**Universidad U-TAD**

DEPARTAMENTO DE INGENIERÍA DE SOFTWARE


# Técnicas matemáticas para el desarrollo de un modelo de simulaciones por inteligencia artificial

Javier Coque Fernández, David García Lleida, Alexis Gómez Chimeno,

Álvaro Martínez Parpolowicz, Claudia Reyero Bardelas y Jorge Tesch Torres

PROFESOR/A GUÍA: Javier Alcoriza Vara

02-2023

# Resumen

Como objetivo de entrega, se propone establecer el marco teórico matemático sobre el que se va a basar todo el desarrollo del modelo de simulaciones propuesto en la asignatura. Concretamente, se establecerá qué modelos poblacionales se utilizarán de forma comparativa y qué métodos matemáticos se aplicarán para llegar a una resolución al problema propuesto. Asimismo, se incluirán posibles variaciones a estos modelos, a tener en cuenta en caso de querer hacer una evolución futura de dichos modelos.

# Tabla de Contenidos

# Lista de Figuras

# Nomenclatura

| Símbolo | Significado, unidades |
|---|---|
| $x_i, y_i$ | Tamaño de la población para i especies distintas |
| $t$ | Tiempo, s |
| $\alpha$ | Tasa de cambio en la tasa de reproducción de la presa |
| $\beta$ | Tasa de cambio en la interacción entre presa y depredador |
| $\omega$ | Tasa de cambio en la pérdida de depredadores |
| $\delta$ | Tasa de cambio en la tasa de reproducción del depredador |
| $s_{ij}$ | Efecto de la especie i en la especie j |
| $r$ | Tasa de crecimiento |
| $K$ | Capacidad de carga de la especie |
| $\gamma$ | Factor de descuento de recompensa |
| $\sigma$ | Factor de aprendizaje (Learning Rate) |
| $Q(s, a)$ | Q-table |
| $\hat{Q}(s, a)$ | Q-table actualizada |
| $R(s, a)$ | Función recompensa por tomar 'a' desde 's'. |
| $p$ | Número de presas adyacentes a un agente |
| $d$ | Número de depredadores adyacentes a un agente |
| $\tau$ | Población total en un determinado instante de tiempo |
| $E$ | Energía actual de un agente |
| $TR$ | Tasa de riesgo total biológico |
| $\mathbb{P}[...]$ | Probabilidad de que se dé el suceso "..." |
| $\mathbb{E}[...]$ | Esperanza matemática de la variable "..." |

# Capítulo 1

# Marco Teórico

En este capítulo se presenta los diferentes modelos teóricos que serán necesario para elaborar el proyecto propuesto.

## 1.1 Modelo Lotka-Volterra

Para la evaluación de un modelo presa-depredador, el primer marco matemático que se estudia son las ecuaciones de Lotka-Volterra.

No obstante, estas ecuaciones no modelan todas las posibles situaciones que se puedan dar en un entorno biológico. Dicho esto, surgen nuevas consideraciones a tener en cuenta en nuevas adaptaciones de este modelo original. Todos estas adaptaciones son presentadas a continuación.

Se introducen las siguientes ecuaciones diferenciales de primer orden no lineales que nos permiten modelar la interacción de dos especies en un entorno cerrado siguiendo nociones básicas de interacción biológica entre presas y depredadores. Estas ecuaciones se definen de la siguiente manera.

$$\frac{dx}{dt} = x(\alpha - \beta y) \tag{1.1}$$

$$\frac{dy}{dt} = -y(\omega - \delta x) \tag{1.2}$$

En estas ecuaciones se tienen en cuenta las siguientes consideraciones.

1. Las presas tienen suministro de comida ilimitado en un tiempo dado y se reproducen exponencialmente a menos que exista algún depredador.
2. La muerte natural de los depredadores también está representada de forma exponencial.
3. La tasa de cambio de la población es proporcional a su tamaño.
4. El entorno no se adapta a favor de ninguna especie ni se tiene en cuenta la adaptación genética.

