Appendix A

Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

Thomas P. Oléron Evans, Steven R. Bishop and Frank T. Smith

A.1 Introduction

In Chapter 13, we attempted to predict extinctions of the predator species in an individual-based predator–prey model known as the NANIA model (Ackland, 2009). To achieve this, letting $u$ and $v$ represent the population densities of the prey and predator species respectively, we partitioned the phase space of the point $(u, v)$ into discrete bins and used data from computer simulations of the model to determine the risk of predator extinction within a certain number of steps, for observations drawn from each bin.

One drawback of this method is that the shape of the decision boundary – the frontier between the region of phase space in which extinctions are predicted and the region in which they are not predicted – is highly dependent on the shape of the bins chosen to partition the space. Using bins constructed from Cartesian products of half-open intervals, as in Chapter 14, results in a non-smooth decision boundary. However, in many cases a smooth boundary may be more appropriate. While reducing the size of the bins may allow for a more complex decision boundary to be traced, in practice, the need for each bin to contain a sufficient volume of data for predictions to be reliable and meaningful prevents a fine-scale partitioning of the space.

In this chapter, we employ a more versatile and powerful method to address the same question posed in Chapter 13: the prediction of predator extinctions in the NANIA model. This method–nonlinear support vector classification–is better suited to identify a decision boundary of the most appropriate form for the given data. The key concepts of the method are described in some depth in Section A.2, while the remaining sections describe the results of applying the method to NANIA model data.
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

A.2 Support Vector Classification

A.2.1 Overview

In this section, we build on and develop the material on binary classification presented in Chapter 13. The concepts and definitions are adapted from Steinwart and Christmann (2008), Young (2008) and Gretton (2013).

Support vector classification (SVC) is a form of binary classification that is widely used in image recognition (and indeed in pattern recognition more generally), where the process of deciding whether an image does or does not contain a particular thing (e.g. a face) must be automated (see e.g. Burges, 1998).

In the linear case, given a data sample (which we will consider to consist of points in $\mathbb{R}^n$), whose points are each assigned the category 1 or $-1$, SVC identifies an $(n-1)$-dimensional hyperplane in $\mathbb{R}^n$ (the decision boundary) with the aim of separating the positively categorised points from the negatively categorised ones. New data points, whose categories are unknown, may then be classified according to the side of the decision boundary on which they lie.

Note that if the true underlying distribution is such that the same point may belong to either category with non-zero probability or if the relationship between the location of a point and its category is nonlinear, a linear SVC procedure of this kind will not generally be able to categorise all points correctly. In such situations, a more general technique of nonlinear support vector analysis may be appropriate, to produce a decision boundary of a different form.

A.2.2 Definitions

As in Chapter 13, suppose that we have a random variable $\Theta$ taking values in $X \times \{-1, 1\}$, distributed according to a probability distribution $P$, which is generally unknown. We will describe $\Theta$ as a data-generating process. For reasons of simplicity, we will suppose that $X = \mathbb{R}^n$ for some $n \in \mathbb{N}$. A particular expression of $\Theta$, $(x, y) \in X \times \{-1, 1\}$, will be described as a data point, where $x = (x_1, \ldots, x_n)$ is a vector of explanatory variables and $y$ is the category or class of a particular point.

In Chapter 14, we introduced the concept of a binary classifier, a function which takes points in $X$ whose category is unknown and assigns them a value of either 1 or $-1$, in an attempt to estimate their category. Here, we consider a more general decision function, whose range is not restricted to $\{-1, 1\}$, but which may rather take any real value.

$$f : X \rightarrow \mathbb{R}$$

(A.1)

If $X \subseteq \mathbb{R}^n$, the following definition describes a simple form for a decision function:

**Definition A.1.** Let $X \subseteq \mathbb{R}^n$ and $b, w \in \mathbb{R}^n$, for some $n \in \mathbb{N}$. The linear decision function $f_{b,w}$ is defined:

$$f_{b,w} : X \rightarrow \mathbb{R}$$

$$x \mapsto \langle x - b, w \rangle$$

(A.2)

where $\langle \cdot, \cdot \rangle$ is the standard inner product on $\mathbb{R}^n$. The hyperplane $H(b, w)$ defined by:

$$H(b, w) = \{ v \in \mathbb{R}^n : \langle v - b, w \rangle = 0 \}$$

is known as the decision boundary.
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

Observe that the decision boundary divides the space into two disjoint regions, with the linear classifier assigning positive values to points on one side of the boundary, negative values to points on the other side and zero to points lying on the boundary itself.

Ideally, given a data point \((x, y)\) drawn from a particular data-generating process, a decision function designed for that process would assign the correct category to \(x\), such that \(f(x) = y\). The following definition provides a basis for evaluating how effectively a decision function performs this task.

**Definition A.2.** Given a data point \((x, y)\) and a real number \(r \in \mathbb{R}\), a loss function \(L\) assigns a non-negative real number (the loss) to these inputs:

\[
L : X \times \{-1, 1\} \times \mathbb{R} \rightarrow [0, \infty) \\
(x, y, r) \mapsto L(x, y, r)
\]  

(A.3)

For the purposes of this document, only loss functions that are independent of \(x\) will be considered. This means that the success of a decision function for a particular data point depends only on the true category of the point and the value assigned by the function. Loss functions can consequently be written as functions of two real variables \(L(y, r)\). Furthermore, loss functions will generally be applied to the output of a particular decision function \(f\), so it is the quantity \(L(y, f(x))\) that is specifically of interest.

The following two loss functions are of particular relevance:

**Definition A.3.** The classification loss \(L_{\text{class}}\) is defined as:

\[
L_{\text{class}} : \{-1, 1\} \times \mathbb{R} \rightarrow [0, \infty) \\
(y, r) \mapsto \begin{cases} 
0, & yr > 0 \\
1, & \text{otherwise}
\end{cases}
\]  

(A.4)

Note that, given a particular data point, provided that a decision function assigns a real number with the correct sign, the classification loss is 0. If, on the other hand, the function fails to match the sign of the true category, the classification loss is equal to 1 in all cases.

