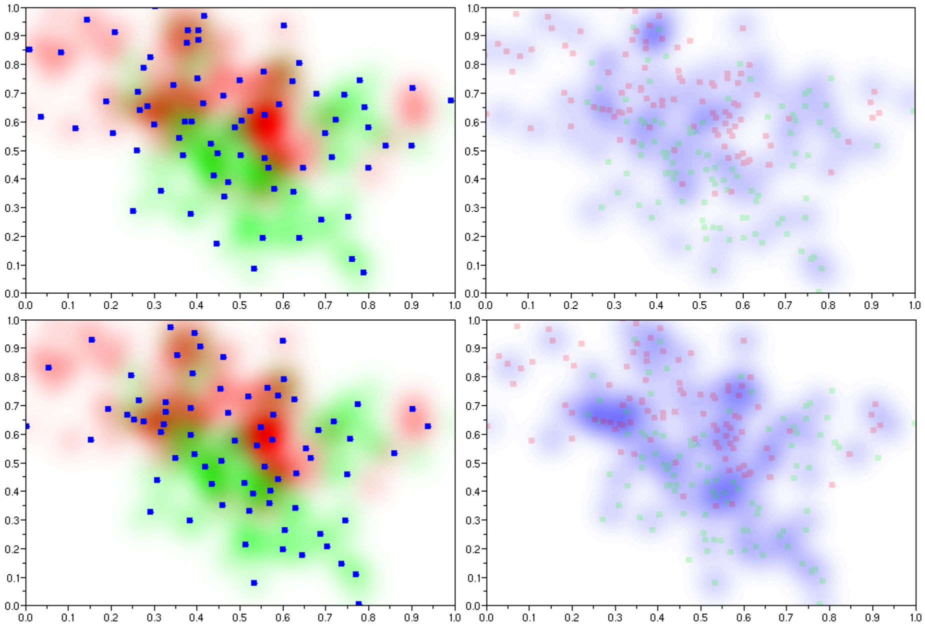

Fig. 2. AIRS memory repertoire for the *elements* dataset (top) compared to the same number of k-means (bottom). The left column illustrates the repertoire/means (blue) superimposed over the class density of the training data (red and green gradients with dark regions representing $p(+) \approx p(-)$). It is clear from inspection that the AIRS repertoire does not follow the density of the data. This is emphasised in the right column, where we illustrate the density of the repertoire/means with the dataset superimposed. The AIRS repertoire appears to have a weak uniform coverage compared to k-means. In higher dimensions, these effects would be much more pronounced, as demonstrated quantitatively in Table (2).

In learning, in order to improve compression, generalisation and discrimination it is necessary to control the granularity of density representation: coarse in coherent, homogeneous regions; fine in ambiguous regions near the decision boundary. AIRS is limited in what it can achieve here. All prototypes share a common pairwise distance constraint and violations are resolved simplistically, with no regard for the engineering goal (or relevant biological dynamics).

2.4 Discriminatory power of the AIRS repertoire

Following training, to classify data AIRS takes an unweighted majority vote amongst the prototypes. This produces very coarse decision boundaries and it is apparent that there is a lot of additional information that AIRS is ignoring. Some form of weighting (e.g. by prototype stimulation to test datum and prototype training fitness) would likely be beneficial. A previous investigation lends some support to this idea [12].

It is worth making entirely explicit that AIRS is essentially a *generative model*: it is an unsupervised learning process repeated in C_i separate compartments; one for each class. The classification decision can be easily interpreted as asking for the compartment that would be most likely to have generated the test instance. However, unlike a generative model, AIRS makes no use of (anything equivalent to) prior probability in either determining the most representative class *or* the most appropriate prototype to select as the best. If the class distribution is skewed, the former will significantly influence AIRS' accuracy. The latter reduces AIRS' choice of "best" to simply closest, rather than a more general criteria of "fittest" developed during the training period.

AIRS is further limited because there is no feedback between data or prototypes of different class compartments. As such, the training process is blind to any ambiguity in the regions where complementary prototypes overlap: the most important regions for classification.

2.5 One-Shot or not?

Prior to commencing Alg. (1) AIRS performs two relatively expensive initialisation procedures. Together, these procedures are $O(nm)$ and $O(nm^2)$, where m is the size of the dataset and n the dimensionality. We suspect this initialisation process is largely responsible for AIRS' often touted "out of the box" performance. However, such convenience is not without cost.

