A few topics in Reinforcement Learning (and related ideas)

E. Rachelson

Outline

1. Reinforcement Learning (and Machine Learning)
2. A personal timeline
3. One word on the Data and Decision Sciences cursus
4. Research overview
5. SuReLI
6. A result I really like
7. Conclusion

What is Reinforcement Learning?

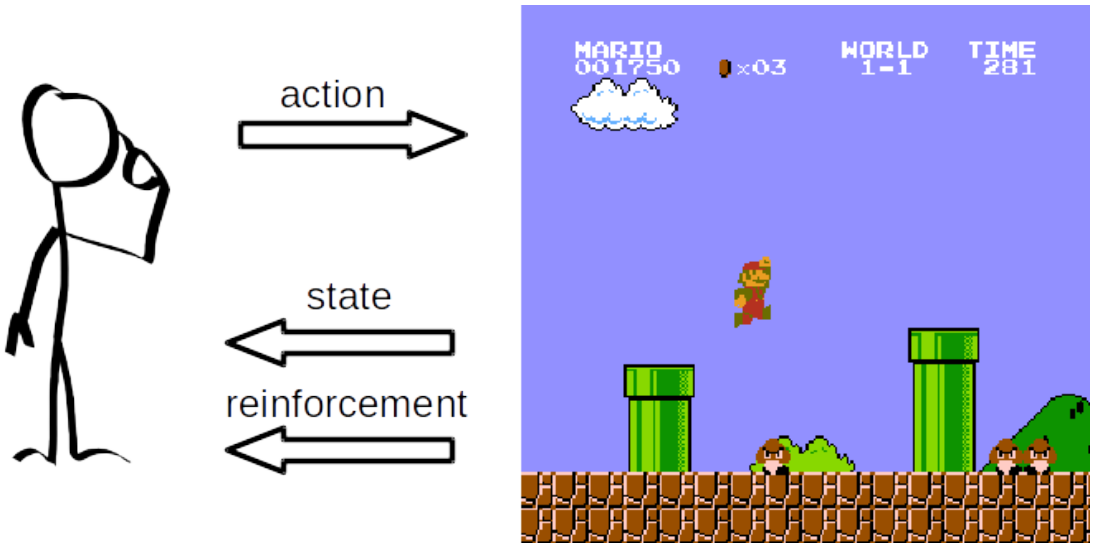
Learning to control a system through interaction.

- No model
- The *learning* version of Stochastic Optimal Control.
- The third branch of Machine Learning (with Sup. and Unsup. L.)
- Applies to industrial scheduling, robotics control, Go playing...

What is Reinforcement Learning?

Reinforcement Learning

Construct a close-loop control policy that maximizes a certain criterion, based on interaction data.



Why is RL hard?

You try to learn a function via incomplete information.

$$f(x) = \text{maximum expected} \sum_{t=0}^{\infty} r_t$$

+ noise ($\mathbb{E}$)
+ non-linearity (approximation issues, CV, stability)
+ many local optima (counter-intuitive results)
+ exploration / exploitation tradeoff for sampling

Machine Learning

Machines that learn?
Let's try to give a general definition.

Machine Learning

Machines that learn?
Let's try to give a general definition.

Machine learning is a field of computer science that gives computer systems the ability to "learn" (i.e. progressively improve performance on a specific task) with data, without being explicitly programmed.

ML tasks

What does ML do? 3 main tasks.

Task	Supervized Learning	Unsupervized Learning	Reinforcement Learning
Goal	Learn a function, $f(x) = y$	Find groups and correlations, $x \in C$	Optimal control, $f(x) = u \ / \ \max \sum r$
Data	$\{(x, y)\}$	$\{x\}$	$\{(x, u, r, x')\}$
Sub-task	Classification, Regression	Clustering, Density estimation, Dimensionnality reduction	Value estimation, Policy optimization
Algo ex.	Neural Networks, SVM, Random Forests	k-means, PCA, HCA	Q-learning