Con las ecuaciones definidas, podemos representar dos gráficas diferentes. Una es dependiente del tiempo, y se va a establecer como el objetivo a satisfacer por el desarrollo del modelo de aprendizaje por refuerzo. Claramente esta depende totalmente del valor de los parámetros asignados, por lo que, de forma ilustrativa, se han escogido los valores $\alpha = 1.1$ $\beta = 0.4$ para las ecuaciones de la presa y $\omega = 0.1$ $\delta = 0.4$ para las ecuaciones del depredador. El gráfico se muestra a continuación.

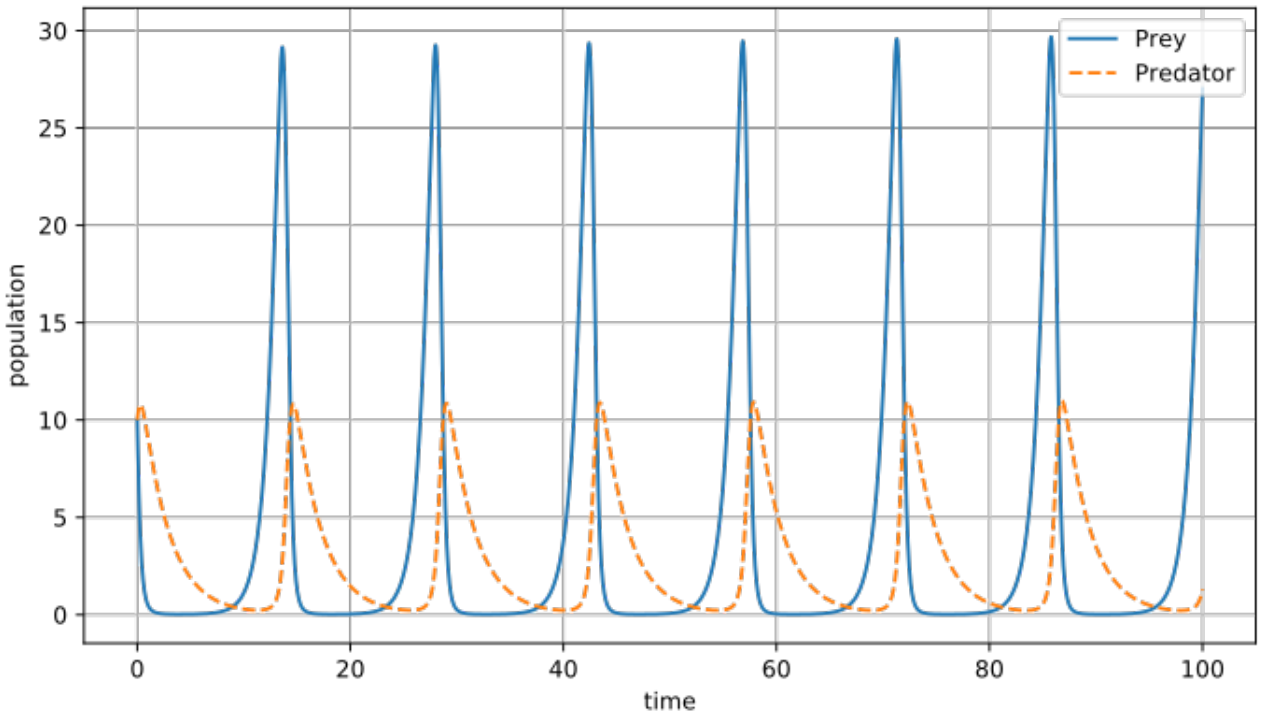


Figura 1.1: Modelo Lotka-Volterra dependiente del tiempo [1]

La segunda gráfica es independiente del tiempo. Claramente podemos dividir la ecuación del depredador con respecto a la presa y obtener lo siguiente:

$$\frac{dy}{dx} = -\frac{y(\delta x - \omega)}{x(\beta y - \alpha)}$$

Esta es claramente resoluble mediante separación de variables porque no hay ningún tipo de acoplamiento ni de no linealidad. Por tanto, se obtiene lo siguiente:

$$f(x, y) = \delta x - \omega \ln x + \beta y - \alpha \ln y$$

Con esta función ya podemos representar el espacio de fases, dando lugar a la segunda gráfica. Esta nos ilustra la dependencia de las dos especies en tamaño para varias condiciones iniciales.

