The Application of Antigenic Search Techniques to Time Series Forecasting

Ian Nunn
School of Computer Science, Carleton University
Ottawa, Canada
ginunn@digitaldoor.net

Tony White
School of Computer Science, Carleton University
Ottawa, Canada
arpwhite@scs.carleton.ca

ABSTRACT

Time series have been a major topic of interest and analysis for hundreds of years, with forecasting a central problem. A large body of analysis techniques has been developed, particularly from methods in statistics and signal processing. Evolutionary techniques have only recently have been applied to time series problems. To date, applications of artificial immune system (AIS) techniques have been in the area of anomaly detection. In this paper we apply AIS techniques to the forecasting problem. We characterize a class of search algorithms we call antigenic search and show their ability to give a good forecast of next elements in series generated from Mackey-Glass and Lorenz equations.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*dynamic programming*

General Terms

Algorithms, Experimentation

Keywords

artificial immune systems, time series, forecasting, antigenic search

1. INTRODUCTION

A *time series* is a sequence of data collected from some system by sampling a system property, usually at regular time intervals. The analysis of time series has a long and rich history with recorded examples going back more than a millennium [9]. They appear in such diverse fields as astronomy, meteorology, seismology, oceanography, signal processing, plant operations and economics among others. Three primary questions arise. Given an unknown series can we identify it by matching it with known series? Given the past performance of a series, can we determine if the current performance is anomalous? Finally, a question of particular interest in all of the above fields is, given a series, can we forecast the next value or set of values in the sequence? It is this last question that we address in this paper.

1.1 Standard Techniques of Time Series Analysis

A variety of techniques have been developed to analyze time series [9][14]. Most fall into one of two broad classes. The first is based on statistical techniques. The second is based on techniques associated with signal processing and spectrum analysis. Only recently have techniques associated with evolutionary computation begun to be investigated. The first application of AIS techniques appears to be that of Dasgupta and Forrest [1] in 1995.

1.2 The Human Immune System and Antigenic Search

The immune system (IS) performs two roles in the protection of the human body. The first role is anomaly or intrusion detection whereby foreign entities are detected and eliminated. The second is an anticipatory role using a long-term associative memory capability [16] that makes vaccination against novel foreign entities effective.

Dasgupta and Forrest [1][2] have used an anomaly detection algorithm based on the properties associated with this first role to study time series anomalies caused by tool breakage in milling machines. Others have studied the anomaly detection problem with other time series including computer system calls [6][18][15].

The authors are unaware of any algorithm developed to study time series forecasting using an IS metaphor. In this paper we present a class of such algorithms that we term *antigenic search*. We will use the term *antigen* to refer to both entities the IS recognizes as foreign and to points in test data. We will use the term *antibody* to refer to all IS entities that have a detection or match capability, to points in training data and to entities in detector and memory sets.

In Section 2 of this paper we review important characteristics of time series and the IS properties applicable to development of time series analysis algorithms. We identify the shortcomings of IS models for developing time series forecasting algorithms and what extensions are necessary to facilitate their development with antigenic search. In Section 3 we discuss related work and give a detailed description of what antigenic search is and a general implementation. Section 4 presents the results of testing various algorithmic variants against data from Lorentz, Mackey-Glass and uni-

GECCO'05, June 25–29, 2005, Washington, DC, USA.
Copyright 2005 ACM 1-59593-010-8/05/0006 ...$5.00.

form random data sets with an analysis of the results. Section 5 summarizes our results and contributions. Section 6 identifies future work.

2. TIME SERIES AND IMMUNE SYSTEM PROPERTIES

Time series are generated by a wide variety of systems. Most systems of widespread interest are extremely large and complex. The weather is a global system influenced by the oceans and their temperatures and currents, the geography of the planet's land masses and the atmosphere. Some are chaotic in nature and most natural systems exhibit considerable noise in their data. Forecasting then is a hard problem.

2.1 Time Series Characteristics

Characteristics associated with time series include:

1. explicitly represented data points. Usually the dynamics of the underlying generator are unknown;
2. noisy data due to random perturbations;
3. the existence of pattern or cycles in the data with regular (seasonal) period;
4. the existence of cycles of irregular period;
5. differences between corresponding cycles both in amplitude and length;
6. the presence of a trend contribution that may be of polynomial or exponential nature;
7. most series report one-dimensional real valued data; however, multivariate analysis of multiple series may be required in some circumstances.

Because of these complexities, filtering and decomposition techniques are traditionally used to aid analysis and forecasting by smoothing noise and separating out contributions from trend, seasonal and cyclical components. The success of evolutionary approaches to other difficult problems motivates their consideration for time series.