Misconceptions and clarifications

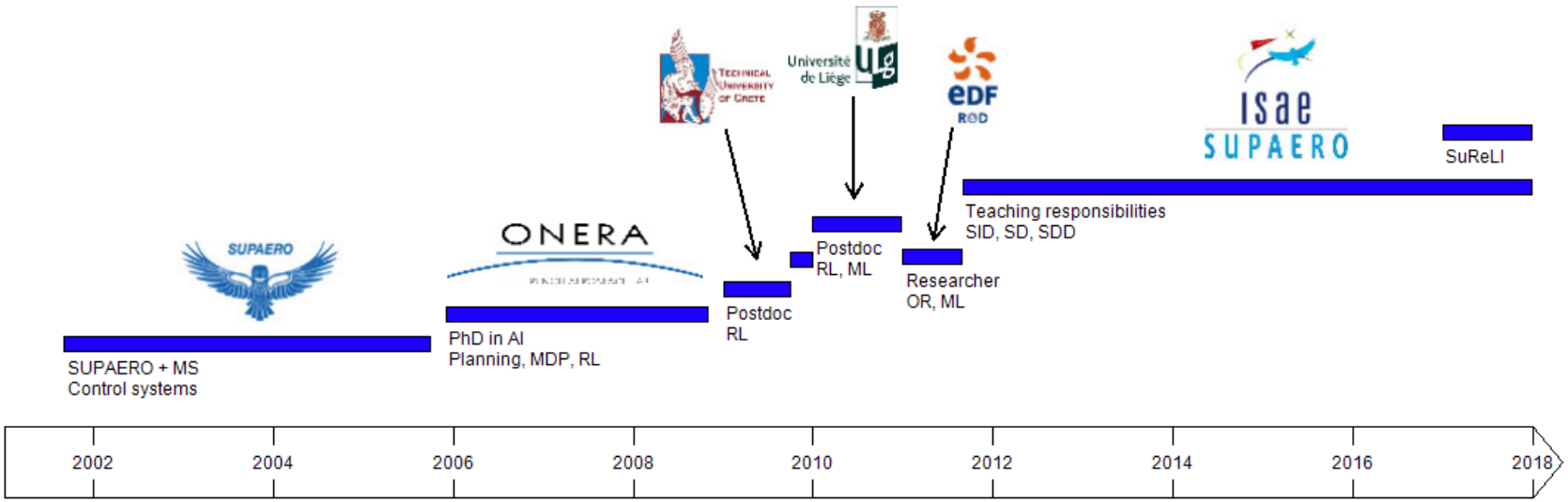
AI ML is only a small (currently fashionable) part of Artificial Intelligence.

BD Big Data refers to working with datasets that have large Volume, Variety, Velocity (, Veracity, and Value).

DL Deep Learning is Machine Learning with Deep Neural Networks.

threat ML / Data Science / Big Data are as much of a threat (to jobs, the society, the economy...) as the combustion engine was in the XIXth century.

A personal timeline



The *Data and Decision Sciences* cursus

Program head
Emmanuel Rachelson
e.rachelson@isae-supaero.fr

>50 trainers, experts from academia or major companies

2015-16: 19 students
2016-17: 30 students
2017-18: 53 students

Supaero curriculum

1st & 2nd year	Gap year	3rd year
Core eng. courses	Professional experience	Data and Decision Sciences program
Elective courses : - Markov chains - Applied optimization - Intro to Big Data ...		Internship

Hackathon

3 days challenge – real-life data
Industrial partners
2016 edition on image analysis with IRT – Saint Exupéry
Link to 2016 edition press release

Masters of Science

- Operations Research
- Applied Mathematics

Syllabus

Data Mining & Machine Learning
Advanced Statistics – Supervized, Unsupervized, Reinforcement Learning

Foundations in Decision Making
Decision Theory – Statistics – Optimization

Tools of Big Data
Databases – Programming – GPGPU
Distributed computing – cloud computing

Digital Economy and Data Uses
Business models – Privacy – Data storytelling

Internships

Dassault, Airbus, Sopra Steria, Air France, Bloomberg, Thales, Start-ups, international research institutes...