While the classification loss would appear to be the ideal loss function for use in classification problems, it is not convex. For practical reasons, it is often beneficial to replace it with the following convex surrogate, which is easier to work with:

**Definition A.4.** The hinge loss \(L_{\text{hinge}}\) is defined as:

\[
L_{\text{hinge}} : \{-1, 1\} \times \mathbb{R} \rightarrow [0, \infty) \\
(y, r) \mapsto \max(0, 1 - yr)
\]  

(A.5)

Given a particular data point, provided that a decision function assigns a real number with the correct sign whose absolute value is greater than or equal to 1, the hinge loss is 0, matching the classification loss. However, for values of \(yr\) less than 1, the hinge loss increases linearly as \(yr\) decreases. Note that the classification and hinge losses also coincide when \(r = 0:\)

\[
L_{\text{class}}(y, 0) = L_{\text{hinge}}(y, 0) = 1, \ \forall y \in \{-1, 1\}
\]
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

Figure A.1 A comparison of the classification loss and the hinge loss

While loss is a measure of how effectively a decision function classifies a particular data point, to measure the overall effectiveness of the function it is necessary to consider how well it performs across all possible data points. The following definition provides such a measure:

Definition A.5. Consider a data-generating process $\Theta$ taking values in $X \times \{-1, 1\}$ distributed according to a probability distribution $P$. Consider also a decision function $f : X \rightarrow \mathbb{R}$ and a loss function $L : \{-1, 1\} \times \mathbb{R} \rightarrow [0, \infty)$. $R_{L,P}(f)$, the risk of $f$ with respect to $L$ and $P$, is the expected loss of the decision function when classifying data points drawn from $P$:

$$R_{L,P}(f) = \mathbb{E}_{\Theta} [L(y, f(x))]$$

It is now possible to define the goal of SVC. Given a data set:

$$D = ((x_1, y_1), \ldots, (x_N, y_N)) \in (X \times \{-1, 1\})^N$$

whose points are specific expressions of $\Theta$ drawn from the distribution $P$ and whose categories $y_1, \ldots, y_N$ are known, the aim is to deduce a decision function $f_D$ whose risk $R_{L,P}(f_D)$ with respect to $P$ and to some specified loss function $L$ is as small as possible. This process is formalised in the following definition.

Definition A.6. Given a set $X$ and a positive integer $N$, a learning method $\mathcal{L}$ is a function which assigns a decision function $f_D$ to a data set $D \in (X \times \{-1, 1\})^N$:

$$\mathcal{L} : (X \times \{-1, 1\})^N \rightarrow X^R$$

where $X^R$ is the set of all functions from $X$ to $\mathbb{R}$. 

(A.6)
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

SVC is therefore a learning method that attempts to use data points whose categories are known to deduce a high-quality decision function to classify data points whose categories are unknown.

A.2.3  Hard-margin Linear SVC

This section and Section 15.2.4 draw on concepts from the field of optimisation theory. A detailed introduction to the relevant parts of this field is provided by Hoppe (2006).

Suppose that \( D = \{(x_1, y_1), \ldots, (x_N, y_N)\} \subseteq (X \times \{-1, 1\})^N \), with \( X \subseteq \mathbb{R}^n \), is a data set drawn from a data-generating process \( \Theta \), whose distribution \( P \) is entirely unknown. The goal is to choose \( b, w \in \mathbb{R}^n \) such that, in terms of the classification loss, the risk of the corresponding linear classifier \( f_{b,w} \) is as low as possible.

Initially, suppose that the data can be perfectly linearly separated into their respective categories. In other words, suppose that there exist \( b, w \in \mathbb{R}^n \) such that the linear classifier \( f_{b,w} \) (see Definition A.1) is such that:

\[
\begin{align*}
  f_{b,w}(x_i) > 0 & \iff y_i = 1, \quad \forall i \in \{1, \ldots, N\} \\
  f_{b,w}(x_i) < 0 & \iff y_i = -1, \quad \forall i \in \{1, \ldots, N\}
\end{align*}
\]  

(A.7)

In such a situation, there will be different choices of \( b \) and \( w \), giving rise to different decision boundaries \( H(b,w) \) and decision functions \( f_{b,w} \), all of which separate the given data correctly. It is necessary to determine which of these functions might be expected to be most effective at assigning categories to further unknown data sets drawn from \( \Theta \).

To do this, it is necessary to consider the following superset of \( H(b,w) \):

\[
M(b,w) = \{ v \in \mathbb{R}^n : \|v - b\| < 1 \} \tag{A.8}
\]

The region \( M(b,w) \) will be described as the margin of the hyperplane \( H(b,w) \). Observe that \( \|v - b\|/\|w\| \) is the shortest distance from the point \( v \) to \( H(b,w) \) (where \( \| \cdot \| \) represents the standard Euclidean norm in \( \mathbb{R}^n \)). Therefore, the margin consists of all points for which the shortest distance to \( H(b,w) \) is less than \( \|w\|^{-1} \). The quantity \( \|w\|^{-1} \) will be referred to as the margin size.

Note that, given \( b \in \mathbb{R}^n \), the vector \( w \) uniquely defines a hyperplane–margin pair \( H(b,w), M(b,w) \) and vice versa. The direction of \( w \) determines the normal to \( H(b,w) \), while the length of \( w \) determines the margin size. Note also, however, that the vector \( b \) is not uniquely determined, since given \( w \in \mathbb{R}^n \), many different vectors \( b \in \mathbb{R}^n \) define the same hyperplane and margin \( H(b,w), M(b,w) \).

Of all possible decision boundaries that correctly separate the data by category, it is intuitively sensible to choose the hyperplane \( H(b_0,w_0) \) for which the minimum distance from any point \( x_i \), \( i \in \{1, \ldots, n\} \) to \( H(b_0,w_0) \) is maximised. This is equivalent to maximising the margin size subject to the constraint that no point \( x_i \) lies in the margin.

The problem of finding the appropriate decision boundary can thus be stated as an optimisation problem. The phrase hard margin indicates that the method requires that all points in \( D \) are correctly classified and that none lies in the margin:
Optimisation problem: linear SVC–hard margin

- Given:
  \[ D = \{ (x_1, y_1), \ldots, (x_N, y_N) \} \in (\mathbb{R}^n \times \{-1, 1\})^N \]

- Maximise:
  \[ \|w\|^{-1} \]

- Over:
  \[ w, b \in \mathbb{R}^n \]

- Subject to:
  \[ y_i (x_i - b, w) \geq 1, \quad i \in \{1, \ldots, N\} \]

Note that the constraint combines the requirements that points are correctly classified and that they do not lie in the margin.