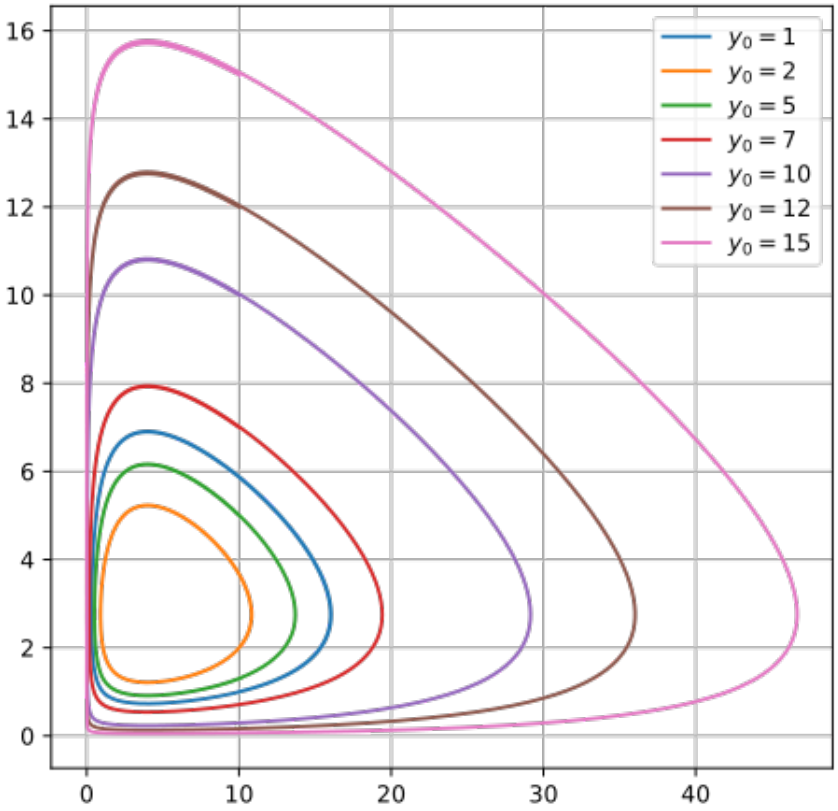


Figura 1.2: Modelo Lotka-Volterra espacio de fases [2]

## 1.2 Aprendizaje por refuerzo

Una de las técnicas para desempeñar papeles adaptativos en situaciones que el ser humano no es capaz de modelar en su totalidad, ampliamente utilizada en diferentes campos de las ciencias, es el aprendizaje por refuerzo.

El aprendizaje por refuerzo es un tipo de aprendizaje automático que nos permite modelar el aprendizaje humano a través de la interacción. Estas interacciones no hacen alusión al aprendizaje que se puede obtener de un profesor en una escuela, sino a la actuación que se debe tomar en base a una situación dada.

Un ejemplo ilustrativo de esta situación es cuando un recién nacido tiene hambre y desarrolla la habilidad de llorar para informar a sus progenitores de que algo no va bien. Esta habilidad claramente no se le ha sido enseñada, la ha desarrollado en base a una situación dada. Puliendo este concepto y llevándolo a un ámbito más formal, se puede definir el objetivo del aprendizaje por refuerzo.

### 1.2.1 Funcionamiento

El objetivo principal del aprendizaje por refuerzo en términos formales es dejar que un agente explore un entorno dado y maximice una función de recompensa. La hipótesis que lo subyace es que el objetivo buscado puede ser alcanzado mediante la maximización acumulativa de una función de recompensa.

### 1.2.2 Definición matemática

### 1.2.3 Proceso de decisión Markoviano

Como se ha mencionado, el objetivo del aprendizaje por refuerzo es maximizar una función de forma acumulativa. Esta acumulatividad obliga a definir una secuencia de estados, de tal manera que siempre que se quiera definir un nuevo paso se tiene que tener en cuenta el inmediatamente anterior.