2.2 Immune System Search and Forecasting Characteristics

The IS [3] has a number of characteristics that make it a suitable metaphor for search and forecasting. We summarize these as:

1. explicit representation of entities, useful for evolutionary and population-based techniques based on clonal selection and affinity maturation [3];
2. an ability to divide a search space into 2 partitions commonly denoted as self and non-self, useful for anomaly detection;
3. an ability to create a population of antibodies each capable of recognizing a class of antigen while being incapable of recognizing any of self;
4. an ability statistically, to cover the entire non-self partition over a short period of time [13] (see [4] for AIS coverage techniques);
5. the ability to perform an approximate match with a class of antigen with varying degrees of strength or *affinity* for the members of the class. This is equivalent to performing pattern matching on noisy signals;
6. the development of a separate class of memory entities whose populations are proportionate in size to their affinity for antigen;
7. anticipation of mutation of antigen through clonal diversity caused by somatic hypermutation.

A comparison of IS characteristics with time series characteristics suggests the IS provides a strong metaphorical basis for building algorithms to apply to time series problems.

What the IS lacks for representing time series however, is any mechanism to represent sequence. The IS appears to remember some antigen for long periods of time while forgetting others in a manner unrelated to the order of infection. It has no mechanism to determine or represent the order of infection. Antibody population size might be used as an indicator if it were proportionate to the temporal order of infection instead of the strength of response to infection.

Introducing sequence into the memory aspect of IS models should enable the extension of IS search to modeling series and forecasting. This is what antigenic search is designed to facilitate.

3. ANTIGENIC SEARCH

Common to all AIS algorithms is a representation of an entity in a problem domain by a chromosomal structure, the genes of which may have any numerical or symbolic specification. This structure, expressed as an n-tuple, corresponds to a point in a space and the set of values of its genes represent the position of the entity in the space. We will refer to this space as a *state space*. Other terms used in the literature include shape space and search space.

In antigenic search we make two modifications to standard AIS techniques. The first is to modify a gene by adding two additional components representing velocity and acceleration. This gives a gene the representation (d^0, d^1, d^2) where d^i is the ith derivative. In principle this could be extended to any number of derivatives.

Our second modification is to specify a memory that preserves sequence information. Any computational structure may be used as long as the sequencing information of the underlying time series can be recovered. This extends the standard memory capability of discrete AIS models beyond that of storing individual unconnected entities to include series or sequences of connected entities.

Whereas Dasgupta and Forrest [2] use an implicit representation of sequence as an encoding of the genome of an antibody, we use an explicit representation via a memory model in which each point in the series is discretely presented in a linked fashion. This allows easy representation and comparison of variable length series and subseries.

The use of derivatives not only captures immediate past performance of a system but enables forecasting next state, real-time anomaly detection, and beneficial bias in search and optimization problems.

In this paper we consider the application of antigenic search principles to the specific problem of next element forecasting in a time series. We use real valued genes but any representation might be used as long as a notion of derivative can be specified for it. We note we have not attempted to apply these ideas to immune network models [7].

3.1 Related Work

NASA uses anomaly detection of time series data to identify component failure. Mahoney and Chan [12] have studied anomalous Shuttle valve operation data using derivative-based models. A first derivative approach using piecewise linear splines was investigated but failed to detect data anomalies satisfactorily.

A path-based approach has since been developed using first and second derivatives of their 1-dimensional data to construct points in a 3-dimensional space. They note that n-dimensional data may be used. When they add first and second derivatives they get a $3n$-dimensional space to work in.

Low-pass filters are employed to smooth the data and improve its continuity. Anomalous behavior of a test series is determined by the use of a Euclidean-based distance measure between a test point and a point of a training series summed over all points in the training series and all test points. They rejected immunological-based approaches due to their lack of a suitably "human comprehensible model".

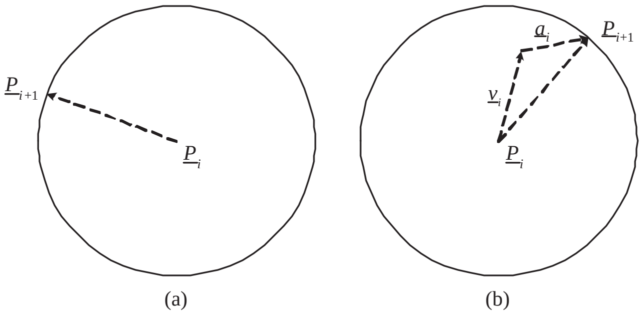
The other piece of work we mention is that of Kennedy et al. on particle swarm optimization (PSO) [10] [11]. This is an evolutionary [8], population-based search technique rather than a time series analysis technique. The relevance to our work lies in its representation of points in a search space by both a position and velocity (first derivative) vectors. The authors use a fitness criterion to guide the search. Each entity in the population adjusts its position and velocity in the search space by the vector difference between its current position and a combination of its personal best position and the population's globally best position to date.