As observed, the optimal solution to this problem is not uniquely determined, since different vectors \(b\) may correspond to the same hyperplane and margin. A further constraint may therefore be imposed to ensure uniqueness (at least in cases where the optimal hyperplane and margin are themselves unique) and simplify the problem. Specifically, we simply impose the condition that \(b\) and \(w\) should be parallel:

\[ b = \lambda w, \text{ for some } \lambda \in \mathbb{R} \quad (A.9) \]

For reasons of practicality, we introduce the scalar variable \(b\), such that:

\[ b = -\|w\| \cdot \|b\| \quad (A.10) \]

\(b\) is therefore a (negative) measurement of the length of \(b\) in units scaled to the margin size. Equation (A.9) can thus be restated as:

\[ b = -\frac{b}{\|w\|^2} w \quad (A.11) \]

Observing that maximising \(\|w\|^{-1}\) is equivalent to minimising \(\|w\|^2\), the optimisation problem can therefore be restated as a quadratic programme over \(w\) and \(b\).

Quadratic optimisation programme: linear SVC–hard margin (adapted)

- Given:
  \[ D = \{ (x_1, y_1), \ldots, (x_N, y_N) \} \in (\mathbb{R}^n \times \{-1, 1\})^N \]

- Minimise:
  \[ \|w\|^2 \]

- Over:
  \[ w \in \mathbb{R}^n, \quad b \in \mathbb{R} \]

- Subject to:
  \[ -y_i (x_i, w) - b \leq 1, \quad i \in \{1, \ldots, N\} \]
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

If a solution \( w_0, b_0 \) to this problem exists, then (using equation (A.11)) it defines a unique hyperplane–margin pair \( H(\mathbf{w}_0, b_0), M(\mathbf{w}_0, b_0) \), and a unique linear classifier \( f_{\mathbf{w}_0, b_0} \). Furthermore, no other variables \( \mathbf{w}, b \) define the same hyperplane–margin pair.

A.2.4 Soft-margin Linear SVC

As in the hard-margin case, suppose that \( D = \{ (x_1, y_1), \ldots, (x_N, y_N) \} \in (X \times \{-1, 1\})^N \), with \( X \subseteq \mathbb{R}^n \), is a data set drawn from a data-generating process \( \Theta \), whose distribution \( P \) is entirely unknown. Once again, the goal is to choose \( b, \mathbf{w} \in \mathbb{R}^n \) such that, in terms of the classification loss, the risk of the corresponding linear classifier \( f_{b, \mathbf{w}} \) is as low as possible.

However, in the soft-margin case, it is not assumed that there exists a linear classifier that perfectly separates the data by category. Instead, points are permitted to lie within the margin or to be incorrectly classified. The optimisation problem is adapted to take this into account by adding the accumulated losses of misclassified points to the objective function.

Quadratic optimisation programme: linear SVC–soft margin

- Given:
  \[ D = \{ (x_1, y_1), \ldots, (x_N, y_N) \} \in (\mathbb{R}^n \times \{-1, 1\})^N \]
- Minimise:
  \[ \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{N} L_{\text{hinge}}(y_i, \langle x_i, \mathbf{w} \rangle + b) \]
- Over:
  \[ \mathbf{w} \in \mathbb{R}^n, \ b \in \mathbb{R} \]

\( C \in (0, \infty) \) is a constant that governs the relative importance of minimising the margin size against minimising the losses from poorly classified points.

In this problem, the hinge loss is used as a surrogate for the classification loss, since its convexity makes the optimisation more tractable. A justification for this substitution is given in Steinwart and Christmann (2008: 34–38).

To simplify the problem, the losses on each data point may be replaced with a vector of constrained variables \( \zeta = (\zeta_1, \ldots, \zeta_N) \):

Quadratic optimisation programme: linear SVC–soft margin (adapted)

- Given:
  \[ D = \{ (x_1, y_1), \ldots, (x_N, y_N) \} \in (\mathbb{R}^n \times \{-1, 1\})^N \]
- Minimise:
  \[ \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{N} \zeta_i \]
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

Over:
\[ w \in \mathbb{R}^n, \ \zeta \in \mathbb{R}^N, \ b \in \mathbb{R} \]

Subject to:
\[ -\zeta_i \leq 0, \quad i \in \{1, \ldots, N\} \]
\[ -\zeta_i \leq y_i(\langle x_i, w \rangle + b) - 1, \quad i \in \{1, \ldots, N\} \]  \(\text{(A.12)}\)

Observe that the constraints on \(\zeta\), if considered together, are equivalent to the statement:
\[ \zeta_i \geq L_{\text{hinge}}(y_i, (\langle x_i, w \rangle + b)) , \quad i \in \{1, \ldots, N\} \]

A.2.5 SVC with Unequal Class Weights

In certain situations, misclassifications of points from one category may not be of the same importance as misclassifications of points from the other category. Alternatively, while misclassifications of both forms may be of equal importance, an imbalance in the number of observations in each category in the training data \(D\) will result in a classifier which is disproportionately successful at classifying points in the more frequently observed category, which may not be desirable, particularly if the distribution of categories in \(D\) does not necessarily match the distribution of the data on which the classifier is to be used.

In situations such as these, it may be desirable to introduce class weights, \(l_{-1}, l_{1} \in \mathbb{R}^{+}\) which determine the loss associated with misclassifications of each sort. The effect is to replace the classification loss \(L_{\text{class}}\) with the weighted classification loss \(L_{w,\text{class},l_{-1},l_{1}}\), defined as follows:

**Definition A.7.** Given class weights \(l_{-1}, l_{1} \in \mathbb{R}^{+}\), the weighted classification loss \(L_{w,\text{class},l_{-1},l_{1}}\) is defined as:
\[ L_{w,\text{class},l_{-1},l_{1}} : \{-1, 1\} \times \mathbb{R} \rightarrow [0, \infty) \]
\[ (y, r) \mapsto \begin{cases} 0, & yr > 0 \\ l_y, & \text{otherwise}. \end{cases} \]  \(\text{(A.13)}\)

The corresponding convex surrogate is the weighted hinge loss \(L_{w,\text{hinge},l_{-1},l_{1}}\), defined as:

**Definition A.8.** Given class weights \(l_{-1}, l_{1} \in \mathbb{R}^{+}\), the weighted hinge loss \(L_{w,\text{hinge},l_{-1},l_{1}}\) is defined as:
\[ L_{w,\text{hinge},l_{-1},l_{1}} : \{-1, 1\} \times \mathbb{R} \rightarrow [0, \infty) \]
\[ (y, r) \mapsto \max(0, l_y(1 - yr)) \]  \(\text{(A.14)}\)

Using a weighted hinge loss in the adapted quadratic optimisation programme above has the effect of replacing the final set of constraints with:
\[ -\zeta_i \leq l_y[y_i(\langle x_i, w \rangle + b) - 1] , \quad i \in \{1, \ldots, N\} \]  \(\text{(A.15)}\)

It should be noted that, since changes to the class weights change the losses associated with misclassifications, the effect of such changes on an SVC procedure can overlap with the effect of changes to the parameter \(C\), by altering the balance between penalties due to misclassification and penalties associated with the margin width.
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

A.2.6 Nonlinear SVC

Clearly, the efficacy of the linear classification techniques described in the preceding sections is highly dependent on the distribution of positively and negatively classified observations throughout the space defined by the explanatory variables. In many cases, it will not be possible to separate the two classes effectively with a linear decision boundary, even in an approximate fashion using a soft-margin approach.