En este caso, el aprendizaje por refuerzo se basa en que la probabilidad de transición es independiente del tiempo. Por tanto, las probabilidades de transición se definen de la siguiente manera:

$$\mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right] = \mathbb{P}\left[S_t = s' \mid S_{t-1} = s\right]$$

Claramente esto define una matriz de estados posibles que denotaremos $\mathcal{P}_{ss'}$.

No obstante, también queremos que estos estados sean dependientes de las acciones tomadas, para poder evaluar qué acciones son mejores. Esto hace que por cada acción tomada y estado evaluado, tengamos que evaluar si recompensar positivamente o negativamente, en función de la solución obtenida. Todo esto, hace que necesitemos más artefactos matemáticos en la definición.

Por un lado, tenemos que redefinir la matriz de probabilidad de estados para que esta dependa de las acciones tomadas. Esto se consigue de la siguiente manera:

$$\mathcal{P}^a_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s, A_t = a\right]$$

Por otro lado, necesitamos tener una función recompensa asociada a cada pareja de estado-acción, luego definimos:

$$\mathcal{R} : \mathcal{S} \times \mathcal{A} \longrightarrow \mathbb{R}$$

Una vez que tenemos bien definidos las cuaternas $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, podemos definir la función ganancia. Esta función ganancia se basa en el principio de alcanzar el mejor objetivo posible al final. Esto significa que tenemos que evaluar la importancia de las diferentes recompensas obtenidas por las parejas estado-acción a lo largo del tiempo y ver cuáles aportan un mayor objetivo final. Esto hace que la función ganancia se defina de la siguiente manera:

$$G_t = R_{t+1} + R_{t+2} + ... = \sum_{k=0}^{\infty} R_{t+k+1}$$

Sin embargo, esta serie es claramente geométrica, por lo que en el infinito no converge. Dicho esto, necesitamos introducirle un factor acotado que compense este hecho. A este factor lo llamaremos $\gamma$ tal que $\gamma \in [0, 1]$. Esto hace que se defina la función ganancia finalmente de la siguiente manera.

$$G_t = R_{t+1} + \gamma R_{t+2} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

Este factor tiene una serie de propiedades que se listan a continuación:

1. Permite trabajar con la incertidumbre de futuras recompensas
2. Permite evaluar si recompensas a corto plazo son mejores que a largo plazo
3. Permite tener una serie de estados convergente

Con estos artefactos definidos tenemos una 5-tupla $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \gamma, \mathcal{R})$ que constituye el proceso de decisión Markoviano para el aprendizaje por refuerzo.

### 1.2.4 Creación de reglas o políticas para el entrenamiento

Si bien ya tenemos definida nuestra función de ganancia a maximizar, falta estipular el proceso de decisión con el que podemos garantizar la optimalidad de la solución para que verdaderamente se produzca un aprendizaje correcto.

Para ello definimos una serie de políticas o reglas en forma de distribución sobre los estados posibles. Esto se formula de la siguiente manera:

$$\pi(a|s) = \mathbb{P}\left[A_t = a \mid S_t = s\right]$$

Para encontrar la mejor distribución de políticas o reglas que maximizarán la ganancia, se introducen dos funciones nuevas. La primera $v_\pi$ es la función estado-recompensa y se define como la esperanza de empezar en el estado S y seguir la política $\pi$.

$$v_\pi(s) = \mathbb{E}_\pi\left[G_t \mid S_t = s\right]$$

Sin entrar en detalles de demostraciones, esta se puede descomponer en términos de la esperanza de una ganancia inmediata y el valor descontado de ganancia del siguiente estado.

$$v_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} \mid S_t = s\right] + \mathbb{E}_\pi\left[\gamma v_\pi\left(S_{t+1}\right) \mid S_t = s\right]$$

La segunda $q_\pi$ es la función acción-recompensa y se define como la esperanza de empezar en el estado $s$, tomar la acción $a$ y seguir la política $\pi$. De igual modo, puede ser representada como la anterior.

$$q_\pi(s, a) = \mathbb{E}_\pi\left[R_{t+1} + \gamma q_\pi\left(S_{t+1}, A_{t+1}\right) \mid S_t = s, A_t = a\right]$$

Para simplificar notación llamaremos $\mathcal{R}_s^a = \mathbb{E}_\pi\left[R_{t+1} \mid S_t = s, A_t = a\right]$

