Multi-agent learning

Introduction

Gerard Vreeswijk, Intelligent Software Systems, Computer Science Department, Faculty of Sciences, Utrecht University, The Netherlands.

Wednesday 28th April, 2021

Introduction

Introduction

- This master course focuses on multi-agent learning (MAL).

Introduction

- This master course focuses on multi-agent learning (MAL).
- A definition:

Introduction

- This master course focuses on multi-agent learning (MAL).
- A definition:

> Multi-agent learning is
> learning in the presence of other agents that learn.

Introduction

- This master course focuses on multi-agent learning (MAL).
- A definition:

> Multi-agent learning is
> learning in the presence of other agents that learn.

- Multi-agent learning is a discipline on the interface of multi-agent systems and machine learning.

Introduction

- This master course focuses on multi-agent learning (MAL).
- A definition:

> Multi-agent learning is
> learning in the presence of other agents that learn.

- Multi-agent learning is a discipline on the interface of multi-agent systems and machine learning.
- MAL differs from single-agent learning. Besides learning, it involves trying to influence other agents by executing the right actions (called teaching).

Introduction

- This master course focuses on multi-agent learning (MAL).
- A definition:

> Multi-agent learning is
> learning in the presence of other agents that learn.

- Multi-agent learning is a discipline on the interface of multi-agent systems and machine learning.
- MAL differs from single-agent learning. Besides learning, it involves trying to influence other agents by executing the right actions (called teaching).
- MAL is a young and active research area. Most contributions are from 2000 and later.

Some forms of machine learning

Some forms of machine learning

- **Statistical learning**. To predict or explain on the basis of "dead" data. The data may be { volatile / non-stationary / noisy }, etc. Nevertheless, the object of learning, the data, does not learn itself.

Some forms of machine learning

- **Statistical learning**. To predict or explain on the basis of "dead" data. The data may be { volatile / non-stationary / noisy }, etc. Nevertheless, the object of learning, the data, does not learn itself.
- **Robot learning**. To learn to function in an unknown and changing environment, possibly by generating one's own learning plan.

Some forms of machine learning

- **Statistical learning**. To predict or explain on the basis of "dead" data. The data may be { volatile / non-stationary / noisy }, etc. Nevertheless, the object of learning, the data, does not learn itself.
- **Robot learning**. To learn to function in an unknown and changing environment, possibly by generating one's own learning plan.
- **Multi-agent learning**. To learn from others who learn from you as well.

Some forms of machine learning

- **Statistical learning**. To predict or explain on the basis of "dead" data. The data may be { volatile / non-stationary / noisy }, etc. Nevertheless, the object of learning, the data, does not learn itself.
- **Robot learning**. To learn to function in an unknown and changing environment, possibly by generating one's own learning plan.
- **Multi-agent learning**. To learn from others who learn from you as well. H. Peyton Young in *Strategic Learning and its Limits* (2004):

Some forms of machine learning

- **Statistical learning**. To predict or explain on the basis of "dead" data. The data may be { volatile / non-stationary / noisy }, etc. Nevertheless, the object of learning, the data, does not learn itself.
- **Robot learning**. To learn to function in an unknown and changing environment, possibly by generating one's own learning plan.
- **Multi-agent learning**. To learn from others who learn from you as well. H. Peyton Young in *Strategic Learning and its Limits* (2004):

> "A social system consists of individuals who are learning about a process in which others are learning.

Some forms of machine learning

- **Statistical learning**. To predict or explain on the basis of "dead" data. The data may be { volatile / non-stationary / noisy }, etc. Nevertheless, the object of learning, the data, does not learn itself.
- **Robot learning**. To learn to function in an unknown and changing environment, possibly by generating one's own learning plan.
- **Multi-agent learning**. To learn from others who learn from you as well. H. Peyton Young in *Strategic Learning and its Limits* (2004):

> "A social system consists of individuals who are learning about a process in which others are learning. The system is self-referential.

Some forms of machine learning

- **Statistical learning**. To predict or explain on the basis of "dead" data. The data may be { volatile / non-stationary / noisy }, etc. Nevertheless, the object of learning, the data, does not learn itself.
- **Robot learning**. To learn to function in an unknown and changing environment, possibly by generating one's own learning plan.
- **Multi-agent learning**. To learn from others who learn from you as well. H. Peyton Young in *Strategic Learning and its Limits* (2004):

> "A social system consists of individuals who are learning about a process in which others are learning. The system is self-referential. Learning the true state of the system is therefore quite unlike learning the parameters that govern a physical process, for example

Some forms of machine learning

- **Statistical learning**. To predict or explain on the basis of "dead" data. The data may be { volatile / non-stationary / noisy }, etc. Nevertheless, the object of learning, the data, does not learn itself.
- **Robot learning**. To learn to function in an unknown and changing environment, possibly by generating one's own learning plan.
- **Multi-agent learning**. To learn from others who learn from you as well. H. Peyton Young in *Strategic Learning and its Limits* (2004):

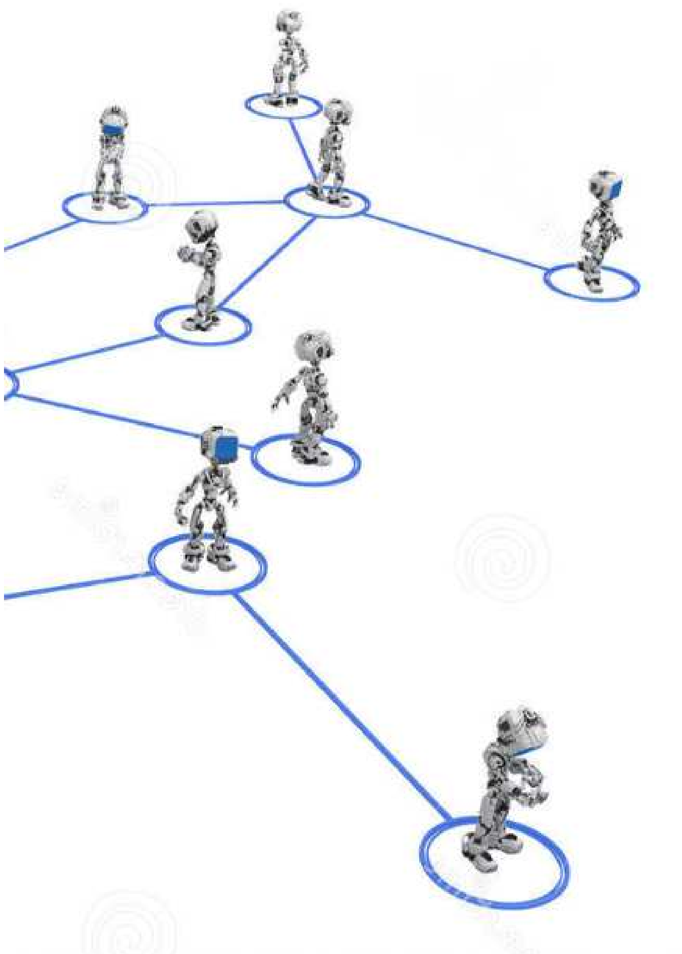
> "A social system consists of individuals who are learning about a process in which others are learning. The system is self-referential. Learning the true state of the system is therefore quite unlike learning the parameters that govern a physical process, for example, or even the parameters that describe a social process external to the observer.

Some forms of machine learning

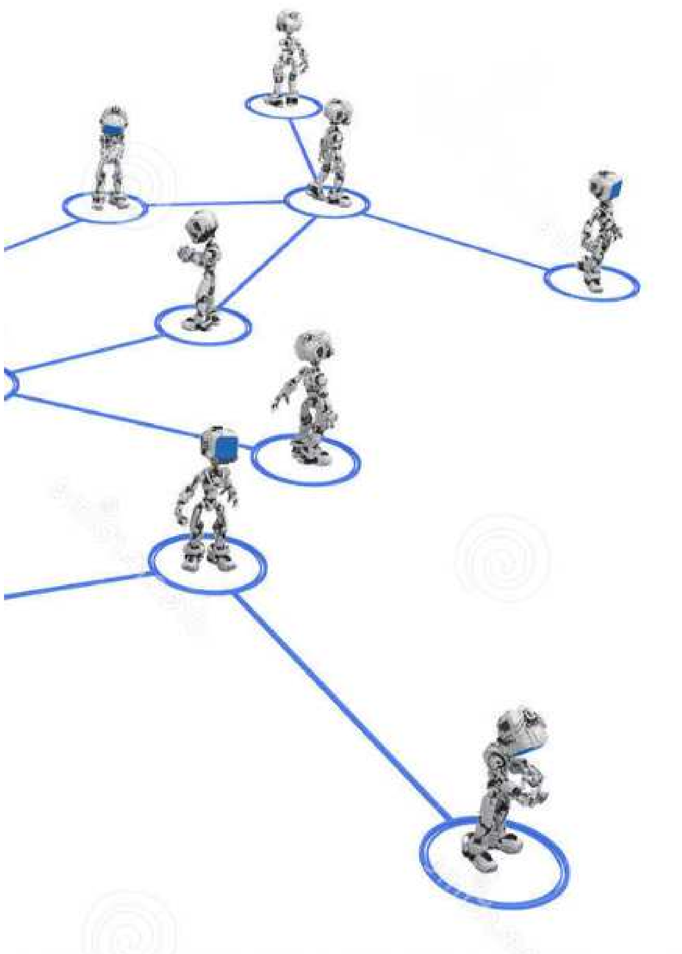
- **Statistical learning**. To predict or explain on the basis of "dead" data. The data may be { volatile / non-stationary / noisy }, etc. Nevertheless, the object of learning, the data, does not learn itself.
- **Robot learning**. To learn to function in an unknown and changing environment, possibly by generating one's own learning plan.
- **Multi-agent learning**. To learn from others who learn from you as well. H. Peyton Young in *Strategic Learning and its Limits* (2004):

> "A social system consists of individuals who are learning about a process in which others are learning. The system is self-referential. Learning the true state of the system is therefore quite unlike learning the parameters that govern a physical process, for example, or even the parameters that describe a social process external to the observer. When the observer is part of the system, the act of learning changes the thing to be learned."

This course

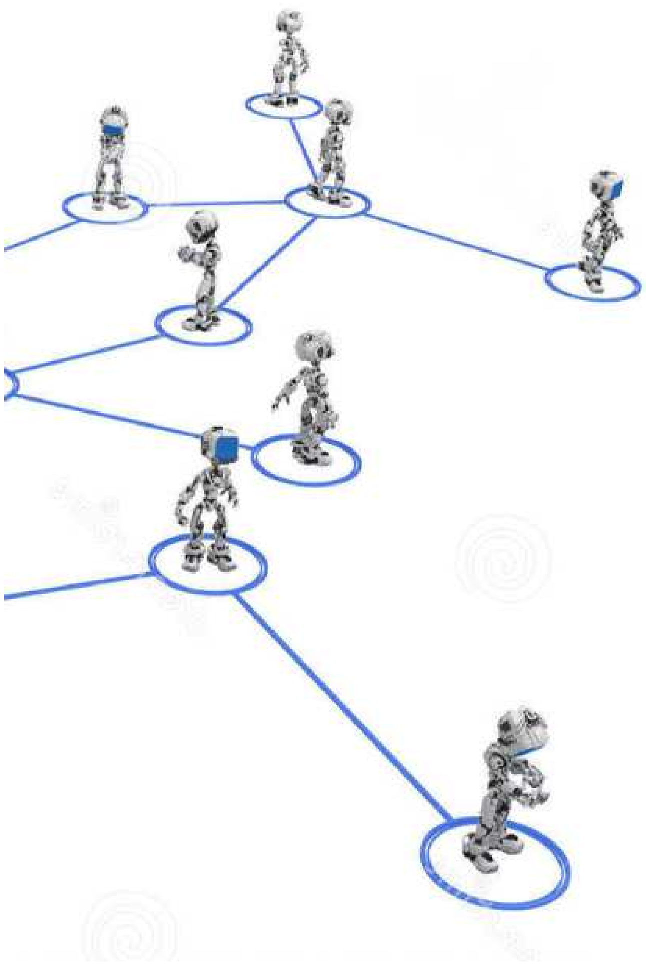


This course



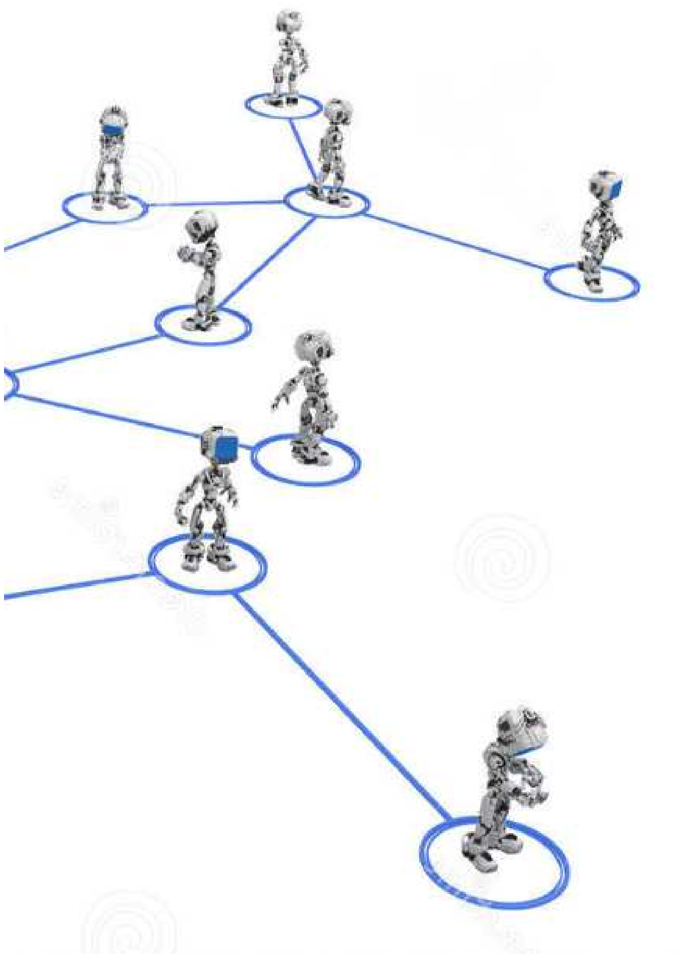
- This course discusses algorithms for machine learning that occur in multi-agent systems.