For such cases, there exists a generalisation of the support vector method, known as kernel-based, nonlinear support vector classification. Here, we describe this technique in general terms only. A thorough explanation of the support vector approach can be found in Steinwart and Christmann (2008).

Essentially, the nonlinear approach involves transforming the vector of explanatory variables into another space by means of a nonlinear function and performing the linear SVC procedure on the transformed vectors in the new space. However, for practical purposes, it turns out that such a procedure can effectively be performed without carrying out the transformation explicitly, simply by considering an appropriate kernel, a function which takes two (untransformed) vectors and returns the inner product of the corresponding transformed vectors in the new space. It transpires that the complete nonlinear SVC procedure may be fully characterised by specifying the relevant kernel function and the values of any necessary parameters.

In this chapter, we employ the following kernel, one of the most commonly used in the field of machine learning.

**Definition A.9.** Given two vectors $x_0, x_1 \in \mathbb{R}^n$, for some $n \in \mathbb{N}$, the Gaussian radial basis function (RBF) $k_r(x_0, x_1)$ is defined as:

$$k_r(x_0, x_1) = \exp \left[ -\frac{\|x_1 - x_0\|^2}{r^2} \right]$$  \hspace{1cm} (A.16)

where $r \in \mathbb{R}^+$ is a parameter to be chosen.

In the context of kernel-based SVC, the parameter $r$ governs the extent to which individual data points influence the shape of the decision boundary as their distance from that boundary increases. Small values of $r$ indicate that influence decreases rapidly with distance, resulting in a highly curved boundary, whose shape will be heavily dependent on the precise location of nearby data points (with a consequent risk of overfitting). Large values of $r$ indicate more even influence across all data points, resulting in a smoother boundary whose shape is less sensitive to the locations of individual points (with a consequent risk of underfitting) (see Ben-Hur and Weston, 2009).

A.3 Using SVC to Predict Extinctions in the NANIA Model

In this section, using the data collected from the simulations described in Chapter 13, we consider the prediction of predator extinctions using nonlinear SVC. As in that chapter, rather than using the complete microstates of the system to inform our predictor, the explanatory variables used for the classification will be based purely on the population sizes of the predator and prey species. Also in line with the work of Chapter 13, we consider predictions of lead
time $k^* = 21$, approximately equal to one complete period of the irregular oscillations of the population dynamics.

The concept is to use a subset $D$ of the categorised population-level observations:

$$(\hat{x}_k, y_k) \in \mathbb{N}^2 \times \{-1, 1\}, \quad k \in \{0, \ldots, 399999 - k^*\}$$

as training data for an SVC procedure. Recall that $\hat{x}_k$ is a vector of population totals for the prey and predator populations at the $k$th observation:

$$\hat{x}_k = (x_{1,k}, x_{2,k}) \in \mathbb{N} \times \mathbb{N}, \quad k \in \{0, \ldots, 399999\}$$

while $y_k$ is the class associated with each observation, equal to 1 if the corresponding simulation is observed to reach extinction of the predator species within $k^*$ steps, and equal to $-1$ otherwise.

The goal is to identify a binary classifier $f_{D,k^*}$ that will distinguish between the subsets $\mathcal{E}_{E,k^*}$ and $\mathcal{E}_{P,k^*}$ (the regions of the model state space in which extinction within $k^*$ should or should not be expected, respectively):

$$f_{D,k^*} : \mathbb{N}^2 \rightarrow \mathbb{R}, \quad \hat{x} \mapsto f_{D,k^*}(\hat{x})$$

such that $f_{D,k^*}$ assigns a positive value to observations $\hat{x}$ for which the system is ‘at risk’ of predator extinction within $k^*$ steps, and a negative value to other observations. Precisely what is meant by the phrase ‘at risk’ will determine the characteristics of the resulting classifier (e.g. its sensitivity and accuracy), with different interpretations leading to the production of classifiers with different properties.

As with the phase–space binning technique presented in Chapter 14, a classifier of this nature should serve as an early warning of extinction in the NANIA model, by returning positive values when the system moves into a region of the state space in which there is a high risk of extinction within $k^*$ steps.

The efficacy of SVC as a method to predict extinctions will depend on the values of the SVC parameter $C$, the class weights $l_{-1}$ and $l_1$, and the Gaussian RBF kernel parameter $r$. As discussed in Section A.2.6, $r$ governs the precise form of the kernel, while we recall that $C$ governs the relative importance of correctly classifying the training points against the importance of maximising the width of the margin around the classification boundary (related to the distance from the margin at which training points are penalised under the hinge loss; see Section A.2). The procedure used to choose these parameters is discussed in Section A.4.

### A.4 Choosing the Parameters for SVC

#### A.4.1 Selection of Training and Test Data

To determine the best values of the parameters $r$ and $C$ for the prediction of predator extinctions using population data, a number of exploratory support vector classifications were performed to explore a region of the combined parameter space of $r$ and $C$. The results of some of the most relevant of these are presented here.

Since extinctions in the data set are relatively rare, occurring only 934 times across the full set of 400,000 observations, the set of positively categorised observations (those for which
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

the predator species is extinct within $k^* = 21$ steps) is also rather small, comprising only 20,196 observations, approximately 5% of the 399,979 observations to which categories may be assigned. Any random sample used as training data for an SVC procedure must therefore necessarily be large, in order to contain sufficiently many such observations to infer useful information about their distribution.

One of the most significant obstacles for the successful application of SVC to this problem is the computing time required to perform the necessary optimisations for the large volume of data collected. The ideal approach to parameter selection – a very thorough investigation of prediction performance for a wide variety of parameters values, involving multiple repetitions of the procedure for each scenario – is consequently not feasible in this case. Instead, conclusions on the most effective parameter values must be drawn from the analysis of a relatively small number of support vector classifications.