Estas dos ecuaciones están estrechamente relacionadas y nos permiten definir lo que se conocen las ecuaciones de Bellman. Estas ecuaciones nos van a permitir calcular la función recompensa dada una política o regla, que es uno de los objetivos de la tarea de aprendizaje relacionando las funciones estado-recompensa y acción-recompensa con las del resto de estados. Estas ecuaciones se definen para ambas funciones de la siguiente manera.

$$v_\pi(s) = \sum_{a \in A} \pi(a \mid s) \left( \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_{\pi)}(s') \right)$$

$$q_\pi(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \sum_{a' \in \mathcal{A}} \pi(a' \mid s') q_\pi(s', a')$$

Cabe destacar que el análisis de complejidad temporal se basa en la resolución de estas ecuaciones lineales para hallar los valores de las recompensas. Dicho esto, esta resolución presenta una complejidad $O(n^3)$.

### 1.2.5 Maximización de la función recompensa y políticas óptimas

Para asegurar la existencia y la unicidad de una política óptima, nos basamos en el teorema de optimalidad del operador de Bellman. No se entra en detalle en este teorema.

Sabiendo esto, podemos definir la maximización de las ecuaciones de Bellman de la siguiente manera:

$$v_*(s) = \max_\pi v_\pi(s) \; ; \; q_*(s,a) = \max_\pi q_\pi(s,a)$$

Por el teorema mencionado, tenemos que la mejor política será la siguiente:

$$\pi_*(a \mid s) = \begin{cases} 1, \text{ si } a = \arg\max_{a \in \mathcal{A}} q_*(s,a) \\ 0 \end{cases} . \tag{1.3}$$

Para encontrar la función de recompensa óptima, volvemos a utilizar las ecuaciones de Bellman.

$$v_*(s) = \max_a \left( \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a v_*(s') \right)$$

$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a \max_{a'} q_*(s', a')$$

# Capítulo 2

# Métodos

En este capítulo se detallan los métodos o procedimientos utilizados para satisfacer el objetivo del trabajo.

## 2.1 Implementación Q-Learning

A la hora de implementar programáticamente el aprendizaje por refuerzo, existen diferentes variantes que se centran en utilizar parte de las definiciones anteriores o modificaciones de las mismas. Tras un análisis de complejidad espacial y temporal, se ha decidido utilizar la implementación del algoritmo Q-Learning.

El algoritmo Q-Learning tiene por objetivo aprender una estrategia que le diga a un agente qué acción tomar bajo unas circunstancias específicas. Este algoritmo utiliza una tabla, denominada Q-table, donde va a guardar para cada estado y para cada una de las posibles acciones a realizar, un valor que nos permite decidir qué acción es mejor. Para ello utiliza la siguiente función:

$$\hat{Q}(s,a) = Q(s,a) + \alpha \cdot \Big[R(s,a) + \Big(\gamma \cdot \max_{a'} Q(s',a')\Big) - Q(s,a)\Big]$$

En esta función podemos observar que la nueva Q-table se va actualizando en base a la anterior más un término donde se tiene en cuenta un factor de aprendizaje ($\alpha$), la función de recompensa ($R(s,a)$), el factor de descuento ($\gamma$), la acción más prometedora del siguiente estado del agente ($\max_{a'} Q(s',a')$) y otra vez el valor anterior de la tabla.

Lo más notable de este algoritmo es poder evaluar cuál es el mejor camino a tomar. Aparte de tener en cuenta el factor de descuento, se utiliza el máximo futuro valor de las acciones del futuro estado del agente. Esta pecularidad hace que el algoritmo sea idóneo con el proyecto propuesto por tres motivos:

1. El aspecto más importante es que los agentes que entrenan necesitan adaptarse al entorno constantemente y por ello necesitan computar su estado en la iteración siguiente y obtener el valor que les devuelva la acción más prometedora.

2. A diferencia de otros algoritmos similares, como SARSA (State-Action-Reward-State-Action), este algoritmo solo computa un paso más allá del estado actual y ninguno más. Esto es importante, puesto que adelantarse demasiados estados y acciones en base al actual puede ser contraproducente según los cambios en el entorno producidos por el resto de agentes que no se conocen en el momento de evaluación de la acción.