3.2 Features of Antigenic Search

As noted in Section 2.2 the IS has what we termed an anticipatory characteristic. When an antigen is identified the IS produces a series of genetically uniform subpopulations termed *clones* of size proportional to their affinity for the antigen. These we may consider to be memory antibodies. Each clone will have its own area of coverage of state space with the likelihood of considerable overlap with other clones. Should a mutated antigen occupy a point in state space not far from its parent it will likely fall into this area of clonal coverage. In this sense, the IS can anticipate future infection. Smith [17] has used this property to model a time series problem, the efficacy of influenza vaccination strategies.

An antibody extends its region of recognition through its clonal children. If we consider any particular lineage, the successive generation of children effectively maps a search tree through state space. As we have noted, however, the IS has no mechanism to record this lineage. We can confer a stronger ability on an antibody than anticipation by adding velocity and acceleration components. These effectively establish a lineage relationship for successive generations of antibody. In addition, the application of these components to an antibody in state space constitutes making a forecast of its next position or in other terms, the location of a preferred child. Its velocity and acceleration may be expressed at each point as a discrete difference from its previous position. Therefore, each child clone of an antibody with its parental lineage represents a unique path or trajectory in state space, a series in time.

The effect of velocity and acceleration components is shown in Figure 1. The circle shown represents a hypersphere in state space. For normal IS search, Figure 1(a), if the probability distribution of children is centered on the parent's position $\underline{p}_i$, a child has equal likelihood of being created anywhere on this surface, say at $\underline{p}_{i+1}$. With antigenic search, the parent positioned at $\underline{p}_i$ with associated velocity $\underline{v}_i$ and acceleration $\underline{a}_i$ will forecast a preferred position at $\underline{p}_{i+1}$ as shown in Figure 1(b).



(a) (b)

Figure 1: An antibody's movement is state space, (a) no preferential direction and (b) forecast direction.

The first three entities in a time series (three are needed to calculate initial velocity and acceleration components) are sufficient to establish a continuous trajectory in state space. Since this trajectory does not likely represent the training series, the memory must be given an additional capability of storing the actual entities and their sequence.

The training phase of an antigenic search algorithm compares the next test entity's position with the forecast position and uses this feedback to guide the direction of mutation of children. Children are added to memory with a two-way link with their parents. This captures the sequential information of a time series. Any memory antibody has the information to allow both forward and backward traversal of the stored series characteristics. In the training phase of antigenic search, evolutionary selection is not required.

3.3 An Antigenic Search Forecasting Algorithm

In this section we will present the core component of the basic algorithm first, the forecasting component second and the child creation component last.

As a first step in the core component, a forecast is made. Next an antigen is retrieved from the test series and matched against the memory. We use a two threshold approach to recognition. One is a match threshold *matchT* that determines whether recognition of antigen by an antibody would be said to occur. The second is a tighter, sufficiency threshold *suffT* that is used to guide the creation of appropriate antibodies for the memory set. This bears some similarity to the two threshold model of Watkins et al. [19] who use a "stimulation threshold" to terminate training on an antigen.

If the first match with an antibody that occurs is sufficient, nothing further is done with the antigen. If the first match made with an antibody is not sufficient, a child capable of a sufficient match is created and added to the memory set. If no match occurs, a child is created capable of a sufficient match and added to the memory set.

```
Parameters:
matchT: the distance within which a match occurs
suffT: the sufficient distance to qualify an antibody

Initialization:
memSet <- new empty structure;

Processing:
while antigen remains do
    Forecast antigen and report difference;
    exit <- false;
    Get next antigen;
    while memSet not empty and exit false do
        Get next antibody;
        distance <- distance between antigen, antibody;
        if distance <= suffT then
            exit <- true;
        else
            if distance <= matchT then
                Create child;
                exit <- true;
            endif
        endif
    endwhile
    if no memory match occurred then
        Create child;
    endif
endwhile
```

The forecasting aspect of the algorithm is given next. Genes have three components, position, velocity and acceleration shown by an index in parentheses in this example. A forecast is made from the last antibody accessed in the memory set. The position of the forecast antigen is computed from the position, velocity and acceleration associated with this antibody. No velocity or acceleration components are calculated for the forecast since the actual next antigen presented will have no such components to compare against.

```
Initialization:
lastC <- chromosome of last matched antibody;
newC <- initialized chromosome of forecast antigen;

Processing:
for each gene of lastC and newC do
    newCGene(0) <- lastCGene(0) + lastCGene(1)
        + lastCGene(2);
    newCGene(1) <- 0;
    newCGene(2) <- 0;
endfor
forecastAntigen <- Create with newC chromosome;
```