This course



- This course discusses algorithms for machine learning that occur in multi-agent systems.
- These algorithms operate on abstract models of strategic interaction between rational decision-makers.

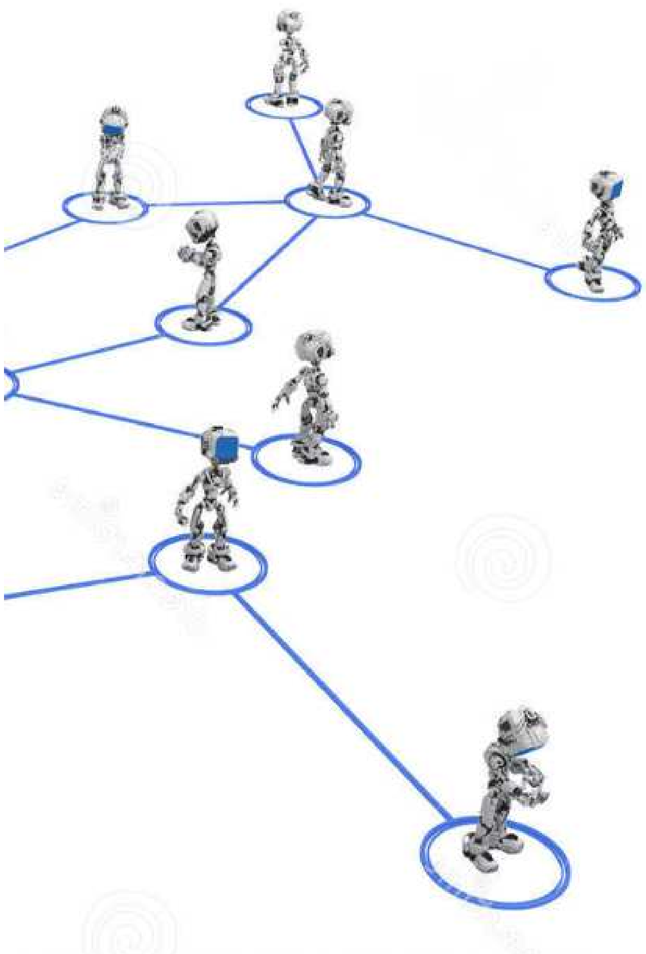
This course



- This course discusses algorithms for machine learning that occur in multi-agent systems.
- These algorithms operate on abstract models of strategic interaction between rational decision-makers.

  So knowledge of game theory helps.

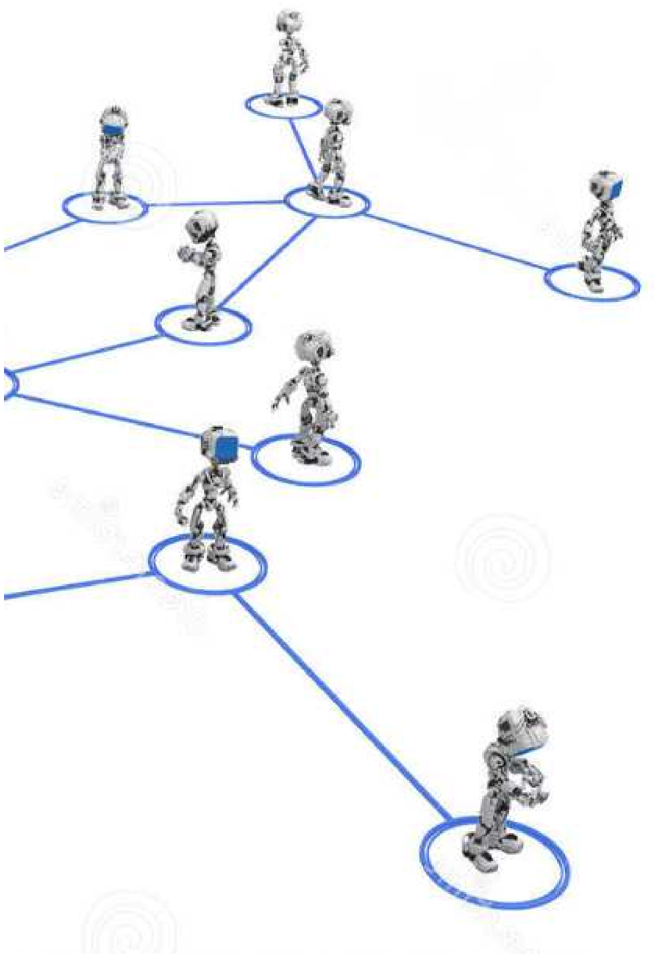
This course



- This course discusses algorithms for machine learning that occur in multi-agent systems.
- These algorithms operate on abstract models of strategic interaction between rational decision-makers.

  So knowledge of game theory helps.
- For a better understanding, you are asked to go over these algorithms yourself.

This course

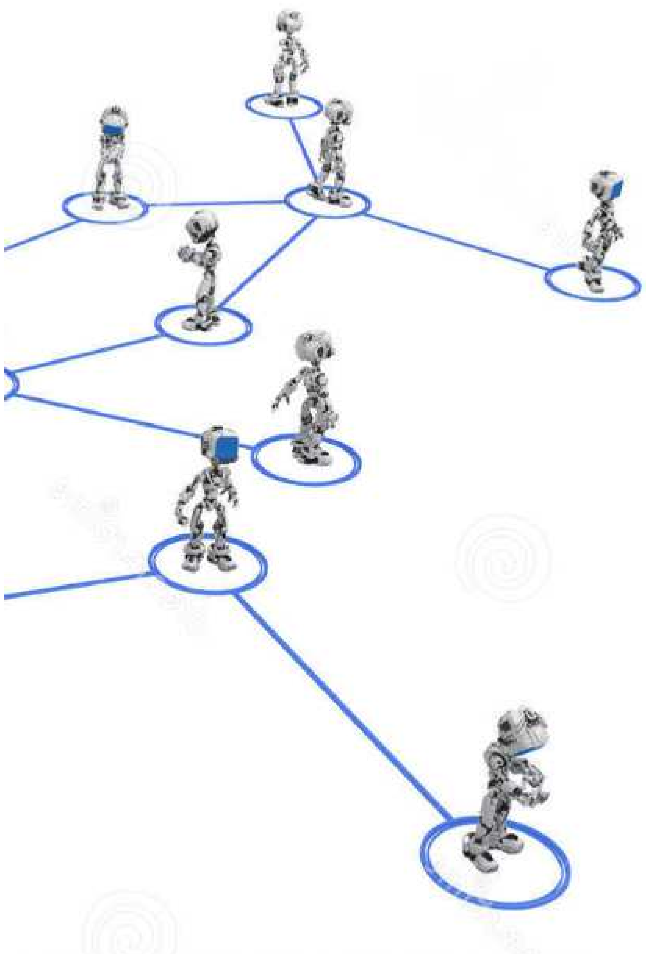


- This course discusses algorithms for machine learning that occur in multi-agent systems.
- These algorithms operate on abstract models of strategic interaction between rational decision-makers.

  So knowledge of game theory helps.
- For a better understanding, you are asked to go over these algorithms yourself.

  You are asked to implement these algorithms yourself.

This course

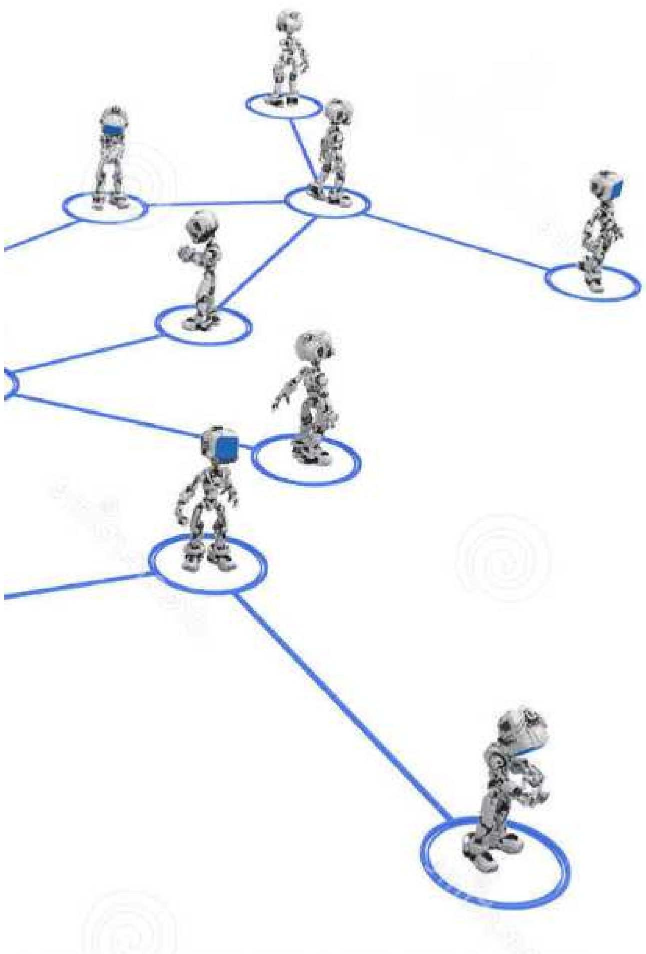


- This course discusses algorithms for machine learning that occur in multi-agent systems.
- These algorithms operate on abstract models of strategic interaction between rational decision-makers.

  So knowledge of game theory helps.
- For a better understanding, you are asked to go over these algorithms yourself.

  You are asked to implement these algorithms yourself.
- Observations and interaction often involves uncertainty.

This course



- This course discusses algorithms for machine learning that occur in multi-agent systems.
- These algorithms operate on abstract models of strategic interaction between rational decision-makers.

  So knowledge of game theory helps.
- For a better understanding, you are asked to go over these algorithms yourself.

  You are asked to implement these algorithms yourself.
- Observations and interaction often involves uncertainty.

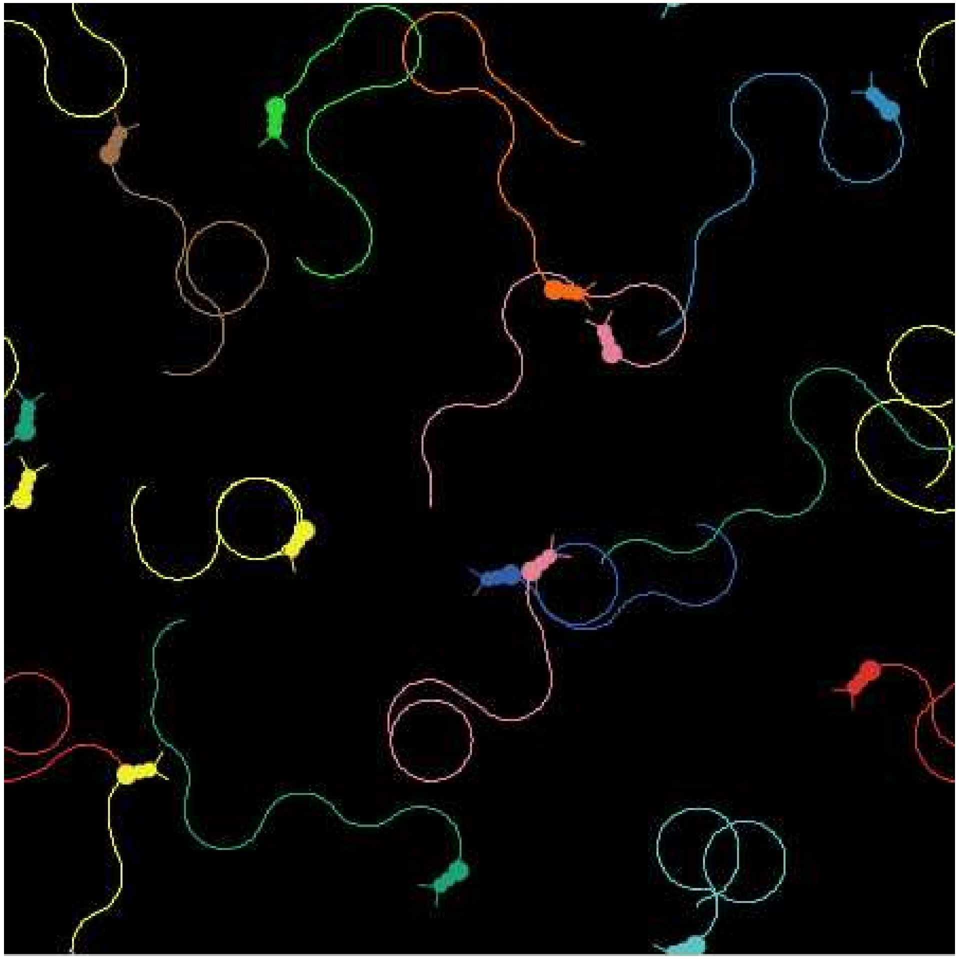
  So knowledge of probability theory helps.