The $O(nm^2)$ step is the computation of an internal parameter – the mean pairwise affinity – which is used as a threshold distance to decide whether a candidate should replace its parent memory cell. In Sect. (2.3) we demonstrated the negative effects of using a uniform fixed threshold. The $O(nm)$ procedure is a "min-max" attribute normalisation. All data are rescaled and translated to lie in a unit bounding box, which simplifies logic by bounding affinity values and legal mutations. However, by computing these bounds at initialisation, AIRS cannot continually learn, as claimed, as such bounds do not remain valid – even for hold-out test data. Further, such normalisation largely presumes a Euclidean distance metric and does not generalise well. In short, this is arbitrary preprocessing, best left to the practitioner. Currently, the internals of AIRS are unnecessarily coupled with this particular initialisation procedure.

Though only polynomial in time, these costs further undermine any practical value in the claim of "one-shot" learning. Both historical and contemporary interest in online learning has been largely driven by its linear-time, fixed-space computational costs – e.g. learning from streaming or massive datasets. In these contexts, an initial linear scan or pairwise comparison is highly undesirable, if not impossible. The current design of AIRS assumes a batch/online compromise that suits neither situation: the computational cost of a batch algorithm and the learning restrictions of a one-shot algorithm.

3 Radial Basis Functions

Having uncovered some theoretical and practical issues with AIRS, we look to something more statistically solid on which to motivate and justify any novel immune-inspired deviations. Radial Basis Functions (RBF) [8, 3] present a simple and elegant compromise between the trade-offs inherent in global (e.g. least squares) and local (e.g. k-nearest neighbour) methods of learning. These trade-off are well documented and we will not labour over them here. An RBF classifies a data point $\hat{x}$ as

$$\begin{aligned}\hat{y} &= f(\hat{x}) \\ &= \sum_{i=1}^{k} \alpha_i \mathcal{K}(c_i, \hat{x}) \\ &= \sum_{i=1}^{k} \alpha_i \exp(-\beta_i ||\hat{x} - c_i||_2^2)\end{aligned}$$

where k is the number of c_i kernel centres (i.e. prototypes), $\mathcal{K}$ is a symmetric distance function parameterised by bandwidth β_i, and α_i are the weights of each prototype, to be found by training.

How the prototypes are chosen is quite arbitrary, though a common approach is to perform a k-means clustering of the data, prior to the supervised learning stage[4]. K-means converges on a local optima of minimising the sum of squared distances between prototypes and their assigned data-points

$$\operatorname*{argmin}_{\mu_1 \dots \mu_k} \sum_{i}^{k} \sum_{x_j \in C_i} ||x_j - \mu_i||_2^2$$

One can certainly question the validity of any optimisation criterion; the only point we wish make here is that there *is* a criterion. However, choosing the correct value for k is somewhat more troublesome. Further, the algorithm must be run several times as the quality of local optima depends on the initial (often randomised) placement of prototypes. Further still, this is typically a batch process, scaling poorly in the size of the dataset.

Regardless, assuming appropriate prototypes the RBF represents each data-point as k features – the distances from each of the k prototypes. The method of least-squares is then employed in this reduced space to solve for $\alpha = \hat{X}^{+} y$ where $\hat{X}^{+}$ is the pseudo-inverse of the transformed training data $\hat{X}_{ij} = \mathcal{K}(c_i, x_j)$. The elegance of this approach is two-fold: During training, the computational burden of a global least-squares solution is eased by reducing the dimensionality. During testing, performance is improved by avoiding lazy-learning. In both cases, the kernel function incorporates beneficial, non-linear locality.

[4] More generally, one can fit a finite mixture model with the EM Algorithm [13]. RBF are essentially unnormalised, symmetric Gaussian mixtures.

3.1 A Comparison between RBF and AIRS

Though the details are somewhat different, there is an obvious high-level similarity in both approaches: *find the best positions for prototypes that can act as a proxy for the full training data and the full feature set.* We now highlight the key differences.

- **Training:** The RBF has a well-defined optimisation criteria, although there is no well-defined manner to choose k. In contrast, AIRS (and brethren) aim to regulate the number of prototypes, but typically have no wider notion of optimally with which to drive evolution. AIRS implicitly partitions the repertoire into classes and fits prototypes to each class. Conversely, the choice of prototype placement for an RBF is *unsupervised*; supervisory information has its influence in the least squares solution for α.
- **Testing:** The RBF uses a combination of the optimal α, the chosen kernel $\mathcal{K}$ and bandwidth β to arrive at a classification decision. Conversely, AIRS simply chooses the partition with the majority of k matching prototypes.
- **Updating:** The cost of having optimal weights is that there is no efficient manner to update an RBF model, without re-computing and inverting $\hat{X}$. Because AIRS is a lazy-learner, it is, in principle, more suitable to update and adapt during execution.
- **Data Access:** A critical difference between both is that RBF uses the full dataset to fit prototypes, whereas AIRS treats data sequentially. One-shot learning may be a desirable feature to retain, but as we have demonstrated above, some significant changes would be required.