When a new antibody is created, two things happen. Its position in state space is chosen using a mutation rate parameter. Then, its associated velocity and acceleration are calculated using its new position and the position and velocity of its parent. The number of new antibodies created is determined by a reproduction rate parameter. This reproduction process is described by the following pseudocode:

```
Parameters:
suffT: the sufficient distance to qualify an antibody
muteRate: the rate in [0, 1] of mutation;
repRate: the number >= 1 of children produced;

Initialization:
lastC <- chromosome of last matched memory antibody;
agC <- chromosome of the current antigen;
newC <- initialized chromosome of new antibody;

Processing:
for repRate times do
    for each gene of lastC and newC do
        Create new gene using muteRate;
        newCGene(0) <- random value, dist. <= suffT;
        newCGene(1) <- newCGene(0) - lastCGene(0);
        newCGene(2) <- newCGene(1) - lastCGene(1);
    endfor
    newAntibody <- Create with newC chromosome;
    memSet <- Add newAntibody;
endfor
```

The motivation to use a dual threshold model was to provide separate capabilities to conduct search and guide reproduction. A match threshold, *matchT*, guides search and can be set relatively wide. A sufficient threshold, *suffT*, specifies the condition under which an antibody is close enough to an antigen that we won't try and improve it by reproduction or, if reproduction occurs, the criterion for selection of a clone.

The above description of our algorithm imposes no specific structure on the memory set but we assume that whatever is used meets the requirement to retain sequence information. We tested three different memory set structures, a FIFO queue, a LIFO queue and a graph structure. Their characteristics are discussed in the next section.

3.3.1 Memory Structures

A FIFO queue was considered as it produces a memory image of a training series in the order of presentation of antigen since it is traversed in the sequence entries are made. This means a test series can be matched by traversing the memory in the natural order in which antibodies are added. The disadvantage with a serial traversal of this sort is the first match is accepted whereas a better fit may exist further down the queue.

The effect of the dual threshold model is to trap new antigen early in the queue traversal when the match is not sufficient causing a new antibody to be created and added to the memory. With no information to guide choice, intuition would suggest the last antibody created would be a better candidate to test first against the next antigen rather than some other antibody produced by a match in the queue. If such were the case, the last antibody is more likely to produce a sufficient match producing no new antibody whereas an early match near the front would more likely produce one, adding to the memory size.

A LIFO queue allows us to test this intuition by providing a reverse order traversal of the series. A possible disadvantage of this structure occurs when matching a test series: the memory is in the reverse order of what may be desirable. A related disadvantage is that subsequences in the test series that might have earlier memory matches are not identified and will be repeated in the memory set.

With both structures, multiple antibodies with overlapping coverage at the sufficient level can easily occur. To limit memory representation to non-overlapping antibodies a graph structure was created using an unordered list in the following manner. Each antibody added to the list is given a secondary list for storing edges. Antigen are tagged with their index position in the time series they are taken from. An antibody is created and added to the memory for an antigen only if no sufficient match in the list is found. Otherwise, the best sufficient match in the memory is chosen

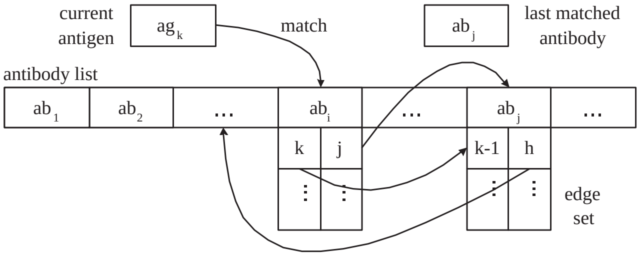


Figure 2: The graph structure used to implement the memory set.

and its edge set augmented with a pair of indexes. One is the index of the antigen that created the current best match. The other is the memory position of the antibody that was the previous best match.

Using Figure 2 as an example, the current antigen ag_k is matched against the entire antibody list. For the best matching antibody ab_i, the index k is placed in its edge set along with the antibody list index j of the last antibody matched, ab_j. This new best match becomes the last best match. To reconstruct the entire sequence, we start from the last best match. The largest antigen index in its edge set is the last entry in the test series. The memory set position associated with it gives the previous best match. This retracing can continue until the first antibody created is reached. Note that any antibody may be referenced a number of times corresponding to the size of its edge set. This structure guarantees no overlapping coverage since new antibodies are added only for antigen that find no match. In this case the new antibody becomes the last matched.

Three algorithmic variants implementing these structures called ***FIFO***, ***LIFO*** and ***Graph*** were investigated. The results are given in Section 4.4.

3.3.2 Reproductive Strategies

We investigated two variants of the algorithm for reproductive techniques. Both implemented memory with a FIFO queue. One mentioned in the last section called the ***FIFO*** variant creates a random mutation within the antigen's sufficient threshold. The other called the ***Exact*** variant creates an antibody as the exact genetic clone of the antigen. Both of these techniques supersede the use of a mutation rate.