Exam preparation

Example of MAL: a pursuit / evasion game

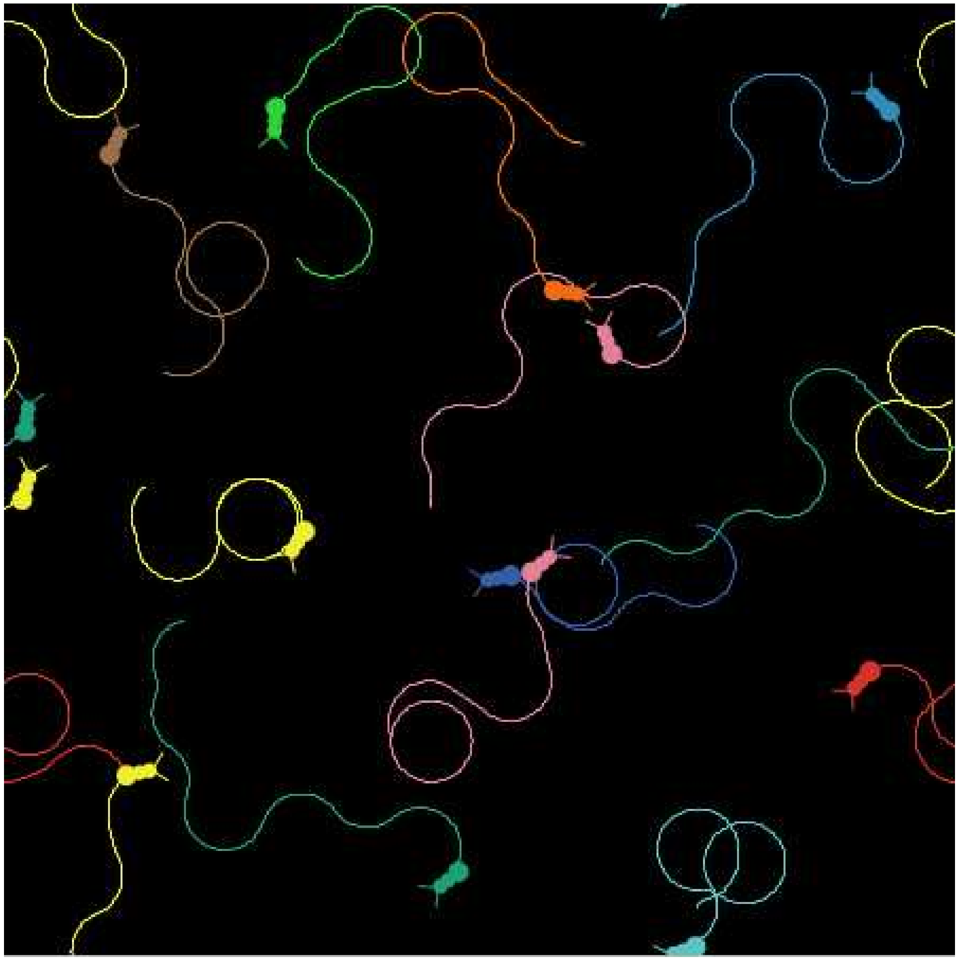
Example of MAL: a pursuit / evasion game

What is it?



Example of MAL: a pursuit / evasion game

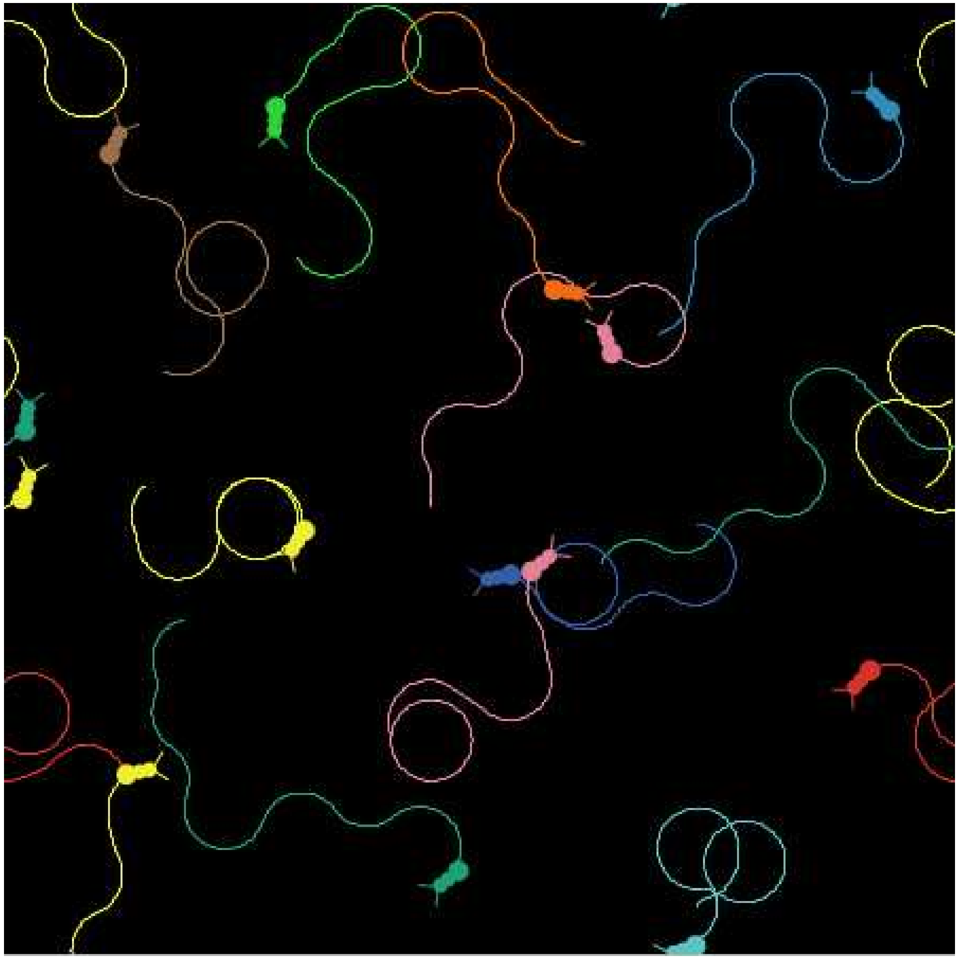
What is it?



- Name of the game: "Bug Hunt".

Example of MAL: a pursuit / evasion game

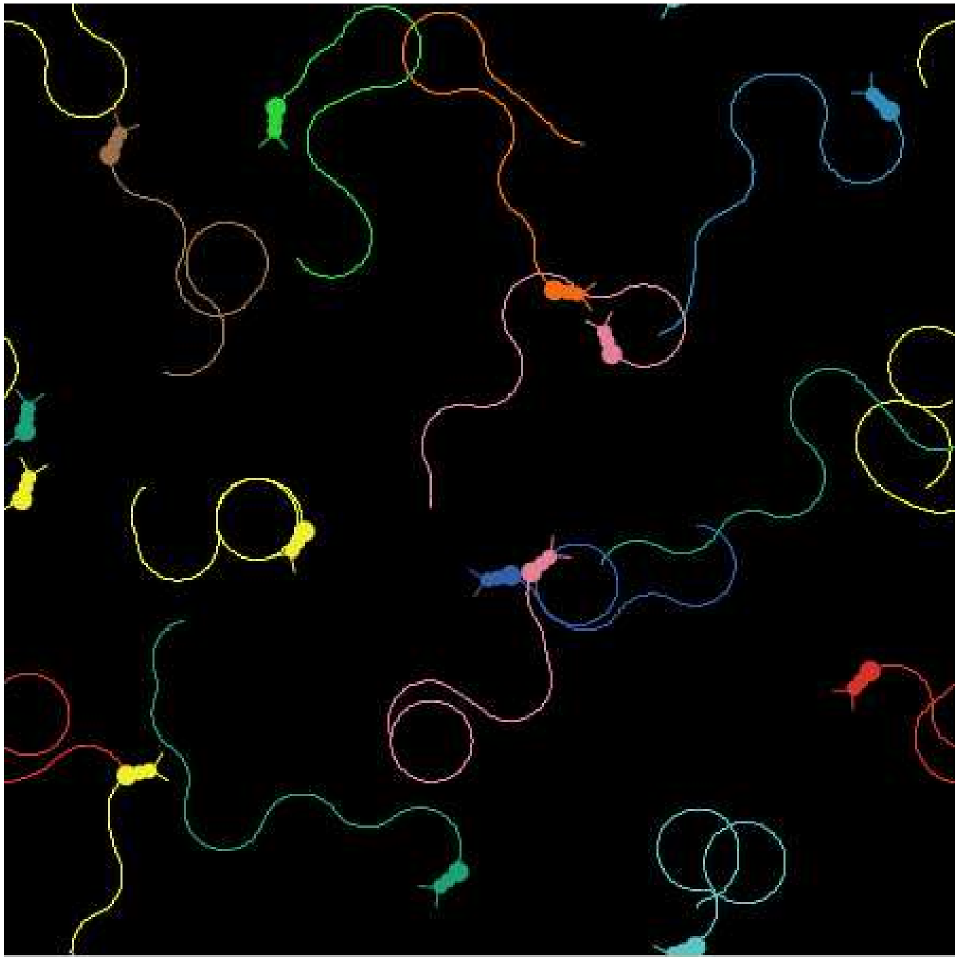
What is it?



- Name of the game: "Bug Hunt".
- n Bugs live on a torus.

Example of MAL: a pursuit / evasion game

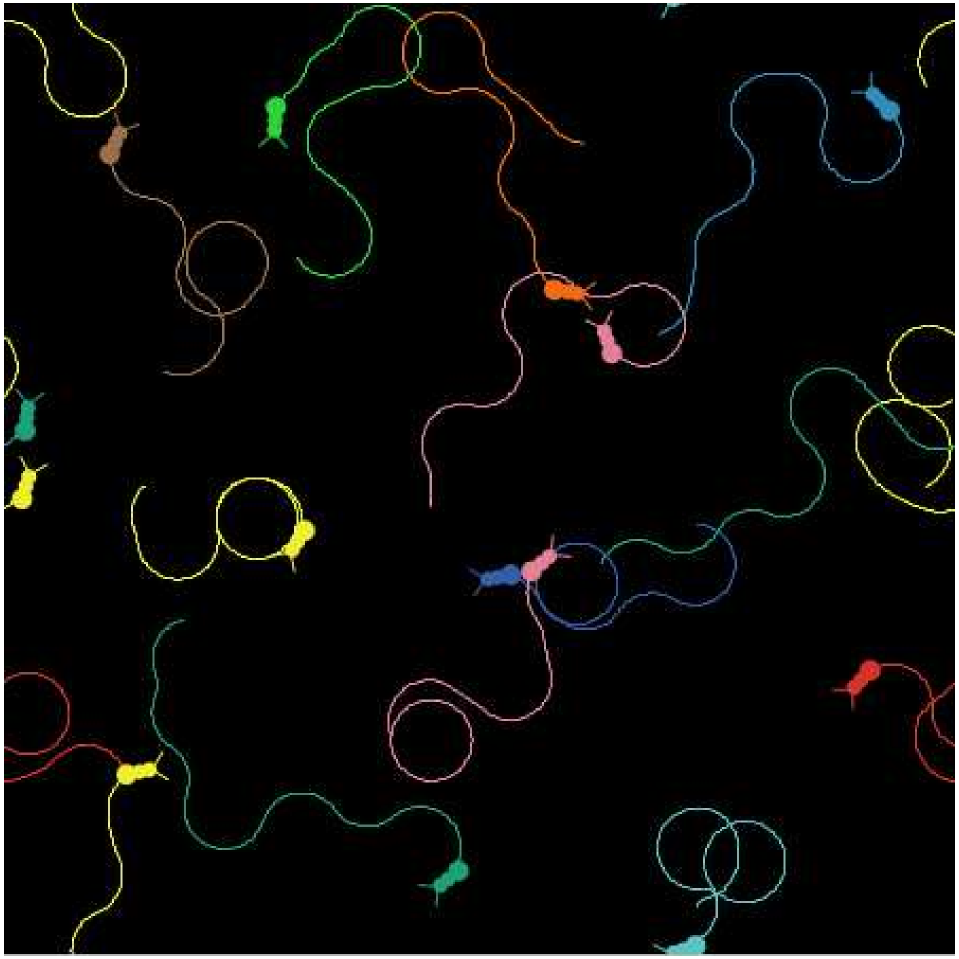
What is it?