Overview of my research contributions (1/2)

Time-dependency in Markov Decision Processes

- Explicit time Bellman operator is a contraction mapping [ISAIM08]
- Explicit time-dependency models + Prioritized Sweeping algos [ICAPS09w]
- Implicit event models + simulation based algos [ECAI08]

Locality in MDPs

- Lipschitz cont. of MDP $\Rightarrow$ Lipschitz cont. of value functions
- LPI, an active sampling policy iteration algorithm [ISAIM10]

Tree-search planning

- Validity of open loop exection for optimistic planning [IJCAI18s]

Overview of my research contributions (2/2)

Learning for Optimisation

- Boolean value suggestion using classifiers [ICTAI10]
- NaiBX, a Naive Bayes multi-label classifier [ICPRAI18]
- Learning to generate subproblems / Branch and Bound control [OLA18]

Uncategorized

- Scaling up Gaussian Processes for Aeronautical design [LNCS17]
- Sample selection in Batch-mode RL [ICAART10]
- Heuristics in applied combinatorial optimization [ROADEF14]
- Attentional tunneling classification [THMS14]

SuReLI

ISAE-SUPAERO Reinforcement Learning Initiative. Since 2017.

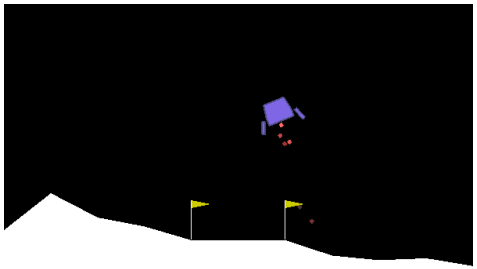
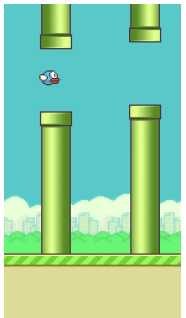
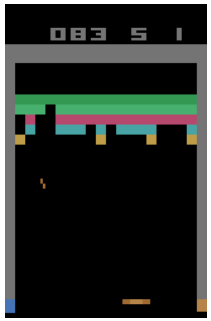
Github: `https://github.com/SuReLI`

Current connections: DCAS, ENAC, LAAS, ONERA, IRT, IRIT, ISIR, Poly Montreal, Jolibrain, Ubisoft, Airbus, Thalès Alenia Space...

SuReLI's current research

- Deep Reinforcement Learning

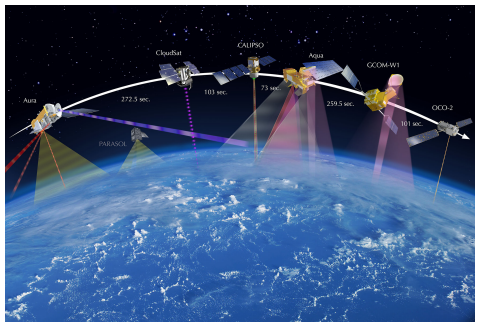
- Scaling up Policy Gradient methods for end-to-end control
  - Benchmarking SoTA algorithms
  - Pre-train / fine-tune / transfer
  - Expert demonstrations
- Deep TD(λ)

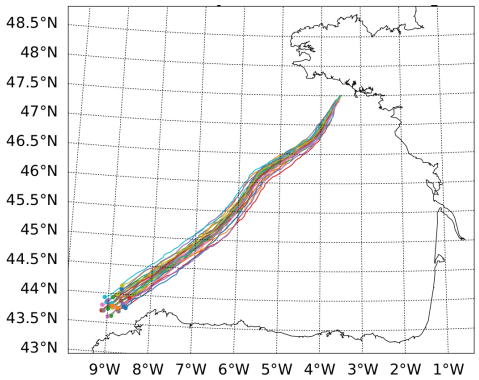





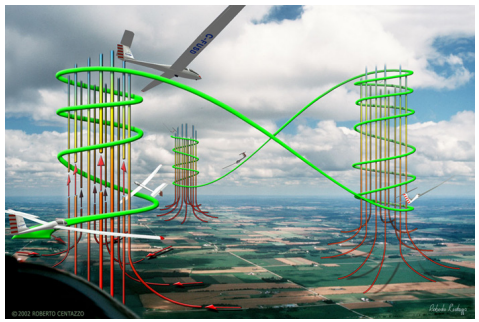
SuReLI's current research

- Deep Reinforcement Learning
- (Monte Carlo) Tree Search

- Validity of open-loop execution, plan. / replan. tradeoff
- State abstraction in tree search
- Distributed MCTS
- Action selection via Multi-agent systems