As a simplification, we used another parametric constraint, that of a reproduction rate of 1. This avoids the complexity of adding multiple clones to the memory at each stage since only one preferred antibody is needed for making a forecast. If multiple clones are used, then a fitness criterion would be needed to recommend a best one as the forecast.

4. TESTING AND RESULTS

For initial conditions, the velocity for the first antibody created and the acceleration for the first two antibodies created are defined as 0.

MatchT and suffT values are related to the size of the space being covered. The Lorenz data coordinates have the following span: $x \in (-18.8, 21.9)$, $y \in (-24.5, 30.2)$ and $z \in (-1.0, 55.1)$. This gives a rectilinear volume of 1.25×10^5. In contrast, the Mackey-Glass data tested has a linear span of $x \in (0.2, 1.4)$ which is less than the matchT used in some tests on Lorenz data. A value that would give good discrimination in a Lorenz space could completely cover the Mackey-Glass space.

We evaluate the accuracy of a forecast by measuring the Euclidean distance $d_{\mathcal{E}}$ between the forecast antigen and actual antigen. For two points $\overline{x} = (x_1, x_2, \cdots, x_n)$ and $\overline{y} = (y_1, y_2, \cdots, y_n)$ in state space this distance is given as

$$d_{\mathcal{E}} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

We term this distance the *forecast error*. All data sets used are of size 1000 and each experiment processes all 1000 antigen. The error we report is the mean error over the 1000 individual forecasts.

4.1 Forecasting with Lorenz Data

We used a discrete time form of the Lorenz equations

$$\begin{aligned} x_i &= x_{i-1} + ha(y_{i-1} - x_{i-1}) \\ y_i &= y_{i-1} + h(x_{i-1}(b - z_{i-1}) - y_{i-1}) \\ z_i &= z_{i-1} + h(x_{i-1}y_{i-1} - cz_{i-1}) \end{aligned}$$

to calculate our dataset. Parameter settings were $a = 10.0$, $b = 28.0$, $c = 8.0/3.0$ and $h = 0.01$ with an initial value of (1, 2, -1).

4.1.1 Effect of matchT

Using ***Exact***, the effect of varying matchT on accuracy is shown in Figure 3. For this we set suffT = 1.0. First we note how the memory rapidly increases in size to its upper bound corresponding to 1000 antigen. The reason for this is as matchT increases, more and more antigen are trapped early in the sequence by antibodies that can match them, but since the match is not sufficient, a new sufficient antibody will be generated and added to the memory.

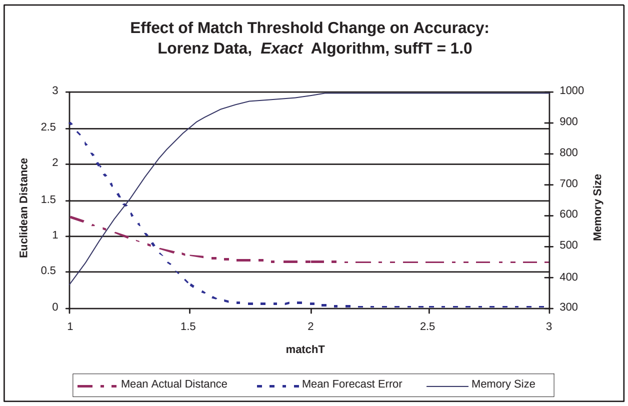


Figure 3: The effect of changing the matchT size on memory size and forecasting error.

A consequence of this is that the last antibody accessed is more often closest in sequence to the previous antigen tested. This generally should offer the best predictive capability of any antibody creating a forecast with least error. This effect we observe as the mean forecast error and note it inversely tracks the memory size.

With smaller memory sizes there is less overlap among antibody coverage so a single antibody will match more antigen. The average of these antibody/antigen distances will be greater than if two antibodies match the same number of antigen, each more closely. The result is seen in the curve on the chart that shows the mean actual distance between a new antigen and the last antibody referenced which corresponds to the the last antigen presented. This distance decreases in an inverse manner to memory size increase. With closer matches, one would expect smaller forecast errors and the two curves decrease in a similar manner with increasing memory size.

The relationship between actual distance and forecast error then is smaller mean error relative to mean actual distance indicates greater forecasting accuracy and better performance.

If matchT is set to 0, the model is effectively a single threshold one based on suffT. Varying suffT produces better forecasts at smaller values as it forces more antibodies to be generated. This in turn results in a closer fit to the training series. A matchT of 2.0 and a suffT of 1.0 produce a low mean forecast error while maintaining good run times. These same parameter values are used in Section 4.1.2 and Section 4.4.

4.1.2 Forecast Error

Figure 4 shows the accuracy of one test using ***Exact*** on Lorentz data. For most of the 1000 iterations of the test the forecast error is small. Closer examination of the data, Figure 5, shows the regions of higher error correspond to regions of rapid change in the data, particularly at the second derivative. The results are listed in Table 1.