- Name of the game: “Bug Hunt”.
- n Bugs live on a torus.
- They have, depending on the value of a global constant a, one of the following objectives

Example of MAL: a pursuit / evasion game

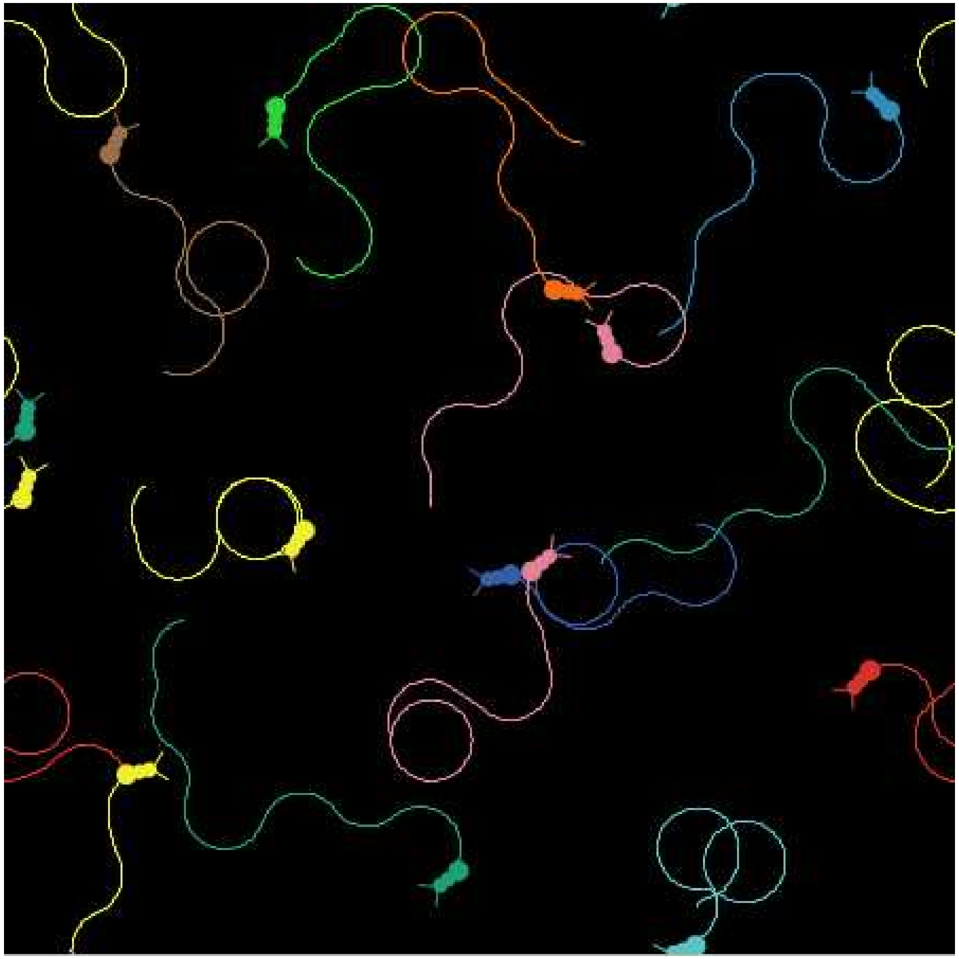
What is it?



- Name of the game: "Bug Hunt".
- n Bugs live on a torus.
- They have, depending on the value of a global constant a, one of the following objectives
  - $a = +1$: to pursue other bugs.

Example of MAL: a pursuit / evasion game

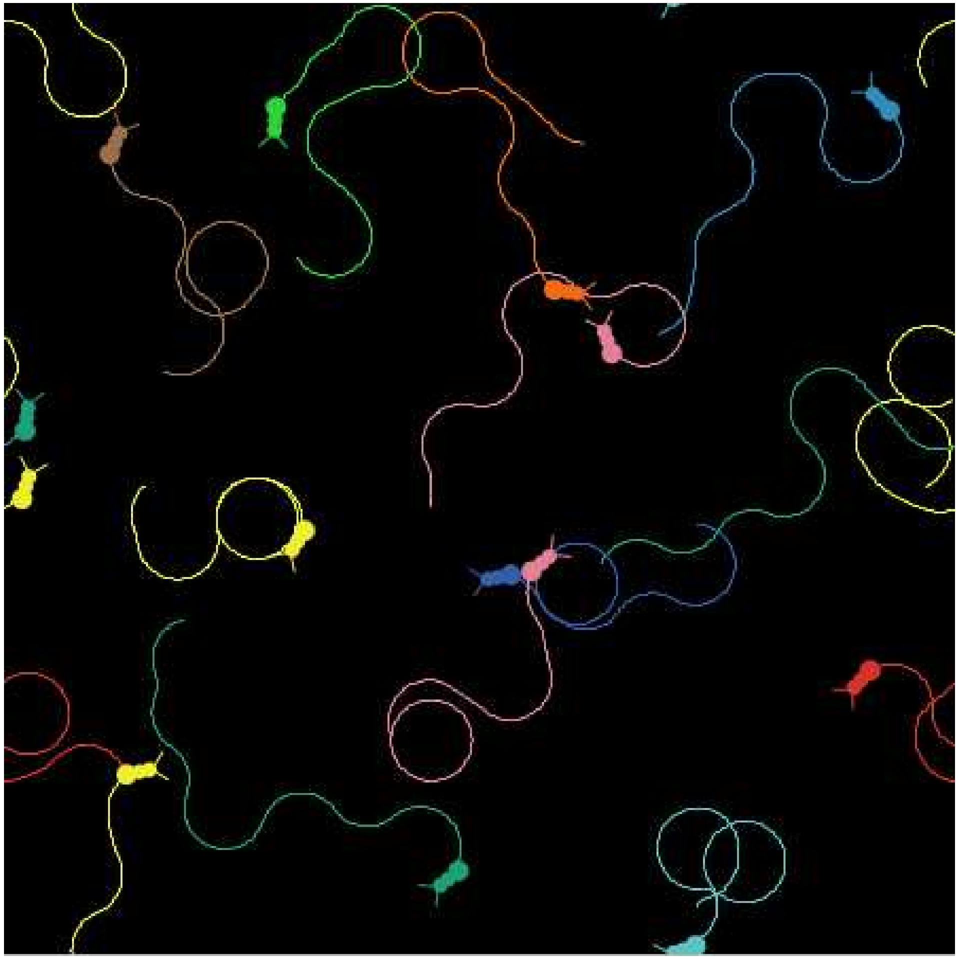
What is it?



- Name of the game: "Bug Hunt".
- n Bugs live on a torus.
- They have, depending on the value of a global constant a, one of the following objectives
  - $a = +1$: to pursue other bugs.
  - $a = -1$: to evade other bugs.

Example of MAL: a pursuit / evasion game

What is it?



- Name of the game: "Bug Hunt".
- n Bugs live on a torus.
- They have, depending on the value of a global constant a, one of the following objectives
  - $a = +1$: to pursue other bugs.
  - $a = -1$: to evade other bugs.
- Bug Hunt is a simple instance of a so-called n-type pursuit ($a = 1$) or evasion ($a = -1$) game.

Example: Pursuit / evasion game

How does it work?

Example: Pursuit / evasion game

How does it work?

- Algorithm:

Example: Pursuit / evasion game

How does it work?

- Algorithm:

1. Determine the location of k nearest bugs.

Example: Pursuit / evasion game

How does it work?

- Algorithm:

  1. Determine the location of k nearest bugs.
  2. Based on this observation,
     turn left d degrees or turn right d degrees

Example: Pursuit / evasion game

How does it work?

- Algorithm:

1. Determine the location of k nearest bugs.
2. Based on this observation,
   turn left d degrees or turn right d degrees
3. Move forward s.

Example: Pursuit / evasion game

How does it work?

- Algorithm:

  1. Determine the location of k nearest bugs.
  2. Based on this observation,
     turn left d degrees or turn right d degrees
  3. Move forward s.

- Global constants:

Example: Pursuit / evasion game

How does it work?

- Algorithm:

  1. Determine the location of k nearest bugs.
  2. Based on this observation,
     turn left d degrees or turn right d degrees
  3. Move forward s.

- Global constants:

  n : number of bugs

Example: Pursuit / evasion game

How does it work?

- Algorithm:

  1. Determine the location of k nearest bugs.
  2. Based on this observation,
     turn left d degrees or turn right d degrees
  3. Move forward s.

- Global constants:

$$\begin{aligned} n &: \text{ number of bugs} \\ a &: \text{ pursue } (1) \text{ or evade } (-1) \end{aligned}$$

Example: Pursuit / evasion game

How does it work?

- Algorithm:

  1. Determine the location of k nearest bugs.
  2. Based on this observation, turn left d degrees or turn right d degrees
  3. Move forward s.

- Global constants:

$$\begin{aligned} n &: \text{ number of bugs} \\ a &: \text{ pursue } (1) \text{ or evade } (-1) \\ k &: \text{ number of nearest bugs to observe} \end{aligned}$$

Example: Pursuit / evasion game

How does it work?

- Algorithm:

  1. Determine the location of k nearest bugs.
  2. Based on this observation,
     turn left d degrees or turn right d degrees
  3. Move forward s.

- Global constants:

$$\begin{aligned} n &: \text{ number of bugs} \\ a &: \text{ pursue } (1) \text{ or evade } (-1) \\ k &: \text{ number of nearest bugs to observe } \ (0 \leq k \leq n) \end{aligned}$$

Example: Pursuit / evasion game

How does it work?

- Algorithm:

  1. Determine the location of k nearest bugs.
  2. Based on this observation,
     turn left d degrees or turn right d degrees
  3. Move forward s.

- Global constants:

  n : number of bugs
  a : pursue (1) or evade (-1)
  k : number of nearest bugs to observe ($0 \leq k \leq n$)
  d : number of degrees to turn left or right

Example: Pursuit / evasion game

How does it work?

- Algorithm:

1. Determine the location of k nearest bugs.
2. Based on this observation,
   turn left d degrees or turn right d degrees
3. Move forward s.

- Global constants:

n : number of bugs
a : pursue (1) or evade (-1)
k : number of nearest bugs to observe ($0 \leq k \leq n$)
d : number of degrees to turn left or right
s : step size

Example: Pursuit / evasion game

Neighbours. The set of k nearest neighbours is determined as follows.

Example: Pursuit / evasion game

Neighbours. The set of k nearest neighbours is determined as follows.

- A bug determines the set of k nearest neighbours, K.

Example: Pursuit / evasion game

Neighbours. The set of k nearest neighbours is determined as follows.

- A bug determines the set of k nearest neighbours, K.
- Then the centroid (center of gravity) of K is determined.

Example: Pursuit / evasion game

Neighbours. The set of k nearest neighbours is determined as follows.

- A bug determines the set of k nearest neighbours, K.
- Then the centroid (center of gravity) of K is determined. The centroid will then be the waypoint for the present bug.

Example: Pursuit / evasion game

Neighbours. The set of k nearest neighbours is determined as follows.

- A bug determines the set of k nearest neighbours, K.
- Then the centroid (center of gravity) of K is determined. The centroid will then be the waypoint for the present bug.
- If the centroid is on the left to the present bug, and the objective is to pursue other bugs, then the bug will turn to the left.

Example: Pursuit / evasion game

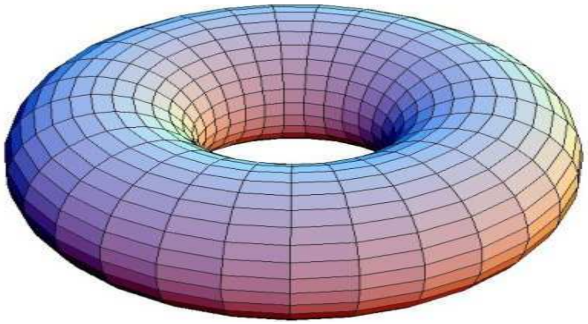
Neighbours. The set of k nearest neighbours is determined as follows.

- A bug determines the set of k nearest neighbours, K.
- Then the centroid (center of gravity) of K is determined. The centroid will then be the waypoint for the present bug.
- If the centroid is on the left to the present bug, and the objective is to pursue other bugs, then the bug will turn to the left. (If the objective is to evade other bugs, then the bug will turn to the right.)

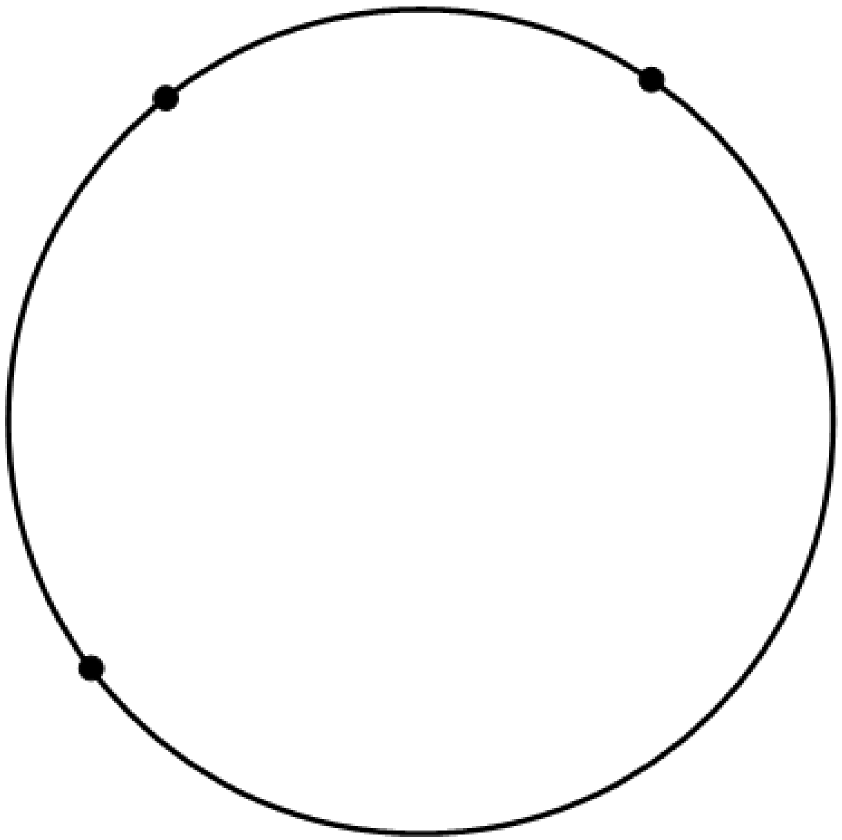
Example: Pursuit / evasion game

Neighbours. The set of k nearest neighbours is determined as follows.