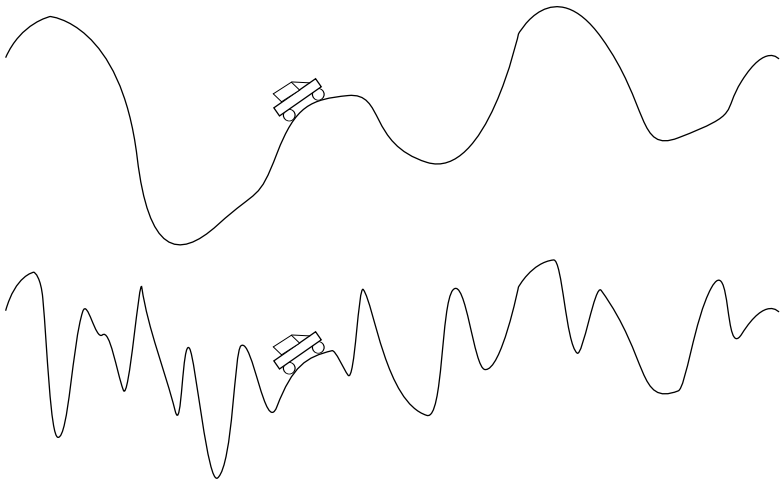




SuReLI's current research

- Deep Reinforcement Learning
- (Monte Carlo) Tree Search
- Theory of RL

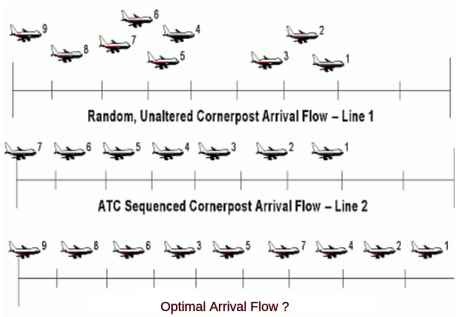
- Continuity analysis
- Non-stationary MDPs

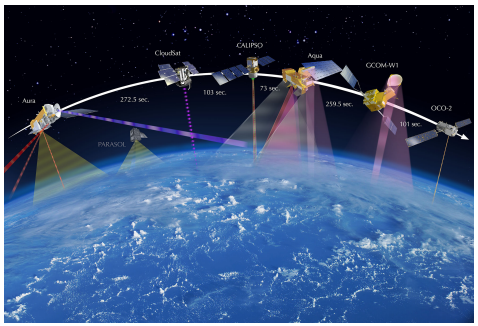


SuReLI's current research

- Deep Reinforcement Learning
- (Monte Carlo) Tree Search
- Theory of RL
- Learning for optimisation

- Sequential subproblem generation
- Parameter control
- Using learned value functions in MIP models

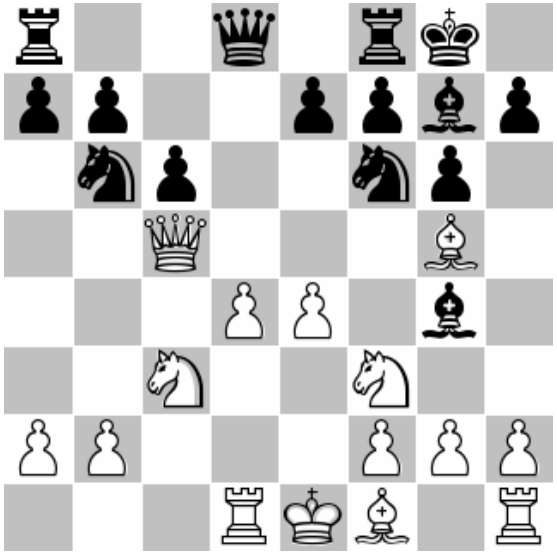




SuReLI's current research

- Deep Reinforcement Learning
  - Scaling up Policy Gradient methods for end-to-end control
  - Deep TD(λ)
  - Applications: iBoat, exoskeleton, Atari games
- (Monte Carlo) Tree Search
  - Validity of open-loop execution, plan. / replan. tradeoff
  - State abstraction in tree search
  - Distributed MCTS
  - Action selection via Multi-agent systems
  - Applications: UAV planning, iBoat, satellite imaging
- Theory of RL
  - Continuity analysis
  - Non-stationary MDPs
- Learning for optimisation
  - Sequential subproblem generation
  - Parameter control
  - Using learned value functions in MIP models
  - Applications: power generation, ATM, freq. allocation, TSP.