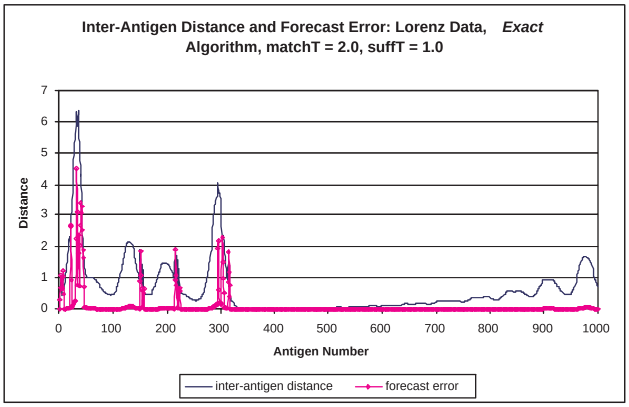


Figure 4: A comparison of the forecasting error and the distance between successive antigen for *Exact* using Lorenz data.

Table 1: Forecast Error and Actual Distance for Lorenz, Mackey-Glass and Uniform Random data using *Exact*

	Lorenz	Mackey-Glass	U. Random
error mean	0.0890508	0.0043713	4.1708794
error stdev	0.3911758	0.0112890	1.6752899
actual mean	0.6562106	0.0289928	1.3192512
actual stdev	0.9059448	0.0199842	0.5098254

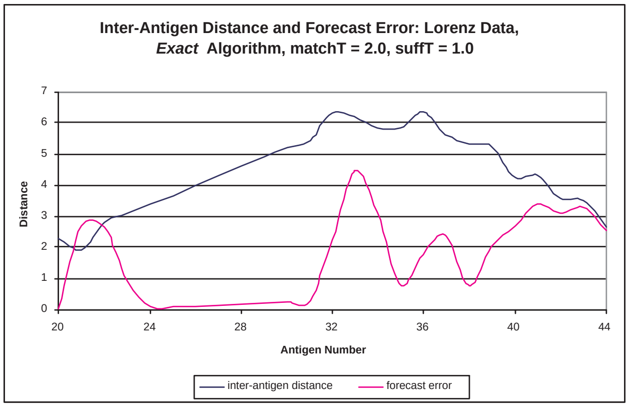


Figure 5: A magnified view of the data for antigen numbers 20 to 44.

4.2 Forecasting with Mackey-Glass Data

The Mackey-Glass time delay differential equation can be written as

$$\frac{dx_t}{dt} = \frac{\beta x_{t-\delta}}{1 + x_{t-\delta}^{\gamma}} - \alpha x_t.$$

Our data is generated with parameter settings $\alpha = 0.1$, $\beta = 0.2$, $\gamma = 10.0$ and $\delta = 17.0$ with an initial value of $x_0 = 1.2$.

Tests similar to those for Lorenz data shown in Figure 3 suggest that a matchT of 0.2 and a suffT of 0.01 produce a low mean error. We repeated the test for values of suffT of 0.1, 0.01, 0.001, 0.001, 0.0001 and 0.00001. These produced memory sizes of 9, 78, 419, 872 and 992. The spikes in forecast error in Figure 6 for suffT = 0.01 that appear larger than the actual antigen/antibody distance recede to much lower levels as suffT is decreased. The results using ***Exact*** are given in Table 1.

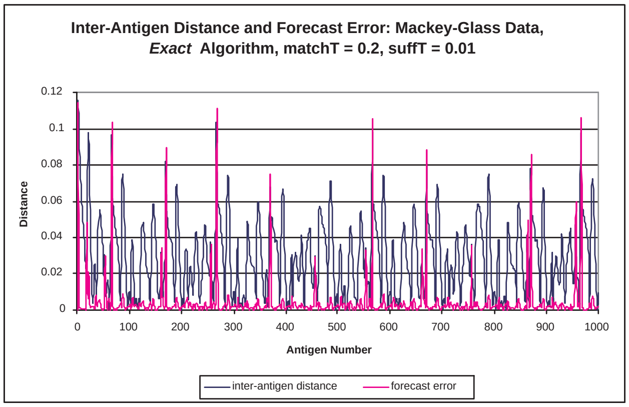


Figure 6: A comparison of the forecasting error and the distance between successive antigen for *Exact* using Mackey-Glass data.