- A bug determines the set of k nearest neighbours, K.
- Then the centroid (center of gravity) of K is determined. The centroid will then be the waypoint for the present bug.
- If the centroid is on the left to the present bug, and the objective is to pursue other bugs, then the bug will turn to the left. (If the objective is to evade other bugs, then the bug will turn to the right.)
- The centroid of k points on a torus is defined as in a 2D plane, but we will have to take into account that the neighborhood and direction on a torus are defined differently.



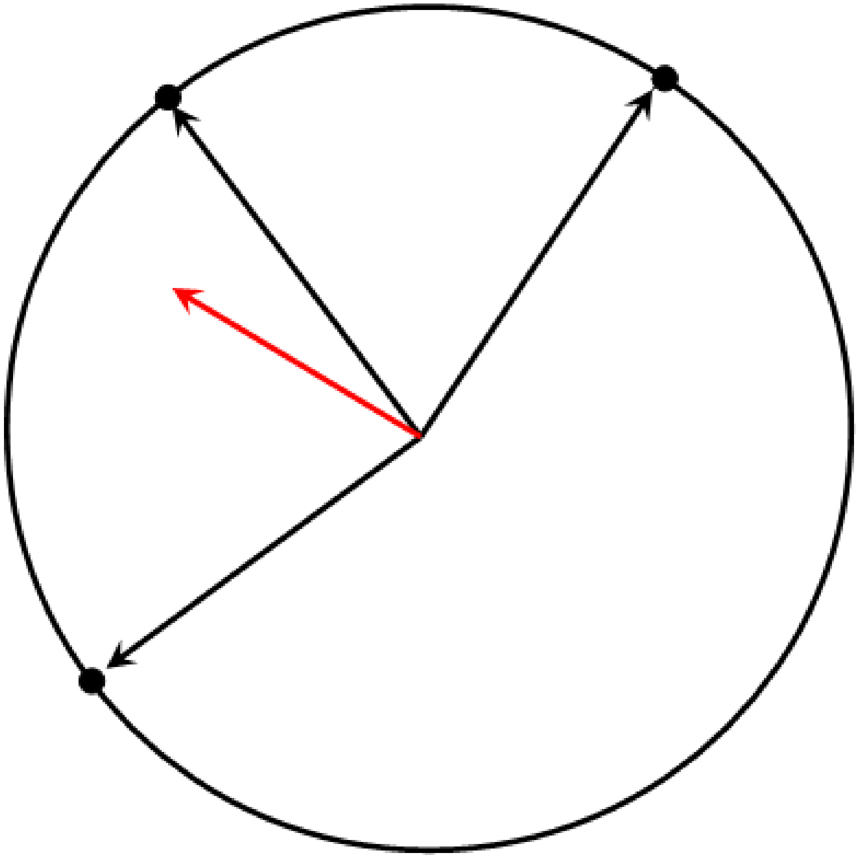
Example: Pursuit / evasion game



The centroid of points on a torus is computed in the same way as the centroid of points on a circle is computed.

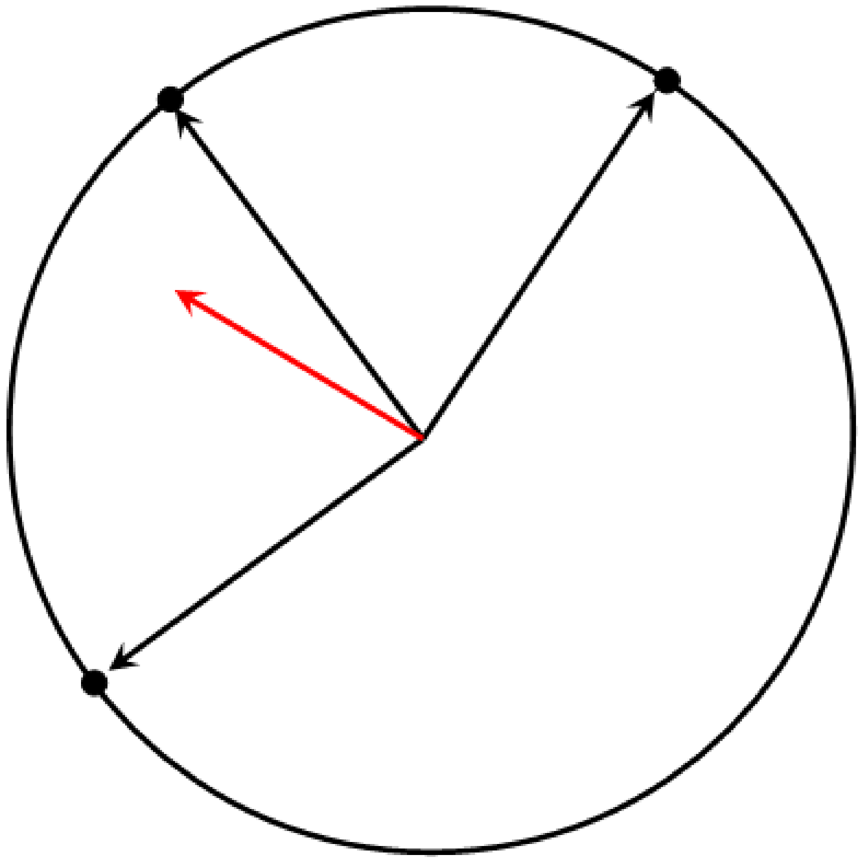
Let us consider the latter.

Example: Pursuit / evasion game



The average of a number of directions is computed by representing degrees as vectors, and then add the vectors.

Example: Pursuit / evasion game



The average of a number of directions is computed by representing degrees as vectors, and then add the vectors.

Then map the sum of the vectors back to the circle. The certainty factor of this average is the length of the sum vector.

Example: Pursuit / evasion game

Why?

Example: Pursuit / evasion game

Why?

- **Motivation**: It would be nice to start this course with a simple visual example of multi-agent learning.

Example: Pursuit / evasion game

Why?

- **Motivation**: It would be nice to start this course with a simple visual example of multi-agent learning.
- **Challenge**: produce a simplest such visual example.

Example: Pursuit / evasion game

Why?

- **Motivation**: It would be nice to start this course with a simple visual example of multi-agent learning.
- **Challenge**: produce a simplest such visual example.
- **Alternatives** (and their disadvantages):

Example: Pursuit / evasion game

Why?

- **Motivation**: It would be nice to start this course with a simple visual example of multi-agent learning.
- **Challenge**: produce a simplest such visual example.
- **Alternatives** (and their disadvantages):
  - Unit circle (Unbounded 1D). Bug traces would be difficult to see. (Bugs frequently re-visit locations in 1D.)

Example: Pursuit / evasion game

Why?

- **Motivation**: It would be nice to start this course with a simple visual example of multi-agent learning.
- **Challenge**: produce a simplest such visual example.
- **Alternatives** (and their disadvantages):
  - Unit circle (Unbounded 1D). Bug traces would be difficult to see. (Bugs frequently re-visit locations in 1D.)
  - Square (Bounded 2D). Boundaries would influence avoidance behaviour. (Be pushed into corners.)

Example: Pursuit / evasion game

Why?

- **Motivation**: It would be nice to start this course with a simple visual example of multi-agent learning.
- **Challenge**: produce a simplest such visual example.
- **Alternatives** (and their disadvantages):
  - Unit circle (Unbounded 1D). Bug traces would be difficult to see. (Bugs frequently re-visit locations in 1D.)
  - Square (Bounded 2D). Boundaries would influence avoidance behaviour. (Be pushed into corners.)
  - The plane (Unbounded 2D). The absence of boundaries would enable simple evading behaviour. (Just flee into open space.)

Example: Pursuit / evasion game

Literature on n-person differential games:

Example: Pursuit / evasion game

Literature on n-person differential games:

- Isaacs (1965): *Differential games*.
- Friedman (1971): *Differential games*.

Example: Pursuit / evasion game

Literature on n-person differential games:

- Isaacs (1965): *Differential games*.
- Friedman (1971): *Differential games*.

The last chapter of Friedman, Ch. 8, is on n-person games

Example: Pursuit / evasion game

Literature on n-person differential games:

- Isaacs (1965): *Differential games*.
- Friedman (1971): *Differential games*.

The last chapter of Friedman, Ch. 8, is on n-person games: “open loop Nash equilibria exist for n-person differential games when there are integral bounds on the control functions”.

Example: Pursuit / evasion game

Literature on n-person differential games:

- Isaacs (1965): *Differential games*.
- Friedman (1971): *Differential games*.

The last chapter of Friedman, Ch. 8, is on n-person games: "open loop Nash equilibria exist for n-person differential games when there are integral bounds on the control functions".

- H. Stalford *et al.* (1973): "Sufficiency conditions for Nash equilibria in N-person differential games" in: *Topics in Differential Games*, Elsevier.
- Hájek (1975): *Pursuit and evasion games.*
- *D. Fudenberg et al. (1988): "Open-Loop and Closed-Loop Equilibria in Dynamic Games with Many players," in J. of Economic Theory* ***44****(1). pp. 1-18.*

Example: Pursuit / evasion game

Literature on n-person differential games:

- Isaacs (1965): *Differential games*.
- Friedman (1971): *Differential games*.

The last chapter of Friedman, Ch. 8, is on n-person games: “open loop Nash equilibria exist for n-person differential games when there are integral bounds on the control functions”.

- H. Stalford *et al.* (1973): “Sufficiency conditions for Nash equilibria in N-person differential games” in: *Topics in Differential Games*, Elsevier.
- Hájek (1975): *Pursuit and evasion games.*
- *D. Fudenberg et al. (1988): “Open-Loop and Closed-Loop Equilibria in Dynamic Games with Many players,” in J. of Economic Theory* ***44****(1). pp. 1-18.*

Warning. Work on DG’s is highly analytical (rather than philosophical, conceptual, or empirical).

Do MAL yourself: “aye” or “nay”

Do MAL yourself: “aye” or “nay”

Do MAL yourself: “aye” or “nay”

- Play in rounds, with four players. For each round there are two possible actions: 1 (raise hand) or 0 (do not). Reward per round: the number of persons with identical actions.

Do MAL yourself: “aye” or “nay”

- Play in rounds, with four players. For each round there are two possible actions: 1 (raise hand) or 0 (do not). Reward per round: the number of persons with identical actions.
- Same, except reward per round = the number of persons with opposite actions.

Do MAL yourself: “aye” or “nay”

- Play in rounds, with four players. For each round there are two possible actions: 1 (raise hand) or 0 (do not). Reward per round: the number of persons with identical actions.
- Same, except reward per round = the number of persons with opposite actions.
- Tragedy of the commons.

Do MAL yourself: "aye" or "nay"

- Play in rounds, with four players. For each round there are two possible actions: 1 (raise hand) or 0 (do not). Reward per round: the number of persons with identical actions.
- Same, except reward per round = the number of persons with opposite actions.
- Tragedy of the commons.
  - Action 1: solicit for a reward; action 0: do not solicit.

Do MAL yourself: "aye" or "nay"

- Play in rounds, with four players. For each round there are two possible actions: 1 (raise hand) or 0 (do not). Reward per round: the number of persons with identical actions.
- Same, except reward per round = the number of persons with opposite actions.
- Tragedy of the commons.
  - Action 1: solicit for a reward; action 0: do not solicit.
  - Reward: one if action 1 and no more than three action 1, else zero.

Do MAL yourself: adaptation in a massive MAS

Do MAL yourself: adaptation in a massive MAS

- Game with ~ 80 players, in rounds. Each round, you have two actions: sit or stand. Reward per round: the number of persons with identical actions.

Do MAL yourself: adaptation in a massive MAS

- Game with ~ 80 players, in rounds. Each round, you have two actions: sit or stand. Reward per round: the number of persons with identical actions.

- Same, except reward per round = the number of persons with opposite actions.

Do MAL yourself: adaptation in a massive MAS

- Game with ~ 80 players, in rounds. Each round, you have two actions: sit or stand. Reward per round: the number of persons with identical actions.
- Same, except reward per round = the number of persons with opposite actions.
- Tragedy of the commons.

Do MAL yourself: adaptation in a massive MAS

- Game with ~ 80 players, in rounds. Each round, you have two actions: sit or stand. Reward per round: the number of persons with identical actions.
- Same, except reward per round = the number of persons with opposite actions.
- Tragedy of the commons.
  - Stand: solicit for a reward ("fish"); sit: do not solicit.

Do MAL yourself: adaptation in a massive MAS

- Game with ~ 80 players, in rounds. Each round, you have two actions: sit or stand. Reward per round: the number of persons with identical actions.
- Same, except reward per round = the number of persons with opposite actions.
- Tragedy of the commons.
  - Stand: solicit for a reward ("fish"); sit: do not solicit.
  - Reward: 1 if stand and no more than 10 stand, else zero.