Since the independence of training and test points for SVC is necessary, training points were selected from (roughly) the first half of the available categorised data ($\tilde{x}_0, y_0$) to ($\tilde{x}_{199989}, y_{199989}$), while test points were selected from the second half ($\tilde{x}_{199990}, y_{199990}$) to ($\tilde{x}_{399978}, y_{399978}$). Examining the data, it was noted that consecutive extinctions were observed at steps 199,797 and 200,006, so the pool of observations from which training points would be selected was reduced still further, to ($\tilde{x}_0, y_0$) to ($\tilde{x}_{199796}, y_{199796}$) to ensure that training and test observations could not be drawn from the same run of the system.

It should be noted that while this addresses the issue of independence between the training and test sets, the independence of points within each set remains an issue. This is because, for points drawn from the same run of a simulation, there exist long-range behavioural correlations which implies that the true categories of such points cannot be considered to be independent. However, the only guaranteed solution to this problem would be to draw only a single point from each run of the simulation, which would require the initial generation of a prohibitively large volume of data. We are therefore forced to accept some degree of category dependence within the training and test sets, and must bear this point in mind when formulating our conclusions.

30,000 training points were selected from the pool of training observations by means of a random uniform distribution, while 190,000 test points were selected in a similar fashion from the pool of test observations. The decision to use 30,000 training points was made because this was observed to be roughly the maximum value for which all desired exploratory support vector classifications could be performed within a reasonable period. 190,000 test points represents the majority of the data in the test pool, allowing classifier performance to be tested on almost the maximum number of available points, while also being a conveniently round number for the purposes of discussion.

A.4.2 Standardisation of Training and Test Data

In order that variations in prey and predator populations could be given equal weighting in the calculation, all training and test points were standardised using the mean and standard deviation of the training set. The reasoning behind using the training mean and standard deviation

---

1 Optimisations were performed in Python (Python Software Foundation, 2012), using the packages NumPy (Numpy Developers, 2012), CVXOPT (Andersen et al., 2014) and SciKit-Learn (SciKit-Learn Community, 2015).
on the test data is that the classifier itself is derived from the training data only, and so, for the classifier to be meaningful, only information related to this data should be used, and the same procedure applied to the training data should be applied to all new data to be classified.

The standardisation procedure is therefore:

\[
\hat{x}_k \mapsto \tilde{z}_k \\
(x_{1,k}, x_{2,k}) \mapsto (z_{1,k}, z_{2,k}) = \left( \frac{x_{1,k} - \bar{x}_{1}^{\text{train}}}{s_{1}^{\text{train}}}, \frac{x_{2,k} - \bar{x}_{2}^{\text{train}}}{s_{2}^{\text{train}}} \right)
\]

where \( \bar{x}_{1}^{\text{train}}, \bar{x}_{2}^{\text{train}} \) are the means and \( s_{1}^{\text{train}}, s_{2}^{\text{train}} \) the standard deviations of the two components of the training observations.

For the purposes of this section, then, we are concerned with the following sets of standardised, categorised training and test population data, renumbered consecutively for convenience:

\[
D_{\text{train}} = \{ (z_{0,0}^{\text{train}}, y_{0}^{\text{train}}), \ldots, (z_{29999,29999}^{\text{train}}, y_{29999}^{\text{train}}) \}
\]

\[
D_{\text{test}} = \{ (z_{0,0}^{\text{test}}, y_{0}^{\text{test}}), \ldots, (z_{189999,189999}^{\text{test}}, y_{189999}^{\text{test}}) \}
\]

The training and test data sets contain 1532 and 9587 positively categorised observations, respectively, and are visualised in Figures A.2 and A.3. From the first figure, the sparsity of useful information available at the edges of the distribution can be observed, with a high level of uncertainty over the appropriate category to allocate to points in these regions. Since the data become particularly sparse along the upper right edge of the distribution in particular, a small number of points in this region could have a large effect on the particular classifier produced by SVC, another point to bear in mind when interpreting the results.

### A.4.3 Selection of Parameter Pairs for Exploratory Classifications

In determining the best values of the SVC parameters \( r \) and \( C \) to predict extinctions (for a lead time of 21 steps), we initially set the class weights of the positively and negatively categorised points to both be equal to 1. This means that misclassifications of positively and negatively categorised training points are of equal cost in terms of finding an optimal classifier. Relating this to the terminology set out in Section 13.2.2, the SVC procedure will therefore be aiming to produce a classifier that maximises the accuracy of its predictions, and it is therefore in terms of the accuracy that the performance of classifiers so produced will be judged.

After an initial informal investigation, it was determined (based on considerations of performance and computing time required) that SVC should be performed on the training data for the following set of parameter pairs:

\[
(r, C) \in \{10^{-2}, 10^{-1}, 10^0, 10^1, 10^2\} \times \{10^2, 10^3, 10^4, 10^5, 10^6\}
\]

### A.4.4 Classifier Accuracy

The results, in terms of classification accuracy on the test data, are presented in Table A.1 and as a heat map in Figure A.4. Some of the classifiers produced are visualised in Figure A.5.
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

Figure A.2 Visualisation of the 30,000 standardised training data points $D_{\text{train}}$, comprising 28,468 points with category $-1$ (dark) and 1532 points with category $1$ (pale). Note that the pale points are plotted over the dark points, thus increasing their apparent prominence.

Note that, since 9587 of the 190,000 points in $D_{\text{train}}$ are positively categorised, a predictor assigning the category $-1$ to all points would have an accuracy of approximately 0.9494, with an effective associated computing time of zero. This is therefore a baseline level for successful prediction, with classifiers performing more poorly than this considered to be of no value.

From Figure A.4, we might hypothesise that considering values of $C$ greater than $10^6$ may result in further increases in predictive accuracy. However, increasing the value of $C$ also increases the computing time required for the SVC procedure, with the classifier for $(r, C) = (10^1, 10^6)$ requiring over 2 days to generate with the software and hardware available. Investigating the performance of classifiers for higher values of $C$ would not therefore have been feasible in general.

From Table A.1, of all those under consideration, the best parameter pairs $(r, C)$ appear to be $(10^0, 10^5)$ and $(10^{-1}, 10^0)$. The classifier produced using the latter pair of parameters provides a better accuracy at the fifth decimal place, but since the difference is equivalent to less than six misclassified observations, it cannot be said with certainty that one classifier is better than the other in general, given the random processes involved in generating and choosing the data.
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

Figure A.3 Visualisation of the 190,000 standardised test data points $D_{\text{test}}$, comprising 28,468 points with category $-1$ (dark) and 1532 points with category 1 (pale). Note that the pale points are plotted over the dark points, thus increasing their apparent prominence.

Table A.1 Prediction accuracy of 21-step-lead-time predator extinctions on $D_{\text{test}}$ of classifiers produced by kernel-based SVC on $D_{\text{train}}$, for 25 different pairs of the SVC parameters $(C, \gamma)$. All accuracies are given to four decimal places.