We propose that each method has something to offer the other. The RBF has a well defined optimisation criteria, a more elegant approach to handling distances, and a more sophisticated weighted decision process. AIRS can naturally perform multiclass classification, has the potential of deriving its own k per class and an inherent, if poorly utilised, capacity for adaptive updating.

We illustrate the potential for contribution in Table (3) with a comparison between AIRS and RBF fit by k-means. This comparison is not entirely fair, as the RBF was fit in a batch setting. Although the RBF benefits from random access to all data, we wish to emphasise the virtues of a higher-level optimisation criteria; the capacity to generalise coherent and particularise ambiguous regions using variable bandwidths; and a weighted decision process. As such, we handicap the RBF to only two centroids per class; it still outperforms AIRS.

Much of the RBF theory cleans up ad-hoc features of AIRS. Note that none of these changes compromise any “immunological metaphor”. On the contrary, in some respects the metaphor is improved by introducing clone populations (weights), weighted responses, and adaptive recognition regions (bandwidths). Indeed, similar ideas have already been explored in the AIS literature, though in somewhat different contexts [1, 2, 20].

	AIRS	RBF (2)
balance	80.93± 4.11	86.18± 3.76 *
breastcancer	96.40± 2.18	96.18± 2.17
diabetes	71.60± 4.40	74.06± 4.93 *
heart-statlog	78.15± 8.63	83.11± 6.50 *
ionosphere	85.53± 5.51	91.74± 4.62 *
iris	94.67± 5.36	96.00± 4.44 *
segment	88.21± 2.48 *	87.32± 2.15
sonar	67.03±11.60	72.62± 9.91 *
vehicle	62.05± 4.89	65.34± 4.32 *
elements	69.85±10.69	73.80± 10.28 *
* significant at p-value of 0.05		

Table 3. Classification accuracy comparison of AIRS and Radial Basis Functions. The RBF is handicapped to only two prototypes per class, compared to the AIRS repertoire size for the same datasets in Table (2).

3.2 A Comparison between Clonal Selection and Iterative Descent

In the learning context, classical iterative descent algorithms, such as k-means and the EM Algorithm, largely embody the "immune principles" that drive AIRS – *clonal selection* (assign responsibility for data amongst prototypes) and *affinity maturation* (optimise prototypes based on assigned data). Thus, although clonal selection and affinity maturation may be a *necessary* element of immune-inspired learning algorithms (in that they approach essential functionality), they do not appear *sufficient* to determine novelty or value.

It would be remiss to not point out that the greedy nature of classical iterative descent algorithms is not lost on the statistical literature. There are many attempts to make these algorithms more adaptive and global in their optimisation ([13, Chapter 6] [18, 17, 4]). Evolutionary approaches have also contributed to this domain, though to our knowledge, the evolutionary search typically encodes *all* mixture parameters in a single genotype, and is then used to find a better initial configuration for iterative descent (see e.g. [10]). This is quite different from AIS learning algorithms, where each clone is, essentially, a mixture component and the emphasis tends to be on continual adaptation and internal dynamics of the repertoire.

By removing these internal dynamics, trivialising the fitness landscape and using pairwise distance as a proxy for an overarching objective function, AIRS leaves stochastic search with no competitive advantage over traditional iterative descent. These issues can all be addressed. However, clonal selection alone seems unlikely to induce convincing immune-like behaviour. Building upon clonal selection appears necessary if AIRS is to offer more than a global optimisation method for a prototype-based learning algorithm. Acknowledging the contributions of, and any similarities to, classical algorithms seems the best way to fruitfully direct novel immune-inspired research in this mature field.

4 Conclusion and Future Work

This paper reinforces warnings that one cannot work exclusively within the immunological metaphor [21]. In the present case, the metaphor has obscured contradictory design decisions and functional omissions with regard to the problem domain. Although these deficiencies may appear manifold, many are elementary and quite straightforward to address. The exception being stochastic search on a uni-modal landscape, which is neither theoretically valid, computationally desirable or biologically plausible. We intend to address this in future work, by trading-off complexity in the fitness landscape against scaling independently from the size of the dataset.