Intro's by Peyton Young, and Shoham & Leyton-Brown

Intro’s by Peyton Young, and Shoham & Leyton-Brown

- Chapter 1 “Introduction” in *Strategic Learning and its Limits* by H. Peyton Young (2004).

Intro's by Peyton Young, and Shoham & Leyton-Brown

- Chapter 1 "Introduction" in *Strategic Learning and its Limits* by H. Peyton Young (2004).
- Chapter 7 "Learning and Teaching", Sec. 7.1 in *Multi-agent systems*, by Shoham & Leyton-Brown (2009).

Intro's by Peyton Young, and Shoham & Leyton-Brown

- Chapter 1 "Introduction" in *Strategic Learning and its Limits* by H. Peyton Young (2004).
- Chapter 7 "Learning and Teaching", Sec. 7.1 in *Multi-agent systems*, by Shoham & Leyton-Brown (2009).

Multiagent learning is fraught with subtleties.

Intro’s by Peyton Young, and Shoham & Leyton-Brown

- Chapter 1 “Introduction” in *Strategic Learning and its Limits* by H. Peyton Young (2004).
- Chapter 7 “Learning and Teaching”, Sec. 7.1 in *Multi-agent systems*, by Shoham & Leyton-Brown (2009).

Multiagent learning is fraught with subtleties.

Three issues:

Intro's by Peyton Young, and Shoham & Leyton-Brown

- Chapter 1 "Introduction" in *Strategic Learning and its Limits* by H. Peyton Young (2004).
- Chapter 7 "Learning and Teaching", Sec. 7.1 in *Multi-agent systems*, by Shoham & Leyton-Brown (2009).

Multiagent learning is fraught with subtleties.

Three issues:

1. The interaction between learning and teaching.

Intro's by Peyton Young, and Shoham & Leyton-Brown

- Chapter 1 "Introduction" in *Strategic Learning and its Limits* by H. Peyton Young (2004).
- Chapter 7 "Learning and Teaching", Sec. 7.1 in *Multi-agent systems*, by Shoham & Leyton-Brown (2009).

Multiagent learning is fraught with subtleties.

Three issues:

1. The interaction between learning and teaching.
2. The settings in which learning takes place and what constitutes learning in those settings.

Intro's by Peyton Young, and Shoham & Leyton-Brown

- Chapter 1 "Introduction" in *Strategic Learning and its Limits* by H. Peyton Young (2004).
- Chapter 7 "Learning and Teaching", Sec. 7.1 in *Multi-agent systems*, by Shoham & Leyton-Brown (2009).

Multiagent learning is fraught with subtleties.

Three issues:

1. The interaction between learning and teaching.
2. The settings in which learning takes place and what constitutes learning in those settings.
3. The yardsticks by which to measure theories of learning in multiagent systems.

Multi-agent learning involves teaching

Example. Suppose the following game is repeated infinitely many times. Each party accumulates its payoffs.

	L	R
T	1, 0	3, 2
B	2, 1	4, 0

Multi-agent learning involves teaching

Example. Suppose the following game is repeated infinitely many times. Each party accumulates its payoffs.

	L	R
T	1, 0	3, 2
B	2, 1	4, 0

- Action B is dominant for the row player.

Multi-agent learning involves teaching

Example. Suppose the following game is repeated infinitely many times. Each party accumulates its payoffs.

	L	R
T	1, 0	3, 2
B	2, 1	4, 0

- Action B is dominant for the row player.
- If row chooses B, column will choose L.

Multi-agent learning involves teaching

Example. Suppose the following game is repeated infinitely many times. Each party accumulates its payoffs.

	L	R
T	1, 0	3, 2
B	2, 1	4, 0

- Action B is dominant for the row player.
- If row chooses B, column will choose L.
- **Problem**: this (single pure) NE is Pareto-dominated by the action profile (T, R).

Multi-agent learning involves teaching

Example. Suppose the following game is repeated infinitely many times. Each party accumulates its payoffs.

	L	R
T	1,0	3,2
B	2,1	4,0

- Action B is dominant for the row player.
- If row chooses B, column will choose L.
- **Problem**: this (single pure) NE is Pareto-dominated by the action profile (T, R).
- **Solution**: row can teach col by playing T throughout.

Multi-agent learning involves teaching

Example. Suppose the following game is repeated infinitely many times. Each party accumulates its payoffs.

	L	R
T	1, 0	3, 2
B	2, 1	4, 0

- Action B is dominant for the row player.
- If row chooses B, column will choose L.
- **Problem**: this (single pure) NE is Pareto-dominated by the action profile (T, R).
- **Solution**: row can teach col by playing T throughout.
- If col has any sense he/she/it will pick up the signal and play R.

It is not always clear who should teach, and when

Consider the coordination game:

	L	R
L	$1, 1$	$-1, -1$
R	$-1, -1$	$1, 1$

It is not always clear who should teach, and when

Consider the coordination game:

	L	R
L	$1, 1$	$-1, -1$
R	$-1, -1$	$1, 1$

It is not always clear who should teach, and when

Consider the coordination game:

	L	R
L	$1, 1$	$-1, -1$
R	$-1, -1$	$1, 1$

- Suppose row and col are passive followers (later: fictitious play).

It is not always clear who should teach, and when

Consider the coordination game:

	L	R
L	$1, 1$	$-1, -1$
R	$-1, -1$	$1, 1$

- Suppose row and col are passive followers (later: fictitious play). If they start mis-coordinating, they may alternate, hence mis-coordinate, endlessly.

It is not always clear who should teach, and when

Consider the coordination game:

	L	R
L	1, 1	−1, −1
R	−1, −1	1, 1

- Suppose row and col are passive followers (later: fictitious play). If they start mis-coordinating, they may alternate, hence mis-coordinate, endlessly.
- Suppose row and col are adamant teachers (later: bully). If they start mis-coordinating, they will remain to do so.

It is not always clear who should teach, and when

Consider the coordination game:

	L	R
L	$1, 1$	$-1, -1$
R	$-1, -1$	$1, 1$

- Suppose row and col are passive followers (later: fictitious play). If they start mis-coordinating, they may alternate, hence mis-coordinate, endlessly.
- Suppose row and col are adamant teachers (later: bully). If they start mis-coordinating, they will remain to do so.
- Is there a learning rule that ensures coordination without an external co-ordinator?

When is a learning algorithm successful?

Game of Chicken:

	S	D
S	$0, 0$	$-1, 1$
D	$1, -1$	$-9, -9$

When is a learning algorithm successful?

Game of Chicken:

	S	D
S	$0, 0$	$-1, 1$
D	$1, -1$	$-9, -9$

- In the presence of an aggressive stubborn opponent ("watch out, I'm crazy!"), it's best to play safe and adapt.

When is a learning algorithm successful?

Game of Chicken:

	S	D
S	$0, 0$	$-1, 1$
D	$1, -1$	$-9, -9$

- In the presence of an aggressive stubborn opponent ("watch out, I'm crazy!"), it's best to play safe and adapt.
- In the presence of an adapting opponent, it is best to bully the opponent.

When is a learning algorithm successful?

Game of Chicken:

	S	D
S	$0, 0$	$-1, 1$
D	$1, -1$	$-9, -9$

- In the presence of an aggressive stubborn opponent (“watch out, I’m crazy!”), it’s best to play safe and adapt.
- In the presence of an adapting opponent, it is best to bully the opponent.
- Best strategy depends on the opponent!

When is a learning algorithm successful?

Game of Chicken:

	S	D
S	$0, 0$	$-1, 1$
D	$1, -1$	$-9, -9$

- In the presence of an aggressive stubborn opponent ("watch out, I'm crazy!"), it's best to play safe and adapt.
- In the presence of an adapting opponent, it is best to bully the opponent.
- Best strategy depends on the opponent!
- **Possible solution**: probe your opponent first to find out whether he is stubborn aggressive or an adapter.

When is a learning algorithm successful?

Game of Chicken:

	S	D
S	$0,0$	$-1,1$
D	$1,-1$	$-9,-9$

- In the presence of an aggressive stubborn opponent ("watch out, I'm crazy!"), it's best to play safe and adapt.
- In the presence of an adapting opponent, it is best to bully the opponent.
- Best strategy depends on the opponent!
- **Possible solution**: probe your opponent first to find out whether he is stubborn aggressive or an adapter.
- **Problem**: it's not that simple.

When is a learning algorithm successful?

Game of Chicken:

	S	D
S	$0, 0$	$-1, 1$
D	$1, -1$	$-9, -9$

- In the presence of an aggressive stubborn opponent ("watch out, I'm crazy!"), it's best to play safe and adapt.
- In the presence of an adapting opponent, it is best to bully the opponent.
- Best strategy depends on the opponent!
- **Possible solution**: probe your opponent first to find out whether he is stubborn aggressive or an adapter.
- **Problem**: it's not that simple. Opponents may behave in complex ways

When is a learning algorithm successful?

Game of Chicken:

	S	D
S	$0, 0$	$-1, 1$
D	$1, -1$	$-9, -9$

- In the presence of an aggressive stubborn opponent ("watch out, I'm crazy!"), it's best to play safe and adapt.
- In the presence of an adapting opponent, it is best to bully the opponent.
- Best strategy depends on the opponent!
- **Possible solution**: probe your opponent first to find out whether he is stubborn aggressive or an adapter.
- **Problem**: it's not that simple. Opponents may behave in complex ways and opponents may differ.

Models of MAL: descriptive models and prescriptive models

Descriptive models of multi-agent learning

Descriptive. Describe what happens in existing processes.

Descriptive models of multi-agent learning

Descriptive. Describe what happens in existing processes.

Typically social or economic processes.

Descriptive models of multi-agent learning

Descriptive. Describe what happens in existing processes.

Typically social or economic processes.

Ideally, the formal model should ...

Descriptive models of multi-agent learning

Descriptive. Describe what happens in existing processes.

Typically social or economic processes.

Ideally, the formal model should ...

- ... reflect reality as much as possible.

Descriptive models of multi-agent learning

Descriptive. Describe what happens in existing processes.

Typically social or economic processes.

Ideally, the formal model should ...

- ... reflect reality as much as possible.
- ... have interesting theoretical properties:

Descriptive models of multi-agent learning

Descriptive. Describe what happens in existing processes.

Typically social or economic processes.

Ideally, the formal model should . . .

- . . . reflect reality as much as possible.
- . . . have interesting theoretical properties:
  - Strategies converge to a NE.

Descriptive models of multi-agent learning

Descriptive. Describe what happens in existing processes.

Typically social or economic processes.

Ideally, the formal model should ...

- ... reflect reality as much as possible.
- ... have interesting theoretical properties:
  - Strategies converge to a NE.
  - Empirical frequencies converge to a NE. (E.g. matching pennies.)

Descriptive models of multi-agent learning

Descriptive. Describe what happens in existing processes.

Typically social or economic processes.

Ideally, the formal model should ...

- ... reflect reality as much as possible.
- ... have interesting theoretical properties:
  - Strategies converge to a NE.
  - Empirical frequencies converge to a NE. (E.g. matching pennies.)
  - Strategies converge to a correlated NE. (Coordinating device = history of play).

Descriptive models of multi-agent learning

Descriptive. Describe what happens in existing processes.

Typically social or economic processes.

Ideally, the formal model should . . .

- . . . reflect reality as much as possible.
- . . . have interesting theoretical properties:
  - Strategies converge to a NE.
  - Empirical frequencies converge to a NE. (E.g. matching pennies.)
  - Strategies converge to a correlated NE. (Coordinating device = history of play).
  - Empirical frequencies end up in periodic or chaotic dynamics.

Prescriptive theories of multi-agent learning

Prescriptive. Prescribe how agents should learn.

Prescriptive theories of multi-agent learning

Prescriptive. Prescribe how agents should learn.

- More relevant to AI.

Prescriptive theories of multi-agent learning

Prescriptive. Prescribe how agents should learn.

- More relevant to AI.
- Prescriptive ~ normative.

Prescriptive theories of multi-agent learning

Prescriptive. Prescribe how agents should learn.

- More relevant to AI.
- Prescriptive ~ normative.

How to measure performance?

Prescriptive theories of multi-agent learning

Prescriptive. Prescribe how agents should learn.

- More relevant to AI.
- Prescriptive ~ normative.

How to measure performance?

- Through self-play.

Prescriptive theories of multi-agent learning

Prescriptive. Prescribe how agents should learn.

- More relevant to AI.
- Prescriptive ~ normative.

How to measure performance?

- Through self-play.
- Through tournaments.

Prescriptive theories of multi-agent learning

Prescriptive. Prescribe how agents should learn.

- More relevant to AI.
- Prescriptive ~ normative.

How to measure performance?

- Through self-play.
- Through tournaments.
    - How to set up a tournament?

Prescriptive theories of multi-agent learning

Prescriptive. Prescribe how agents should learn.

- More relevant to AI.
- Prescriptive ~ normative.

How to measure performance?

- Through self-play.
- Through tournaments.
  - How to set up a tournament?
  - Which other algorithms to incorporate in a tournament?

Prescriptive theories of multi-agent learning

Prescriptive. Prescribe how agents should learn.

- More relevant to AI.
- Prescriptive ~ normative.

How to measure performance?

- Through self-play.
- Through tournaments.
  - How to set up a tournament?
  - Which other algorithms to incorporate in a tournament?
- No learning algorithm performs optimal against all possible opponents.