<table>
<thead>
<tr>
<th>log$_{10}C$</th>
<th>log$_{10}\gamma$</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td>0.9532</td>
<td>0.9563</td>
<td>0.9562</td>
<td>0.9505</td>
<td>0.9256</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>0.9506</td>
<td>0.9528</td>
<td>0.9563</td>
<td>0.9520</td>
<td>0.9269</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>0.9501</td>
<td>0.9520</td>
<td>0.9559</td>
<td>0.9529</td>
<td>0.9304</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0.9500</td>
<td>0.9517</td>
<td>0.9553</td>
<td>0.9539</td>
<td>0.9355</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>0.9498</td>
<td>0.9508</td>
<td>0.9543</td>
<td>0.9548</td>
<td>0.9415</td>
</tr>
</tbody>
</table>
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

Figure A.4 Prediction accuracy of 21-step-lead-time predator extinctions on $D_{\text{test}}$ of classifiers produced by kernel-based SVC on $D_{\text{train}}$ for 25 different pairs of the SVC parameters $(C, r)$, visualised as a heat map. Note that the lowest accuracy represented on the colour scale corresponds to the predictive accuracy of a classifier which assigns the class $-1$ to all points. All cells representing accuracies equal to or lower than this value are coloured black.

A.4.5 Visual Comparison of Classifiers

Figure A.5 allows us to compare the classifiers produced by SVC visually. The fragmentation and ‘bubbling’ of the decision boundary for the figures corresponding to $r = 10^0$ and $r = 10^1$ are indications of overfitting of the corresponding classifiers. An even higher degree of overfitting is visible in the classifiers with $r = 10^2$, which classify almost all points positively, except in the immediate vicinity of particular negatively classified training points, thus explaining the poor performance of these classifiers, as observed in Figure A.4.

The problem of small numbers of points on the upper-right edge of the training data having a disproportionate impact on the shape of the resulting classifier, predicted in Section A.4, can be clearly observed, particularly in the case of the two positively categorised points lying near the point $(1, 5)$ in the training data (see Figure A.2). From Figure A.5, we see that these two points have evidently caused a large and misleading region of positive classification to appear in many of the classifiers visualised, even in the case of the most successful classifier, corresponding to $(r, C) = (10^{-1}, 10^0)$. This problem could be alleviated by using a larger set of training points.

A.4.6 Classifier Efficiency

Rather than comparing classifiers in terms of their accuracy alone, we are also interested in their efficiency, in terms of how much computing time is required to derive them from $D_{\text{train}}$. Figure A.6 plots the accuracy of all classifiers against the logarithm of the computing time (in seconds) required to complete the corresponding SVC procedure with the computing software...
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

Figure A.5  Visualisation of different classifiers derived from the standardised training data shown in Figure A.2. For reasons of clarity, the axes are unlabelled and are identical to those used in Figure 15.2. The paler two shades represent regions in which a classifier assigns category $-1$; the darker two shades represent regions in which a classifier assigns category $1$. The middle two of these shades represent the margin around the decision boundary.
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

Figure A.6  
Comparison of the 21-step-lead-time prediction accuracy of classifiers generated with SVC against the computing time required to create them from training data, for different parameter pairs \((r, C)\). Computing time is measured in seconds.
Points for which no other classifier was found displaying both improved accuracy and reduced computing time are indicated with a bold border. The dotted line indicates the accuracy of a classifier which assigns the category \(-1\) to all points of the training data.
Seven classifiers of interest are labelled, corresponding to the following parameter pairs: A \((10^{-2}, 10^2)\), B \((10^{-1}, 10^2)\), C \((10^0, 10^2)\), D \((10^1, 10^2)\), E \((10^0, 10^3)\), F \((10^{-1}, 10^6)\) and G \((10^0, 10^4)\).

and hardware available. For the purposes of performing repeated SVC procedures to gain reliable estimates of classifier performance over different training and test sets, or to investigate the effect of varying the lead time or the relative class weights, it is necessary to consider classifiers which can be generated relatively quickly.

It should be noted that the computing time presented here must be considered as an exclusively relative measure, of value only as a comparative tool, since it clearly depends on the available hardware and on the precise way in which the SVC procedure is programmed.

Clearly, the only parameter pairs that are of interest to us are those that are not dominated by any other pair under consideration, in the sense that their corresponding classifier is both less accurate and slower to calculate than that of the dominating pair. The points representing those parameter pairs that are not dominated in this way are outlined in bold in Figure A.6.

Of these, two (labelled A and B) produce classifiers whose prediction accuracy on the test set barely exceeds that of a predictor that makes exclusively negative predictions. The remaining four, labelled C–F, are of interest to us and will be compared in more detail, along with the parameter pair of the point labelled G, whose performance is effectively indistinguishable from that of F, as discussed above.
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

A.5 Comparing SVC Parameters Using ROC Curves and Precision-recall Curves

A.5.1 Predicting Extinctions Using SVC with Variable-class Weights

To meaningfully compare the classifiers generated by our SVC procedures, it is necessary to look beyond accuracy as a measure of predictive performance and to take a broader view. In predicting predator extinctions, the quantities of particular interest are the true positive rate, the false positive rate and the precision of a given predictor. The first of these measures will tell us the proportion of extinctions that are identified, the second tells us how frequently extinctions will be falsely predicted in situations where no extinction will occur and the third tells us how confident we can be of an extinction occurring when one has been predicted.

At this point, we note that, in using SVC to create classifiers from our observations, we do have some degree of control over these quantities that has not yet been exploited. By varying the relative class weights in the support vector classification procedure, we can shift the penalty associated with false positives relative to the penalty associated with false negatives (see Section A.2.5). Raising the cost of false negatives by increasing the weight of positively categorised observations in the training set will have the effect of increasing the true positive rate of the resulting classifier (since the procedure will try harder to correctly classify positively categorised points), at the cost of increasing the false positive rate (since errors associated with false positives are relatively less costly) and decreasing the precision (since a greater proportion of positive predictions will be false alarms).

For a given pair of SVC parameters \((r, C)\), we can therefore create a family of predictors of varying sensitivity to extinction by varying the relative class weights of positively and negatively categorised observations. Such families may then be visualised and compared by means of ROC curves and precision-recall curves, in order to more thoroughly assess the relative value of classifications performed with the relevant parameters.