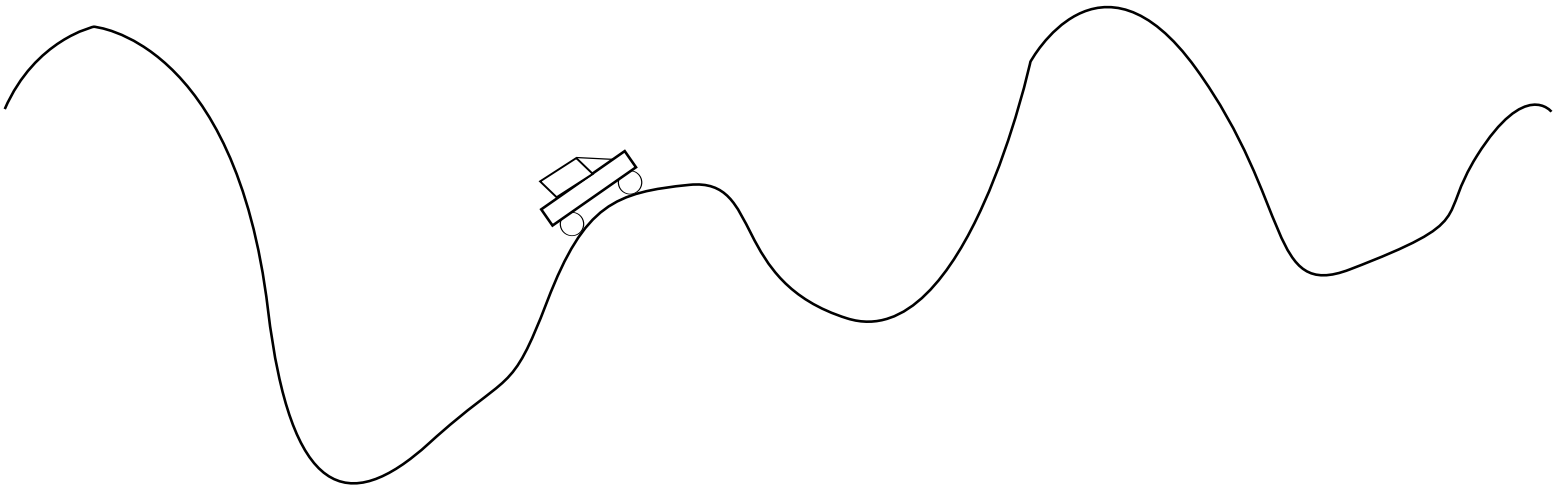
On the locality of action domination in seq. decision making



If I change slightly this chessboard, does the optimal action change?
How much can I change the chessboard?
Presentation of our 2010 ISAIM paper with almost no maths.