Performance standards

Performance standards

1. **Auto-compatible**. Approximate Pareto-optimality in self-play.

Performance standards

1. **Auto-compatible**. Approximate Pareto-optimality in self-play.
2. **Safety**. At least earn the maxmin (security value).

Performance standards

1. **Auto-compatible**. Approximate Pareto-optimality in self-play.
2. **Safety**. At least earn the maxmin (security value).
3. **Targeted optimality**. Best response against a limited class of opponents.

Performance standards

1. **Auto-compatible**. Approximate Pareto-optimality in self-play.
2. **Safety**. At least earn the maxmin (security value).
3. **Targeted optimality**. Best response against a limited class of opponents.
4. Often weakened forms with additional demands can be attained.

Performance standards

1. **Auto-compatible**. Approximate Pareto-optimality in self-play.
2. **Safety**. At least earn the maxmin (security value).
3. **Targeted optimality**. Best response against a limited class of opponents.
4. Often weakened forms with additional demands can be attained.

   Efficient targeted learning. For every $\epsilon > 0$ and $0 < \delta < 1$, there exists an M polynomial in $1/\epsilon$ and $1/\delta$, such that after M steps, with probability $\geq 1 - \delta$, (1), (2) and (3) are achieved within ϵ.

Performance standards

1. **Auto-compatible**. Approximate Pareto-optimality in self-play.
2. **Safety**. At least earn the maxmin (security value).
3. **Targeted optimality**. Best response against a limited class of opponents.
4. Often weakened forms with additional demands can be attained.

   Efficient targeted learning. For every $\epsilon > 0$ and $0 < \delta < 1$, there exists an M polynomial in $1/\epsilon$ and $1/\delta$, such that after M steps, with probability $\geq 1 - \delta$, (1), (2) and (3) are achieved within ϵ.
5. **Rational**. Approximate a best response if the opponent(s) settle on stationary strategies.

Performance standards

1. **Auto-compatible**. Approximate Pareto-optimality in self-play.
2. **Safety**. At least earn the maxmin (security value).
3. **Targeted optimality**. Best response against a limited class of opponents.
4. Often weakened forms with additional demands can be attained.

   Efficient targeted learning. For every $\epsilon > 0$ and $0 < \delta < 1$, there exists an M polynomial in $1/\epsilon$ and $1/\delta$, such that after M steps, with probability $\geq 1 - \delta$, (1), (2) and (3) are achieved within ϵ.
5. **Rational**. Approximate a best response if the opponent(s) settle on stationary strategies.
6. **No regret**. At any point, earn no less than any pure strategy would have.

Some learning algorithms

Some learning algorithms

- **Fictitious play**. Follow the behaviour of your opponents, and play a best response at any round.

Some learning algorithms

- **Fictitious play**. Follow the behaviour of your opponents, and play a best response at any round.
- **Bully**. Stick to your plan and hope your opponents will follow.

Some learning algorithms

- **Fictitious play**. Follow the behaviour of your opponents, and play a best response at any round.
- **Bully**. Stick to your plan and hope your opponents will follow.
- **Reinforcement learning**. Play actions with optimal past payoffs.

Some learning algorithms

- **Fictitious play**. Follow the behaviour of your opponents, and play a best response at any round.
- **Bully**. Stick to your plan and hope your opponents will follow.
- **Reinforcement learning**. Play actions with optimal past payoffs. Vary sometimes to explore.

Some learning algorithms

- **Fictitious play**. Follow the behaviour of your opponents, and play a best response at any round.
- **Bully**. Stick to your plan and hope your opponents will follow.
- **Reinforcement learning**. Play actions with optimal past payoffs. Vary sometimes to explore. Actions may be conditionalised by counter-action profiles.

Some learning algorithms

- **Fictitious play**. Follow the behaviour of your opponents, and play a best response at any round.
- **Bully**. Stick to your plan and hope your opponents will follow.
- **Reinforcement learning**. Play actions with optimal past payoffs. Vary sometimes to explore. Actions may be conditionalised by counter-action profiles.
- **No-regret learning**. Play actions with optimal *hypothetical* past payoffs. (No need to explore!)

Some learning algorithms

- **Fictitious play**. Follow the behaviour of your opponents, and play a best response at any round.
- **Bully**. Stick to your plan and hope your opponents will follow.
- **Reinforcement learning**. Play actions with optimal past payoffs. Vary sometimes to explore. Actions may be conditionalised by counter-action profiles.
- **No-regret learning**. Play actions with optimal *hypothetical* past payoffs. (No need to explore!)
- **Bayesian learning**. Maintain a probability distribution on a set of opponent strategies. Update. (Problem: this set may miss out on the true opponent strategies.)

Some learning algorithms

- **Fictitious play**. Follow the behaviour of your opponents, and play a best response at any round.
- **Bully**. Stick to your plan and hope your opponents will follow.
- **Reinforcement learning**. Play actions with optimal past payoffs. Vary sometimes to explore. Actions may be conditionalised by counter-action profiles.
- **No-regret learning**. Play actions with optimal *hypothetical* past payoffs. (No need to explore!)
- **Bayesian learning**. Maintain a probability distribution on a set of opponent strategies. Update. (Problem: this set may miss out on the true opponent strategies.)
- **Evolutionary learning**. Many players. Each player follows one out of n possible strategies. Players interact 1-1 and randomly with other players many times. Those that perform well multiply fast.

Define the learning problem

What does a player know ex ante (i.e., before learning)?

Define the learning problem

What does a player know ex ante (i.e., before learning)?

- The structure of the game

Define the learning problem

What does a player know ex ante (i.e., before learning)?

- The structure of the game (actions, payoffs . . .)

Define the learning problem

What does a player know ex ante (i.e., before learning)?

- The structure of the game (actions, payoffs ...)
- The cognitive make-up of its opponents.

Define the learning problem

What does a player know ex ante (i.e., before learning)?

- The structure of the game (actions, payoffs . . .)
- The cognitive make-up of its opponents.

What is observable, what is there to learn?

Define the learning problem

What does a player know ex ante (i.e., before learning)?

- The structure of the game (actions, payoffs . . .)
- The cognitive make-up of its opponents.

What is observable, what is there to learn?

- Own actions.

Define the learning problem

What does a player know ex ante (i.e., before learning)?

- The structure of the game (actions, payoffs ...)
- The cognitive make-up of its opponents.

What is observable, what is there to learn?

- Own actions.
- Own payoffs.

Define the learning problem

What does a player know ex ante (i.e., before learning)?

- The structure of the game (actions, payoffs . . .)
- The cognitive make-up of its opponents.

What is observable, what is there to learn?

- Own actions.
- Own payoffs.
- Opponents' actions (i.e, counter action profiles).

Define the learning problem

What does a player know ex ante (i.e., before learning)?

- The structure of the game (actions, payoffs . . .)
- The cognitive make-up of its opponents.

What is observable, what is there to learn?

- Own actions.
- Own payoffs.
- Opponents' actions (i.e, counter action profiles).
- Opponent's payoffs (i.e, counter payoff profiles).

Literature

There is no book specifically dedicated to multi-agent learning.

Literature

There is no book specifically dedicated to multi-agent learning.

- Two important monographs on learning in games:

Literature

There is no book specifically dedicated to multi-agent learning.

- Two important monographs on learning in games:

  H. Peyton Young (2004): *Strategic Learning and it Limits*, Oxford UP.

Literature

There is no book specifically dedicated to multi-agent learning.

- Two important monographs on learning in games:

  H. Peyton Young (2004): *Strategic Learning and it Limits*, Oxford UP.

  D. Fudenberg and D.K. Levine (1998), *The Theory of Learning in Games*, MIT Press.

Literature

There is no book specifically dedicated to multi-agent learning.

- Two important monographs on learning in games:

  H. Peyton Young (2004): *Strategic Learning and it Limits*, Oxford UP.

  D. Fudenberg and D.K. Levine (1998), *The Theory of Learning in Games*, MIT Press.

- Some review articles (chapters) on multi-agent learning:

Literature

There is no book specifically dedicated to multi-agent learning.

- Two important monographs on learning in games:

  H. Peyton Young (2004): *Strategic Learning and it Limits*, Oxford UP.

  D. Fudenberg and D.K. Levine (1998), *The Theory of Learning in Games*, MIT Press.

- Some review articles (chapters) on multi-agent learning:

  Shoham *et al.* (2009): *Multi-agent Systems*. Ch. 7: "Learning and Teaching". Reviews several directions in multi-agent learning. Theorems and results are mentioned but typically not proven. Studying this chapter without also studying background literature per subject is actually pointless.

Literature

There is no book specifically dedicated to multi-agent learning.

- Two important monographs on learning in games:

  H. Peyton Young (2004): *Strategic Learning and it Limits*, Oxford UP.

  D. Fudenberg and D.K. Levine (1998), *The Theory of Learning in Games*, MIT Press.

- Some review articles (chapters) on multi-agent learning:

  Shoham *et al.* (2009): *Multi-agent Systems*. Ch. 7: "Learning and Teaching". Reviews several directions in multi-agent learning. Theorems and results are mentioned but typically not proven. Studying this chapter without also studying background literature per subject is actually pointless.

  Vidal (2010, unpublished): *Fundamentals of Multiagent Systems*. Ch. 5: "Learning in Multi-agent systems". Nicer but less ambitious than Shoham *et al.*'s Ch. 7.

Literature

There is no book specifically dedicated to multi-agent learning.

- Two important monographs on learning in games:

  H. Peyton Young (2004): *Strategic Learning and it Limits*, Oxford UP.

  D. Fudenberg and D.K. Levine (1998), *The Theory of Learning in Games*, MIT Press.

- Some review articles (chapters) on multi-agent learning:

  Shoham *et al.* (2009): *Multi-agent Systems*. Ch. 7: "Learning and Teaching". Reviews several directions in multi-agent learning. Theorems and results are mentioned but typically not proven. Studying this chapter without also studying background literature per subject is actually pointless.

  Vidal (2010, unpublished): *Fundamentals of Multiagent Systems*. Ch. 5: "Learning in Multi-agent systems". Nicer but less ambitious than Shoham *et al.*'s Ch. 7.

  "An Overview of Cooperative and Competitive Multiagent Learning," P.J. 't Hoen *et al.* (2006). In: *Proc. of the 1st Int. Workshop on Learning and Adaptation in MAS* (LAMAS 2005), LNAI **3898**, pp. 1-46, Springer.

MAL can invoke complex behaviour

MAL can invoke complex behaviour
Example: Cournot dynamics

Example: Cournot dynamics

Cournot dynamics is a case of MAL with real-valued actions.

Example: Cournot dynamics

Cournot dynamics is a case of MAL with real-valued actions.

- Firm 1 and 2 produce beer in continuous quantities of q_1 and q_2 hectoliters a day.

Example: Cournot dynamics

Cournot dynamics is a case of MAL with real-valued actions.

- Firm 1 and 2 produce beer in continuous quantities of q_1 and q_2 hectoliters a day.
- Let a be fixed.

Example: Cournot dynamics

Cournot dynamics is a case of MAL with real-valued actions.

- Firm 1 and 2 produce beer in continuous quantities of q_1 and q_2 hectoliters a day.
- Let a be fixed.
  - The sales price per unit linearly depends on the total output and is defined by $s = \max\{0, a - (q_1 + q_2)\}$.

Example: Cournot dynamics

Cournot dynamics is a case of MAL with real-valued actions.

- Firm 1 and 2 produce beer in continuous quantities of q_1 and q_2 hectoliters a day.
- Let a be fixed.
  - The sales price per unit linearly depends on the total output and is defined by $s = \max\{0, a - (q_1 + q_2)\}$.
  - The production costs per unit is defined as c, with $0 < c < a$.

Example: Cournot dynamics

Cournot dynamics is a case of MAL with real-valued actions.

- Firm 1 and 2 produce beer in continuous quantities of q_1 and q_2 hectoliters a day.
- Let a be fixed.
  - The sales price per unit linearly depends on the total output and is defined by $s = \max\{0, a - (q_1 + q_2)\}$.
  - The production costs per unit is defined as c, with $0 < c < a$.
  - The profit for Firm 1 per unit is therefore

$$\text{Profit}_1(q_1) = \begin{cases} q_1(a - (q_1 + q_2)) - q_1 \cdot c & \text{if } q_1 + q_2 \leq a, \\ -q_1 \cdot c & \text{else.} \end{cases}$$

Example: Cournot dynamics

Cournot dynamics is a case of MAL with real-valued actions.

- Firm 1 and 2 produce beer in continuous quantities of q_1 and q_2 hectoliters a day.
- Let a be fixed.
  - The sales price per unit linearly depends on the total output and is defined by $s = \max\{0, a - (q_1 + q_2)\}$.
  - The production costs per unit is defined as c, with $0 < c < a$.
  - The profit for Firm 1 per unit is therefore

$$\text{Profit}_1(q_1) = \begin{cases} q_1(a - (q_1 + q_2)) - q_1 \cdot c & \text{if } q_1 + q_2 \leq a, \\ -q_1 \cdot c & \text{else.} \end{cases}$$

- At the end of each week q_1 and q_2 are made public, and both firms adapt their production to the new situation

Example: Cournot dynamics

Cournot dynamics is a case of MAL with real-valued actions.