In Section A.4, it was observed that the complete set of 399979 categorisable observations for the prediction of 21-step-lead-time predator extinctions contains 20,196 positively categorised and 379,783 negatively categorised observations. When the SVC class weights are equal to 1 for both classes, the ratio of the total class weights is therefore approximately equal to \(1:18.8\), positive to negative. In conducting classifications to compare predictor performance as the class weight is varied, it would be interesting to consider a range of class weights that includes both the scenario in which this ratio is \(1:1\) and the scenario in which the ratio is reversed.

For this reason, holding the weight of negatively categorised observations equal to 1, we consider a geometric sequence of weights for the positively categorised observations \(R^0, \ldots, R^8\) where the common ratio is given by \(R = \sqrt[8]{379783/20195}\). Observe that \(R^0\) corresponds to the original situation in which the ratio of positive to negative total class weights is equal to \(1:18.8\); \(R^4\) corresponds to a ratio of \(1:1\), and \(R^8\) corresponds to a ratio of approximately \(18.8:1\).

A new set of training data \(D_{\text{train}}\) of 50,000 observations (a greater number than in previous sections since the parameter combinations requiring the most computer time are now disregarded) and a new set of test data \(D_{\text{test}}\) of 190,000 observations were generated and standardised using the mean and standard deviation of the training set, as before. For each of the five parameter pairs \(C-G\) identified in Section A.4, SVC procedures were performed using these new standardised data sets for the nine combinations of class weights listed above.
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

A.5.2 Maintaining the Relative Importance of the SVC Penalties While Varying the Class Weights

There is a subtlety related to comparing the performance of different predictors generated by SVC as the class width is varied. As noted in Section A.2.5, changing class weights can have the potentially inadvertent effect of altering the balance between penalties associated with misclassifications and penalties associated with the classifier margin width, which should be governed by the parameter $C$. Since we wish to compare the performance of different classifiers, corresponding to different class weights, while simultaneously maintaining the relative importance of these two types of penalty, it may be desirable to tweak the SVC objective function in order to maintain the balance between them.

In an attempt to achieve this, we multiply $C$ by a correction factor to ensure that the maximum theoretical penalty (i.e. if every point were to be misclassified) in all SVC procedures to be compared is held constant. This maximum penalty is calculated in terms of the weighted classification loss (see Definition A.7), with its convex surrogate, the weighted hinge loss (see Definition A.8), used in our SVC procedures as usual.

This is not a perfect solution by any means, since the only points that we would expect to contribute to the misclassification penalty are those that lie in regions in which the distributions of positively and negatively categorised points strongly overlap, and the ratio of positively to negatively categorised points in these regions will not generally be the same as the ratio across a complete data set. However, since holding the weight of one class constant and varying the other certainly have the effect of skewing the relative importance of classification accuracy and margin width, and since other potential corrections, such as fixing the sum of the two class weights to be equal to 2, do not seem any more justified than the approach proposed, we consider the method chosen to be an acceptable compromise.

A.5.3 Results and Discussion

The performance of each of the newly generated weighted SVC classifiers on the new test set $D_{\text{test}}$ is summarised in Table A.2. Performance is measured in terms of true positive rate, false positive rate and precision. Recall that the true positive rate is the proportion of positively categorised observations that are correctly classified as positive, the false positive rate is the proportion of negatively categorised observations that are incorrectly classified as positive and the precision is the proportion of those observations classified as positive that do indeed belong to the positive category.

As previously discussed, we may observe from the table that increasing the relative class weight associated with the positively categorised observations has the effect of increasing the true positive rate of the resulting SVC classifier at the cost of simultaneously increasing the false positive rate and decreasing the precision. This effect can be observed more clearly in Figure A.7, in which the classifiers produced for parameter sets $D$, $G$ and $F$ (presented in order of increasing $C$), for positive to negative class weight ratios of $R_0$, $R_2$, $R_4$, $R_6$ and $R_8$, are visualised. In the figure, we see that increasing the relative weight of the positive category has the effect of ‘pulling in’ the main region of negative classification more and more tightly around the area of highest density of negatively categorised points, leading to the identification of greater numbers of positively classified observations and the simultaneous misclassification of increasing numbers of negatively classified observations.
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

Table A.2
Summary of the performance of classifiers for the five parameter pairs \((r, C)\) identified in Section A.2.6: \(C\) \((10^2, 10^2)\), \(D\) \((10^1, 10^2)\), \(E\) \((10^0, 10^2)\), \(F\) \((10^{-1}, 10^8)\) and \(G\) \((10^0, 10^4)\). \(l_+\) and \(l_-\) are the weights assigned to the positive and negative categories, respectively, and \(R = \sqrt{379783/20195}\). For all SVC procedures represented here, \(C\) was multiplied by a correction factor to account for the different class weights, as described in Section A.5.2.

<table>
<thead>
<tr>
<th>Parameter set</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>True positive rate:</td>
<td>(R^0)</td>
<td>0.1817</td>
<td>0.2762</td>
<td>0.2244</td>
<td>0.2369</td>
</tr>
<tr>
<td></td>
<td>(R^1)</td>
<td>0.4068</td>
<td>0.3978</td>
<td>0.4031</td>
<td>0.4498</td>
</tr>
<tr>
<td></td>
<td>(R^2)</td>
<td>0.6173</td>
<td>0.5733</td>
<td>0.6136</td>
<td>0.6356</td>
</tr>
<tr>
<td></td>
<td>(R^3)</td>
<td>0.7523</td>
<td>0.7295</td>
<td>0.7483</td>
<td>0.7681</td>
</tr>
<tr>
<td>(l_+ / l_-) =</td>
<td>(R^4)</td>
<td>0.8553</td>
<td>0.8255</td>
<td>0.8528</td>
<td>0.8610</td>
</tr>
<tr>
<td>False positive rate:</td>
<td>(R^5)</td>
<td>0.9313</td>
<td>0.8988</td>
<td>0.9310</td>
<td>0.9345</td>
</tr>
<tr>
<td></td>
<td>(R^6)</td>
<td>0.9742</td>
<td>0.9592</td>
<td>0.9728</td>
<td>0.9745</td>
</tr>
<tr>
<td></td>
<td>(R^7)</td>
<td>0.9881</td>
<td>0.9796</td>
<td>0.9883</td>
<td>0.9885</td>
</tr>
<tr>
<td></td>
<td>(R^8)</td>
<td>0.9960</td>
<td>0.9885</td>
<td>0.9938</td>
<td>0.9950</td>
</tr>
<tr>
<td>Precision:</td>
<td>(R^0)</td>
<td>0.0046</td>
<td>0.0082</td>
<td>0.0053</td>
<td>0.0061</td>
</tr>
<tr>
<td></td>
<td>(R^1)</td>
<td>0.0193</td>
<td>0.0192</td>
<td>0.0184</td>
<td>0.0238</td>
</tr>
<tr>
<td></td>
<td>(R^2)</td>
<td>0.0518</td>
<td>0.0497</td>
<td>0.0513</td>
<td>0.0560</td>
</tr>
<tr>
<td></td>
<td>(R^3)</td>
<td>0.0960</td>
<td>0.0949</td>
<td>0.0944</td>
<td>0.1046</td>
</tr>
<tr>
<td>(l_+ / l_-) =</td>
<td>(R^4)</td>
<td>0.1698</td>
<td>0.1589</td>
<td>0.1678</td>
<td>0.1769</td>
</tr>
<tr>
<td></td>
<td>(R^5)</td>
<td>0.2765</td>
<td>0.2541</td>
<td>0.2762</td>
<td>0.2807</td>
</tr>
<tr>
<td></td>
<td>(R^6)</td>
<td>0.4101</td>
<td>0.3815</td>
<td>0.4041</td>
<td>0.4154</td>
</tr>
<tr>
<td></td>
<td>(R^7)</td>
<td>0.5131</td>
<td>0.4795</td>
<td>0.5177</td>
<td>0.5278</td>
</tr>
<tr>
<td></td>
<td>(R^8)</td>
<td>0.6410</td>
<td>0.5537</td>
<td>0.6020</td>
<td>0.6314</td>
</tr>
</tbody>
</table>