3. Implementaciones más eficientes y avanzadas como el Deep Q-Learning escapan de la disponibilidad hardware del proyecto, ya que utilizarlas supondría una complejidad computacional mayor que no se puede permitir con los recursos actualmente disponibles. Nótese que esta complejidad se evalúa más adelante.

Por último, falta por comentar cómo se va a producir la evaluación de las acciones de los diferentes agentes según sus estados. Esto se comenta en el siguiente apartado.

### 2.1.1 Evaluación del estado y de las acciones del agente

El enfoque principal de las acciones y estados del Q-learning es resolver un problema donde las condiciones del entorno se mantienen estables. Esta noción es fácil de entender con nociones de videojuegos básicas.

Supongamos que un jugador llega a un determinado nivel o pantalla de un videojuego. El jugador tiene que entender cómo funciona el mapa y los diferentes elementos para poder encontrar la manera más eficiente de resolver el nivel. Esto es fácil de computar con la variante elegida, ya que los estados y acciones del agente son por dónde moverse y qué acciones realizar, tal y como si jugara el nivel un jugador humano. No obstante, para el problema planteado la situación cambia.

Cuando se busca modelar especies en un entorno, no se puede asumir que el entorno sea constante, puesto que se ve afectado por diferentes estímulos. Cabe decir, que tampoco interviene un único agente que intenta descubrir la manera de conseguir su objetivo, sino múltiples agentes que persiguen objetivos comúnes y distintos. Estas dos condiciones hacen que se tenga que replantear cómo evalúar los estados y acciones de los agentes.

Para computarlo, se introducen las nociones de evaluación de riesgos.

#### Introducción al riesgo biológico

Como el objetivo es modelar un entorno presa-depredador con disponibilidad de alimento, se necesita evaluar qué riesgo biológico supone tomar determinadas acciones en diferentes estados.

Sin entrar en detalles de comportamientos específicos de especies, podemos entender los riesgos básicos de supervivencia siguiendo los principios de las cadenas tróficas. Dicho esto, los riesgos toman en cuenta las siguientes características:

1. Existe una especie depredador máximo en lo más alto de la cadena trófica que no es alimento de nadie y, por tanto, su objetivo principal es comer a sus presas potenciales para sobrevivir y reproducirse.

2. Existe una especie intermedia cuyo objetivo es intentar sobrevivir a sus depredadores y alimentarse con sus presas inmediatas para poder sobrevivir y reproducirse.

3. Existe una presa mínima que no es depredador de ninguna otra, que no tiene por qué ser presa del depredador máximo y cuyo objetivo es mantener el entorno natural para que puedan existir otras especies.

4. La energía es la esencia básica de supervivencia que se agota a través del tiempo, un individuo sin energía entra en un estado de muerte.

Estas características son sin duda un resumen de comportamientos biológicos más complicados, pero recogen de forma eficiente los instintos animales más primitivos.

Ahora bien, falta por definir la conceptualización de estos conceptos dentro de un entorno biológico. Para ello, se debe tener en cuenta cómo de peligroso es para un individuo estar cerca de sus depredadores, cómo de bueno es estar cerca de sus presas, cómo le afecta la pérdida y ganancia de energía, y cómo afecta la población total a sus acciones.

Con estos 4 factores claramente modelizamos los instintos anteriormente definidos y concretamos qué se debe tener cuenta para evaluar el riesgo biológico. Dicho de otra manera, sin necesidad de ningún elemento adicional se puede definir la siguiente función, que discretiza de forma adecuada la tasa de riesgo biológico o TR.

$$TR(p, E, d, \tau) = \frac{p\left(1 - \frac{E}{100}\right)^2 - d\left(\frac{E}{100}\right)^2}{\tau}$$

Esta función es la misma para todas las especies independientemente de su lugar en la cadena trófica, salvo para la presa mínima.

Analizando uno a uno los términos de la función, podemos ver lo siguiente:

1. El primer sumando del numerador computa el número de presas adyacentes ($p$) por una necesidad de energía cuadrática $\left(1 - \frac{E}{100}\right)^2$ donde $E$ es la energía del individuo en cuestión. Esto evalúa si se tiene una necesidad de obtener comida urgentemente y si se tiene al alcance.