- Firm 1 and 2 produce beer in continuous quantities of q_1 and q_2 hectoliters a day.
- Let a be fixed.
  - The sales price per unit linearly depends on the total output and is defined by $s = \max\{0, a - (q_1 + q_2)\}$.
  - The production costs per unit is defined as c, with $0 < c < a$.
  - The profit for Firm 1 per unit is therefore

$$\text{Profit}_1(q_1) = \begin{cases} q_1(a - (q_1 + q_2)) - q_1 \cdot c & \text{if } q_1 + q_2 \leq a, \\ -q_1 \cdot c & \text{else.} \end{cases}$$

- At the end of each week q_1 and q_2 are made public, and both firms adapt their production to the new situation (= max. 2nd degree eq.).

Example: Cournot dynamics

Example: Cournot dynamics

Questions:

Example: Cournot dynamics

Questions: (1) how does adaptation proceed?

Example: Cournot dynamics

Questions: (1) how does adaptation proceed? (2) does every begin situation leads to the same outcome?

Example: Cournot dynamics

Questions: (1) how does adaptation proceed? (2) does every begin situation leads to the same outcome? (3) is there an equilibrium?

Example: Cournot dynamics

Questions: (1) how does adaptation proceed? (2) does every begin situation leads to the same outcome? (3) is there an equilibrium? (4) if so, how many?

Example: Cournot dynamics

Questions: (1) how does adaptation proceed? (2) does every begin situation leads to the same outcome? (3) is there an equilibrium? (4) if so, how many? (5) does adaptation lead to an equilibrium?

Example: Cournot dynamics

Questions: (1) how does adaptation proceed? (2) does every begin situation leads to the same outcome? (3) is there an equilibrium? (4) if so, how many? (5) does adaptation lead to an equilibrium?

- Dynamic approach. Start with random quantities. Adapt them at the end of every round.

Example: Cournot dynamics

Questions: (1) how does adaptation proceed? (2) does every begin situation leads to the same outcome? (3) is there an equilibrium? (4) if so, how many? (5) does adaptation lead to an equilibrium?

- Dynamic approach. Start with random quantities. Adapt them at the end of every week.

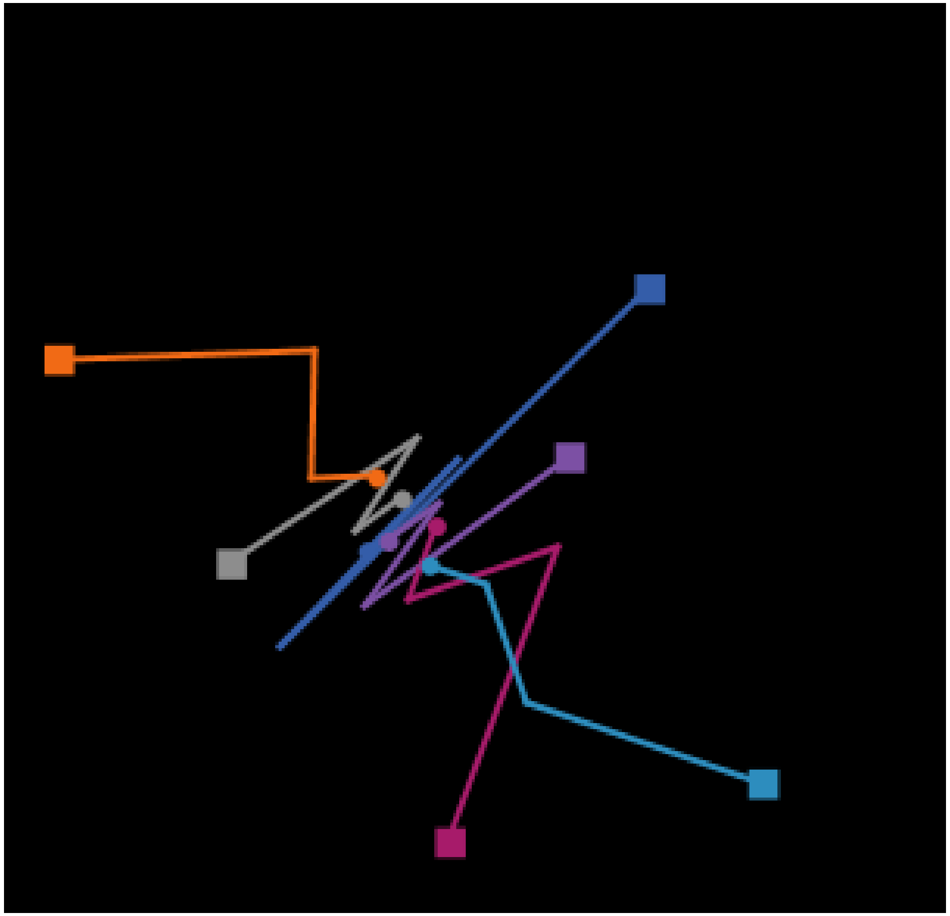


Fig: traces from 6 different starts.

Example: Cournot dynamics

Questions: (1) how does adaptation proceed? (2) does every begin situation leads to the same outcome? (3) is there an equilibrium? (4) if so, how many? (5) does adaptation lead to an equilibrium?

- Dynamic approach. Start with random quantities. Adapt them at the end of every week.
- Static approach. The pair (Q_1, Q_2) is called a Cournot equilibrium iff Q_1 is a best response to Q_2 and *vice versa*.

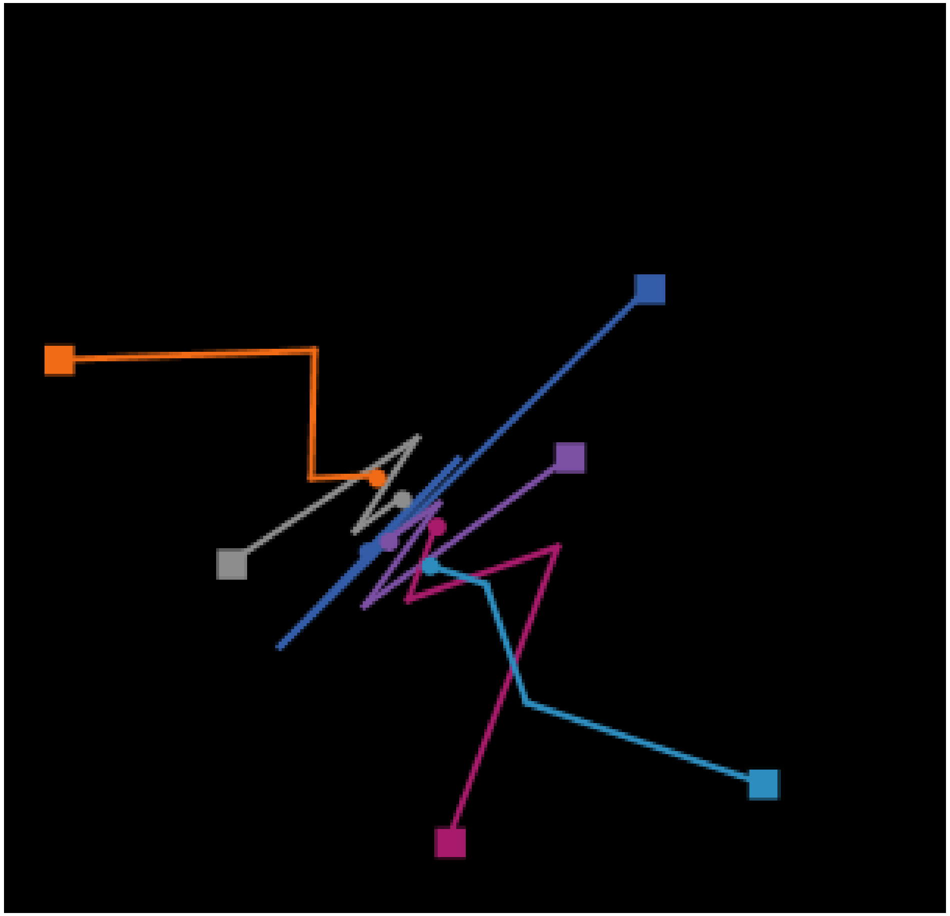


Fig: traces from 6 different starts.

Example: Cournot dynamics

Example: Cournot dynamics

- A Cournot Equilibrium can be computed by taking partial derivatives:

$$\frac{\partial}{\partial q_1}\mathrm{Profit}_1(q_1) = \begin{cases} \frac{\partial}{\partial q_1}[q_1(a-(q_1+q_2)) - q_1 \cdot c] & \text{if } q_1+q_2 \leq a, \\ \frac{\partial}{\partial q_1}[-q_1 \cdot c] & \text{else.} \end{cases}$$

Likewise for $\frac{\partial}{\partial q_2}\mathrm{Profit}_2(q_2)$.

Example: Cournot dynamics

- A Cournot Equilibrium can be computed by taking partial derivatives:

$$\frac{\partial}{\partial q_1}\text{Profit}_1(q_1) = \begin{cases} \frac{\partial}{\partial q_1}[q_1(a-(q_1+q_2)) - q_1 \cdot c] & \text{if } q_1+q_2 \leq a, \\ \frac{\partial}{\partial q_1}[-q_1 \cdot c] & \text{else.} \end{cases}$$

  Likewise for $\frac{\partial}{\partial q_2}\text{Profit}_2(q_2)$.

- Set partial derivatives to zero. We then have two equations with two unknowns. Solution:

$$\begin{aligned} q_1 &= \begin{cases} (a-c-q_2)/2 & \text{if } q_2 \leq a-c, \\ 0 & \text{else.} \end{cases} \\ q_2 &= \begin{cases} (a-c-q_1)/2 & \text{if } q_1 \leq a-c, \\ 0 & \text{else.} \end{cases} \end{aligned}$$

Example: Cournot dynamics

- A Cournot Equilibrium can be computed by taking partial derivatives:

$$\frac{\partial}{\partial q_1}\mathrm{Profit}_1(q_1) = \begin{cases} \frac{\partial}{\partial q_1}[q_1(a-(q_1+q_2)) - q_1\cdot c] & \text{if } q_1+q_2 \le a, \\ \frac{\partial}{\partial q_1}[-q_1\cdot c] & \text{else.} \end{cases}$$

  Likewise for $\frac{\partial}{\partial q_2}\mathrm{Profit}_2(q_2)$.

- Set partial derivatives to zero. We then have two equations with two unknowns. Solution:

$$q_1 = \begin{cases} (a-c-q_2)/2 & \text{if } q_2 \le a-c, \\ 0 & \text{else.} \end{cases}$$

$$q_2 = \begin{cases} (a-c-q_1)/2 & \text{if } q_1 \le a-c, \\ 0 & \text{else.} \end{cases}$$

- For linear prices and costs $\Rightarrow$ convergence to a unique equilibrium.

Example: Cournot dynamics

Now for more realistic price and cost functions:

Example: Cournot dynamics

Now for more realistic price and cost functions:

- Let sales price per unit be reciprocal: $s = 1/(q_1 + q_2)$.

Example: Cournot dynamics

Now for more realistic price and cost functions:

- Let sales price per unit be reciprocal: $s = 1/(q_1 + q_2)$.
- Let each party have different production costs per unit, say α and β.

Example: Cournot dynamics

Now for more realistic price and cost functions:

- Let sales price per unit be reciprocal: $s = 1/(q_1 + q_2)$.
- Let each party have different production costs per unit, say α and β.
- Let adaptation proceed gradually, through learning: $\text{new} = (1 - \delta) \cdot \text{old} + \delta \cdot \text{input}$.

Example: Cournot dynamics

Now for more realistic price and cost functions:

- Let sales price per unit be reciprocal: $s = 1/(q_1 + q_2)$.
- Let each party have different production costs per unit, say α and β.
- Let adaptation proceed gradually, through learning: $\text{new} = (1 - \delta) \cdot \text{old} + \delta \cdot \text{input}$.
- If $\alpha/\beta \in (3 - 2\sqrt{2}, 3 + 2\sqrt{2})$ then the equilibrium is stable.*

Example: Cournot dynamics

Now for more realistic price and cost functions:

- Let sales price per unit be reciprocal: $s = 1/(q_1 + q_2)$.
- Let each party have different production costs per unit, say α and β.
- Let adaptation proceed gradually, through learning: $\text{new} = (1 - \delta) \cdot \text{old} + \delta \cdot \text{input}$.
- If $\alpha/\beta \in (3 - 2\sqrt{2}, 3 + 2\sqrt{2})$ then the equilibrium is stable.*
- If $\alpha/\beta \in (4/25, 25/4)$ then trajectory remains bounded.*

Example: Cournot dynamics

Now for more realistic price and cost functions:

- Let sales price per unit be reciprocal: $s = 1/(q_1 + q_2)$.
- Let each party have different production costs per unit, say α and β.
- Let adaptation proceed gradually, through learning: $\text{new} = (1-\delta) \cdot \text{old} + \delta \cdot \text{input}$.
- If $\alpha/\beta \in (3 - 2\sqrt{2}, 3 + 2\sqrt{2})$ then the equilibrium is stable.*
- If $\alpha/\beta \in (4/25, 25/4)$ then trajectory remains bounded.*
- Whenever $0.16 < \alpha/\beta \leq 0.171\dots$ or $5.828\dots \leq \alpha/\beta \leq 6.25$ there is periodicity, semi-periodicity, or chaos.*

Fig: chaotic trajectory.

* Tönu Puu. Chaos in Duopoly Pricing. *Chaos, Solitions & Fractions* **1**(6), pp. 573-581, 1991.