To compare the parameter sets of interest (C–G) in terms of their ability to produce successful classifiers, we require a means of comparing their performance for all relative class weights simultaneously. To do this, we plot the performances of all classifiers generated with each parameter set as an ROC curve and as a precision-recall curve, and compare the shapes of these curves. The ROC curves are presented in Figure A.8, and the precision-recall curves are presented in Figure A.9. These curves allow for the identification of, for example, the false positive rate or precision that can be attained with a particular parameter set, if a specified true positive rate is required.
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

Parameter Set \((r,C)\)

\[
\begin{align*}
\frac{l_+}{l_-} &= R^4 \\
\frac{l_+}{l_-} &= R^4 \\
\frac{l_+}{l_-} &= R^4 \\
\frac{l_+}{l_-} &= R^4 \\
\frac{l_+}{l_-} &= R^4 \\
\frac{l_+}{l_-} &= R^4 \\
\end{align*}
\]

\(D\ (10^1,10^2)\) \hspace{1cm} \(G\ (10^2,10^4)\) \hspace{1cm} \(F\ (10^{-1},10^6)\)

**Figure A.7** Visualisation of different classifiers derived from a set of 50,000 standardised training points. The axes and shading are as in Figure A.5. \(l_+\) and \(l_-\) are the class weights, and \(R = \sqrt{379783/20195}\). For all SVC procedures, \(C\) was multiplied by a correction factor to account for the different class weights, as described in Section A.5.2
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

The figures indicate that, despite the qualitative difference in the shapes of the classifiers produced in the $z_1 - z_2$ plane (see Figure A.7), four of the five parameter sets, C and E-G, exhibit extremely similar performance profiles as the relative class weights are varied, with the largest differences occurring for the classifiers with the lowest true positive rate (in this case, this occurs where the positive and negative categories are assigned equal weight $I_+ / I_- = R^0$). While parameter set D may be observed to produce slightly inferior classifiers in general, where a true positive rate of greater than 0.5 is desired, the other four parameter sets appear to produce classifiers whose performance is nearly indistinguishable.

Comparing the performance of the SVC classifiers with the best performance obtained by the binned classifiers discussed in Chapter 13, also visualised in the figures, we again observe very little difference in the ROC plot, though the SVC classifiers do appear to offer visibly improved performance when precision is included in the comparison, as seen in the precision-recall plots, particularly for true positive rates of less than 0.7. We would certainly expect that the use of SVC should offer some improvement in performance over the classifiers...
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

Figure A.9  Precision-recall curves for classifiers generated with five different SVC parameter sets, for the prediction of lead time 21 extinctions of the predator species. Each curve consists of nine points, representing classifiers generated with positive to negative class weight ratios of \( R^0 \) to \( R^8 \), using the \( C \) correction factor described in Section A.5.2. Also included is a curve representing the best performances attained by the binned data predictors. Note that the term “recall” is synonymous with “true positive rate”. The dotted line at \( y = 0.0502 \) (the proportion of positively categorised points in the data) represents performances that could be achieved by classifiers making random predictions according to a Bernoulli variable, independently of the data.

Produced from binned data, since, while the two methods are making use of the same information, with the binned data, the decision boundary is forced to follow the boundaries of the bins, while with SVC it is free to take on a form that better corresponds to the distribution of the data.

The similarity in performance between the different SVC classifiers may indicate that, for those exhibiting true positive rates in excess of 0.5, classifiers are attaining near optimal performance given the amount of useful information on future extinctions that is conveyed by the two dimensions used in these SVC procedures. This would suggest that obtaining better predictions of extinction would require the inclusion of additional information on the state of the system. In particular, it may suggest that better predictive performance (for a 21-step lead time, at least) could only be achieved through introducing some form of consideration of the positions of individual agents.
Appendix A: Using Support Vector Analysis to Predict Extinction Events in Multi-Agent Models

A.6 Conclusion

In this chapter, we have applied the method of nonlinear support vector classification to the problem of predicting predator extinctions in the NANI A model. We have observed that while this method offers some improvement in classifier performance over the binning procedure set out in Chapter 13, such gains are relatively modest, and it may be the case that near optimal classification performance has been achieved given the amount of information on predator extinctions conveyed by the population densities, for a lead time of 21 steps.

However, there is a significant amount of spatial information that could be drawn from the simulations, which has not been used in the classifications performed in this chapter or in Chapter 14. The two methods presented in these chapters, phase space binning and support vector analysis, are both conducive to the incorporation of additional explanatory variables to describe the distribution of predator and prey individuals within the system at any given time, and it would be interesting to investigate what additional predictive value, if any, could be gained from such supplementary information.

It may also be the case that 21 steps is too distant a time horizon over which to predict extinctions with a high degree of reliability. A thorough investigation of the predictive performance that can be achieved for different time horizons would therefore also be of value. In practice, an early warning system to detect future predator extinctions would operate at many time horizons simultaneously, communicating both the risk of extinction and the time remaining in which interventions could be made to protect the endangered species.

The method presented here and that presented in Chapter 14 are clearly applicable to a far wider range of models and problems than the prediction of predator extinctions. Any model of real-world system from which data can be collected and in which rare events may be observed would be amenable to predictive methods such as these.

References