4.3 Forecasting with Uniform Random Data

Lastly we ran a test of our algorithm with random data uniformly distributed on the interval (-1, 1). Figure 7 shows the result of a test using parameter settings of 0.2 for matchT and 0.05 for suffT. The results using ***Exact*** are given in Ta-

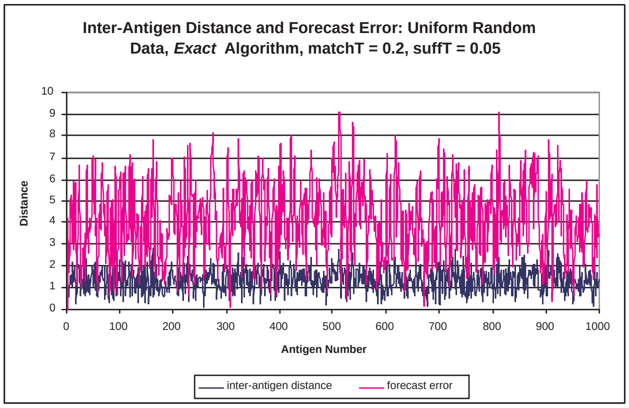


Figure 7: A comparison of the forecasting error and the distance between successive antigen for *Exact* using uniform random data.

ble 1.

For Lorenz and Mackey-Glass data, the mean error relative to the mean actual distance is 14% and 15% respectively. For the uniform random data, it is 316% indicating the algorithm has much worse predictive capability.

A true measure of the goodness of these results depends on an external criterion to compare them against and for which we have no examples. However it is interesting to look at the ratio of the volume of space of a cube of twice the mean error on a side, to the rectilinear volume of space computed in beginning of Section 4. For the Lorenz case, this number is 2.8×10^{-8}. For The 1-dimensional Mackey-Glass data it is 6.2×10^{-3} and for the random data, 72.5. While the error in the first two cases is a small fraction of the size of the problem space, in the random case it is orders of magnitude larger attesting to an expected inability to forecast random data.

4.4 Memory Structure Tests

Tests on the memory structures introduced in Section 3.3.1 are discussed in this section. All tests were performed on Mackey-Glass data with matchT = 0.2 and suffT = 0.01. The results are for 20 runs on each data structure and are shown in Table 2. Since the property of the graph structure is a set of non-overlapping antibodies, the results for 20 runs are identical.

Table 2: Forecast Error for FIFO, LIFO and Graph Data Structures Using Mackey-Glass Data, 20 runs.

	FIFO	LIFO	Graph
mean	0.021916631	0.29319738	0.056521207
stdev	0.000509427	0.001183307	0.054286758
mem size	980	802	76

For a FIFO queue, growth of the memory was observed to be linear at almost a 1:1 ratio at 980:1000, antibody:antigen. Its performance for forecasting was significantly better than the other structures.

For the LIFO queue with 802 antibodies generated, the memory saving over the FIFO case was about 19% but the accuracy was considerably less.

The graph structure had the lowest memory size at 76 antibodies. With no no overlap in coverage, this is probably the lower limit on memory size for the parameter settings chosen. The cost of this improvement is a decrease in forecasting accuracy over the FIFO case of about 2.5 times.

As a general observation, forecasting accuracy increases as more antibodies are added to the memory.

4.5 Comparison with Related Work

A requirement of single event anomaly detection is that a binary partition can be created in state space either explicitly by representing or enumerating the members of one partition or implicitly by a function or relation that describes all the members of one partition. This partition which we call a $\mathcal{K}$*-partition* may in some cases be implemented as a memory structure. It may represent either the self or non-self sets of AIS methodology [5]. If a series of events are to be examined for anomalies as is the case with a time series, sequence must be added to the space in the form of an order relation with explicit representation (connection). Finally, if forecasting is to be implemented, a capability must be added that allows a change in position to be computed in state space.

In PSO [10] there is no mechanism for partitioning state space or representing the sequence of entities. Although it has a limited dynamic in the form of position and velocity information, in its current form it is unsuitable for analyzing time series.

The work of Mahoney and Chan [12] is closest to ours having both a sequence representation of time series data in their path and box model, and a dynamic in the first and second derivative that could be applied to forecasting. Their use of these derivatives as ordinates of a point in state space is different from our representation of them as information associated with a point represented in the space by position only. This gives a more complex representation, particularly with respect to distance measures. It is unknown if it offers improved performance. They reject a population based evolutionary approach due to its inability to provide a human comprehensible graphical representation of a solution whereas we have attempted to retain these aspects of AIS methodology.

The common approach to representing a series in AIS algorithms and the one used by Dasgupta and Forrest [1] is to use a sliding window to sample the data and encode the sample as a single point in a sample space. This is a compact representation for anomaly detection but offers no possibility for forecasting. The information lost in the encoding process would also place a lower bound of the sensitivity of the detection function. With our approach, we can tune the detection limits to suit an application through the parametric thresholds. The explicit representation of the series enables the easy design of algorithms that match on subsequences of a series.