When the strictly one-shot requirement on AIRS is relaxed, it begins to resemble many of the idiotypic-network style algorithms that preceded and inspired it – Neal's *meta-stable memory* [15], Timmis' *RAIN* [6], Von Zuben and de Castro's *CLONALG* and *ai-Net* [5, 16]. Because AIRS is still essentially an unsupervised algorithm, run in class-specific partitions, any of these unsupervised AIS could equally be used to determine a repertoire of memory cells. But based on previous experience with some of these algorithms, we suspect that a good deal of what has been discussed with regard to RBF and iterative descent would largely translate to these settings. Rather, we would propose that there may be an opportunity for unifying this lineage of work, by acknowledging (and leveraging) existing research in machine learning and non-parametric statistics. Decoupling the statistical aspects of AIS from the immunological component aids clarity and correctness. Closure on the contribution of clonal selection would clear the way for more focused and sophisticated immunological contributions; that can be transparently motivated and communicated without metaphor.

References

1. Oscar Alonso, Fabio Gonzalez, Fernando Niño, and Juan Galeano. A solution concept for artificial immune networks: A coevolutionary perspective. In *6th International Conference in Artificial Immune Systems*, pages 35–46, 2007.
2. Paul Andrews and Jon Timmis. Adaptable lymphocytes for artificial immune systems. In *7th International Conference in Artificial Immune Systems*, pages 376–386. Springer, 2008.
3. Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.
4. Olivier Cappe; and Eric Moulines. Online em algorithm for latent data models. *Submitted to the Journal of the Royal Statistical Society Series B*, Dec 2008.
5. L. N. de Castro and F. J. Von Zuben. The clonal selection algorithm with engineering applications. In *GECCO'00, Workshop on Artificial Immune Systems and Their Applications*, 2000.
6. Leandro N. De Castro and Jonathan Timmis. *Artificial Immune Systems: A New Computational Intelligence Approach.* Springer Verlag London, 2002.
7. D. E. Goodman, L. Boggess, and A. Watkins. An investigation into the source of power for airs, an artificial immune classification system. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 3, pages 1678–1683 vol.3, 2003.

8. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.
9. Charles A. Janeway, Paul Travers, Mark Walport, and Mark Schlomchik. *Immunobiology*. Garland, 2001.
10. Wolfgang Jank. The em algorithm, its randomized implementation and global optimization: Some challenges and opportunities for operations research. In *Perspectives in Operations Research*, pages 367–392. Springer, 2006.
11. J. Macqueen. Some methods for classification and analysis of multivariate observations. In *Proc. Fifth Berkeley Symp. on Math. Statist. and Prob.*, volume 1, pages 281–297, 1967.
12. Gaurav Marwah and Lois Boggess. Artificial immune systems for classification: Some issues. In *1st International Conference in Artificial Immune Systems*, volume 1, pages 149–153. Springer, 2002.
13. Geoffrey J. Mclachlan and Thriyambakam Krishnan. *The EM Algorithm and Extensions*. Wiley-Interscience, 1 edition, November 1996.
14. Lingjun Meng, Peter van der Putten, and Haiyang Wang. A comprehensive benchmark of the artificial immune recognition system (airs). In *Advanced Data Mining and Applications*, pages 575–582. Springer, 2005.
15. Mark Neal. Meta-stable memory in an artificial immune network. In *Lecture Notes in Computer Science, Volume 2787*, 2003.
16. Nunes and F. J. Von Zuben. An evolutionary immune network for data clustering. In *Neural Networks, 2000. Proceedings. Sixth Brazilian Symposium on*, pages 84–89, 2000.
17. Yoram Singer and Manfred K. Warmuth. Batch and on-line parameter estimation of gaussian mixtures based on the joint entropy. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 578–584, Cambridge, MA, USA, 1999. MIT Press.
18. Mingzhou Song and Hongbin Wang. Highly efficient incremental estimation of gaussian mixture models for online data stream clustering. In Kevin L. Priddy, editor, *Intelligent Computing: Theory and Applications III*, volume 5803, pages 174–183. SPIE, 2005.
19. T. Stibor and J. Timmis. An investigation on the compression quality of ainet. In *Foundations of Computational Intelligence, 2007. FOCI 2007. IEEE Symposium on*, pages 495–502, 2007.
20. Thomas Stibor. Discriminating self from non-self with finite mixtures of multivariate bernoulli distributions. In *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 127–134, New York, NY, USA, 2008. ACM.
21. J. Timmis, A. Hone, T. Stibor, and E. Clark. Theoretical advances in artificial immune systems. *Theor. Comput. Sci.*, 403(1):11–32, 2008.
22. Andrew Watkins, Jon Timmis, and Lois Boggess. Artificial immune recognition system (airs): An immune-inspired supervised learning algorithm. *Genetic Programming and Evolvable Machines*, 5(3):291–317, 2004.
23. Andrew B. Watkins. Airs: A resource limited artificial immune classifier. Master's thesis, Mississippi State University, MS. USA, 2001.
24. Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, June 2005.