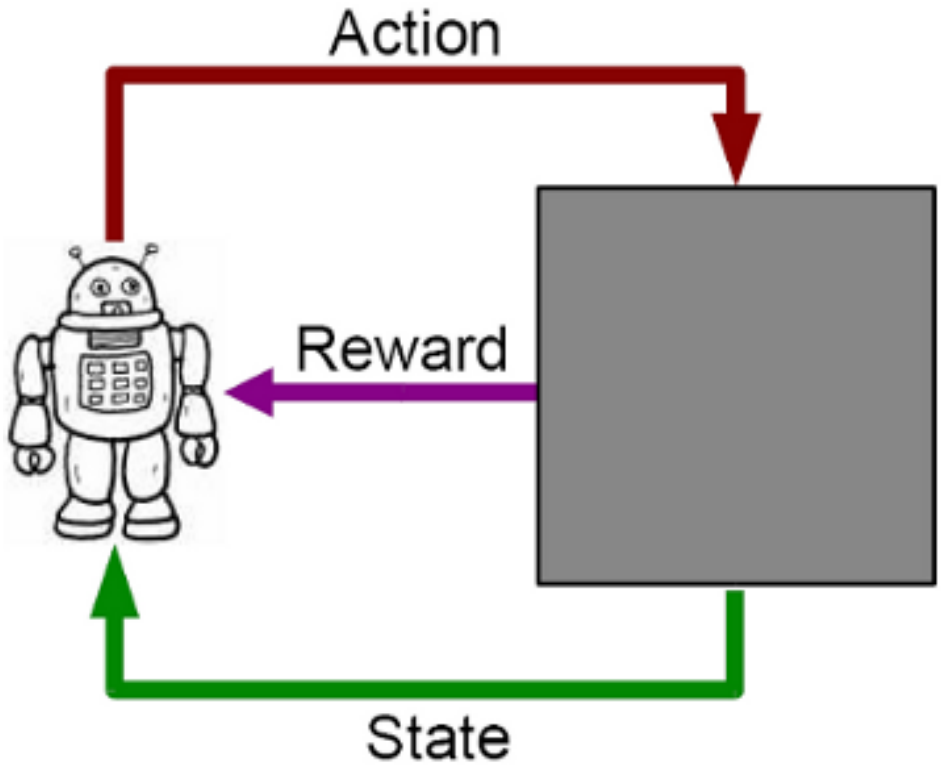
Background

Sequential decision making



Find the best sequence of L/R actions
or the best control policy
to reach the summit.

Background

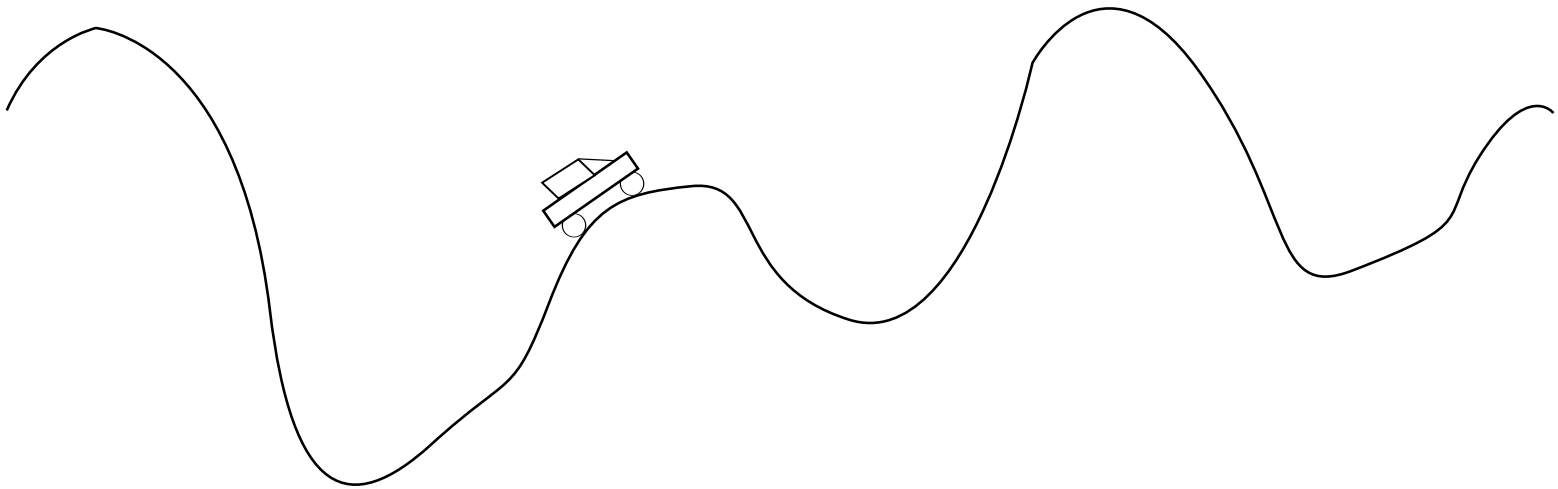


- States s
- Actions a
- Transitions $p(s'|s,a)$
- Instant rewards $r(s,a)$

Goal: optimize a cumulative reward $\sum_{t=0}^{\infty} \gamma^t r_t$.

How local is the knowledge gained from experience?