5. OBSERVATIONS AND CONCLUSIONS

In this paper we have introduced a new class of AIS search algorithm we call antigenic search. Two ideas distinguish it from traditional AIS search techniques. The first, by adding first and second derivatives to antibodies, enables a form of directed search that makes forecasting possible. The second, by adding sequential memory capability, enables ancestral retracing and the memory of sequences, a property useful for anomaly detection.

We investigated three memory structures and their impact on memory size. All show linear complexity in growth with the training series size being the upper bound. The FIFO structure performed best on forecasting tests while the graph structure had the best memory size performance. All have advantages and disadvantages leaving the application to direct the choice.

We described two algorithms, ***Exact*** and ***FIFO***, that when applied to Lorenz and Mackey-Glass time series data, provided a good ability to forecast succeeding members in the series. A test against random data revealed no forecasting ability which is expected.

In this initial approach, when a forecast is made, deviation from the actual next value is corrected by immediate feedback. Unlike with neural network approaches, the information provided by these corrections is thrown away. Consequently, if an attempt is made to forecast a series of steps without feedback, the algorithm will calculate a trajectory in state space that may quickly deviate from the actual future series. Finding a way to effectively utilize all training information to enable more than next event forecasting remains an open problem.

6. FUTURE WORK

In certain fields such as plant operations and computer systems, anomaly detection is of more interest than forecasting. Consequently we are investigating the application of antigenic search techniques to the anomaly and intrusion detection problems. Another area of current research concerns the representation of multiple series from a memory perspective. Work in this paper was conducted with a single series, but many time series problems offer multiple training series. Early consideration points to a number of alternatives approaches to this problem. Finally, the importance of memory constraints is being investigated along with specific techniques including population regulation by aging.

7. REFERENCES

[1] D. Dasgupta and S. Forrest. Tool breakage detection in milling operations using a negative-selection algorithm. Technical Report CS95-5, Department of Computer Sciences, University of New Mexico, 1995.

[2] D. Dasgupta and S. Forrest. Novelty detection in time series data using ideas from immunology. In *ISCA 5th International Conference on Intelligent Systems*, Reno, Nevada, 1996.

[3] L. N. de Castro and J. Timmis. *Artificial Immune Systems: A New Computational Approach.* Springer-Verlag, London, 2002.

[4] P. D'haeseleer, S. Forrest, and P. Helman. An immunological approach to change detection: algorithms, analysis, and implications. In *the 1996 IEEE Symposium on Computer Security and Privacy*, pages 110–119, Oakland, California, 1996.

[5] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri. Self-nonself discrimination in a computer. In *Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy*, pages 202–212, Oakland, CA, 1994. IEEE Computer Society Press.

[6] S. A. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. *Journal of Computer Security*, 6(3):151–180, 1998.

[7] N. K. Jerne. Towards a network theory of the immune system. *Ann. Immunol.*, 125(C):373–389, 1974.

[8] C.-F. Juang. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 34(2):997–1006, 2004.

[9] M. Kendall and J. Ord. *Time Series.* Edward Arnold, Seven Oaks, Kent, third edition, 1999.

[10] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proc. IEEE Int'l. Conf. on Neural Networks*, Perth, Australia, 1995. IEEE Service Center. www.engr.iupui.edu/ shi/Coference/psopap4.html.

[11] J. Kennedy, R. C. Eberhart, and Y. Shi. *Swarm Intelligence.* Morgan Kaufmann, 2001.

[12] M. V. Mahoney and P. K. Chan. Learning rules for time series anomaly detection. Technical report, Florida Institute of Technology, Melbourne, Florida, October 2004. http://cs.fit.edu/˜mmahoney/nasa/siam.pdf.

[13] A. S. Perelson and G. Weisbuch. Immunology for physicists. *Reviews of Modern Physics*, 69(4):1219–1267, 1997.

[14] D. Pollock. *A Handbook of Time-Series Analysis, Signal Processing and Dynamics.* Academic Press, London, 1999.

[15] S. Singh. Anomaly detection using negative selection based on the r-contiguous matching rule. In *1st International Conference on Artificial Immune Systems*, pages 99–106, Canterbury, UK, 2002.

[16] D. Smith, S. Forrest, and A. Perelson. Immunological memory is associative. In *Workshop Notes, Workshop 4: Immunity Based Systems, Intnl. Conf. on Multiagent Systems*, pages 62–70, 1996.

[17] D. J. Smith. *The Cross-Reactive Immune Response: Analysis, Modeling and Application to Vaccine Design.* PhD thesis, Department of Computer Science, University of New Mexico, December 1997.

[18] A. B. Somayaji. *Operating System Stability and Security through Process Homeostasis.* PhD thesis, Department of Computer Science, University of New Mexico, July 2002.

[19] A. Watkins, J. Timmis, and L. Boggess. Artificial immune recognition system (AIRS): An immune-inspired supervised learning algorithm. *Genetic Programming and Evolvable Machines*, 5(3):291–317, Sept. 2004.