2. El restando del numerador computa el número de depredadores adyacentes ($p$) por un factor cuadrático de miedo en base a la energía $\left(\frac{E}{100}\right)^2$. Esto evalúa si merece la pena alejarse del depredador en base a la energía que se posee en ese instante.

3. El denominador computa el número total de individuos actuales en el entorno. Esto permite evalúar si compensa tomar determinadas acciones o no en base a la población total.

4. La imagen de la función está acotada al intervalo $[-1, 1]$ en base a la concepción de que la energía máxima que posee cualquier especie es 100. Cabe destacar que un valor negativo se entiende como un riesgo alto y un valor positivo se entiende como el menor riesgo posible.

Una vez que se ha comprendido este análisis detallado, se pasa al siguiente punto, donde, a través de la TR, se resuelve la problemática inicialmente planteada.

**Solución de la evaluación de acciones y estados**

Para que se efectúe una evaluación de las acciones del agente en función de sus estados, se recurre a evalúar el riesgo efectivo de cometer acciones bajo un estado concreto. En este punto, se utiliza la función de riesgo total o TR para computar si una acción debe ser tomadada o no.

Finalmente, entendemos la evaluación de cada una de las acciones de un agente según su estado como un el valor que toma la TR.

## 2.2 Estimación de parámetros del modelo Lotka-volterra

Para estimar los coeficientes que modifican el comportamiento de las ecuaciones diferenciales, existen diversos métodos que permiten estimarlos de forma discretizada.

En este caso, como solo se dispone de información empírica sobre el número de presas y el número depredadores, se necesita buscar un método que realice dicha estimación.

Dicho esto, no merece la pena inventar un método nuevo que lo realice, por lo que se trabajará con el modelo propuesto por Kloppers, P. y Greeff, Johanna [6]
Para adaptar dicho trabajo se ha tenido las siguientes consideraciones:

- Existen 2 matrices que actúan de forma equivalente que la matriz $X$ en el artículo, denominadas $X$ e $Y$. La matriz X sirve para estimar los coeficientes de la primera ecuación diferencial y la matriz Y para estimar los coeficientes de la segunda ecuación diferencial.
- Estas matrices contienen por filas los siguientes valores:
  $\overline{x}_{j+1,j} = \dfrac{x(t_{j+1}) + x(t_j)}{2}$ y $\overline{xy}_{j+1,j} = \dfrac{x(t_{j+1})y(t_{j+1}) + x(t_j)y(t_j)}{2}$
- Para la matriz Y los valores son análogos siendo sutituido $\overline{x}_{j+1,j}$ por $\overline{y}_{j+1,j}$

# Capítulo 3

# Artículos relacionados

Algunos resultados teóricos plausibles del modelaje de la dinámica de poblaciones con aprendizaje por refuerzo serían los siguientes:

- A Study of AI Population Dynamics with Million-agent Reinforcement Learning
- Reinforcement learning in complementarity game and population dynamics
- Deep reinforcement learning and population dynamics for water systems control

# Referencias


[1] Krishnavedala Ian Alexander. Own work. URL: https://commons.wikimedia.org/w/index.php?curid=75212926.

[2] Krishnavedala. Own work. URL: https://commons.wikimedia.org/w/index.php?curid=75213940.

[3] Yaodong Yang, Lantao Yu, Yiwei Bai, Jun Wang, Weinan Zhang, Ying Wen, and Yong Yu. An empirical study of AI population dynamics with million-agent reinforcement learning. *CoRR*, abs/1709.04511, 2017.

[4] David Silver. Lectures on reinforcement learning. URL: https://www.davidsilver.uk/teaching/, 2015.

[5] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[6] P. Kloppers and Johanna Greeff. Lotka–volterra model parameter estimation using experiential data. *Applied Mathematics and Computation*, 224:817–825, 11 2013.

[7] Jost J;Li W;. Reinforcement learning in complementarity game and population dynamics. URL: https://pubmed.ncbi.nlm.nih.gov/25353428/.