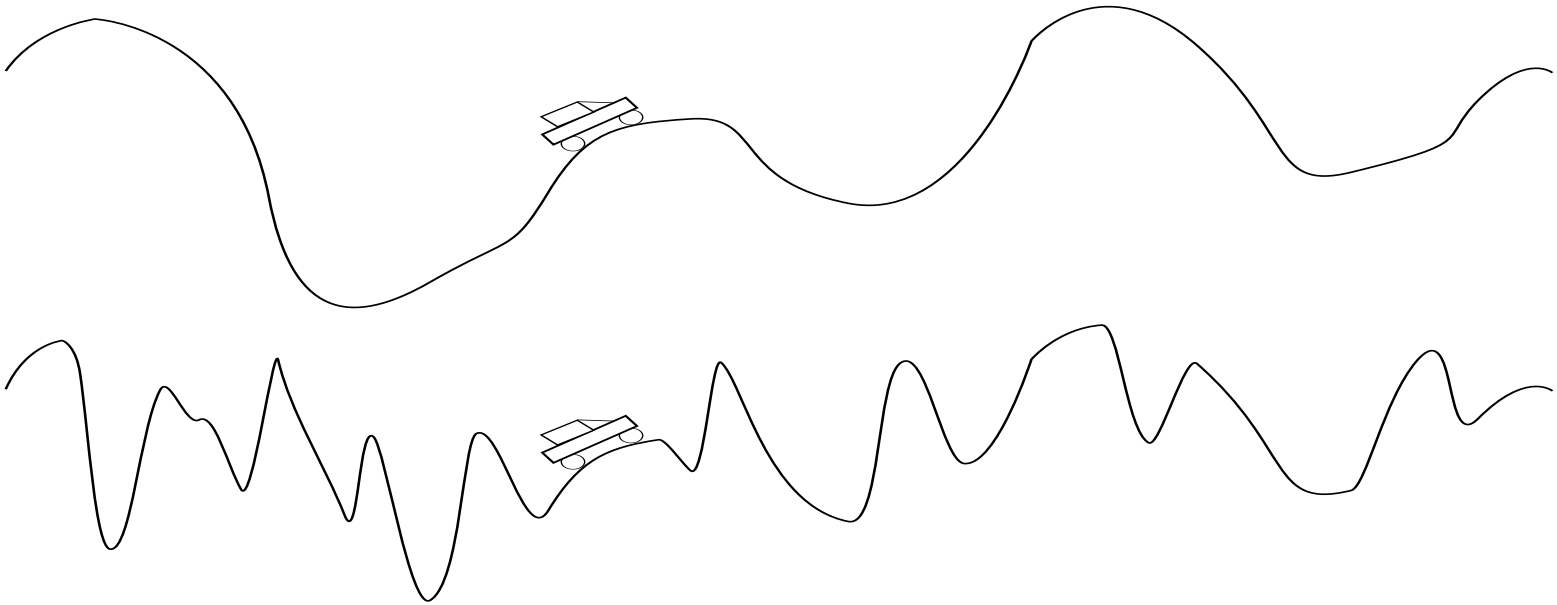
Learning an improved policy



Intuition indicates that a “good” action in a given position remains “good” *around* this position.

Environment smoothness

Ability to generalize $\leftrightarrow$ regularity of the environment



But the environment's model is unknown:

it is still possible to learn make an hypothesis on its *smoothness*.

Focus of this contribution

- Formalize the notion of smoothness for the environment,
- Derive properties for the optimal policy and its expected gain,
- Exploit these properties in an algorithm for RL problems.

Lipschitz-continuous environment

Model smoothness $\leftrightarrow$ Lipschitz continuity

Given a small move $(\hat{s}, \hat{a})$ from (s, a), an environment is Lipschitz continuous iff:

- The transition function does not change too much, ie.
  $p(s'|s, a)$ is close to $p(\hat{s}'|\hat{s}, \hat{a})$
  Note: supposes a metric on distributions.
- The reward model does not change too much, ie.
  $r(s, a)$ is close to $r(\hat{s}, \hat{a})$

Key theorem 1

Notation (value function): $Q^\pi(s,a) =$ expected gain of applying a in s and then following π.

Theorem (Lipschitz-continuity of the Q-function)

In a Lipschitz continuous environment, the value function of a piecewise constant policy is Lipschitz continuous if $\gamma L_p < 1$.

$\gamma L_p < 1$?

- γ is the discount rate on immediate rewards
- L_p is the maximum change rate of $p(s'|s,a)$ across states

$\Rightarrow$ The environment's spatial variations (L_p)
need to be compensated by
the discount on temporal variations (γ)
to obtain smoothness guarantees on the Q-function.

Key theorem 2

Suppose the action space is discrete.

Given a playing strategy π, let's optimize the first action we take.

Theorem (Influence radius of a sample)

If the best action a^* in s dominates all other actions by at least Δ, then it also dominates in all $s' \in B\big(s, \rho(s)\big)$

$$\rho(s) = \frac{\Delta}{2L_{Q^\pi}}$$

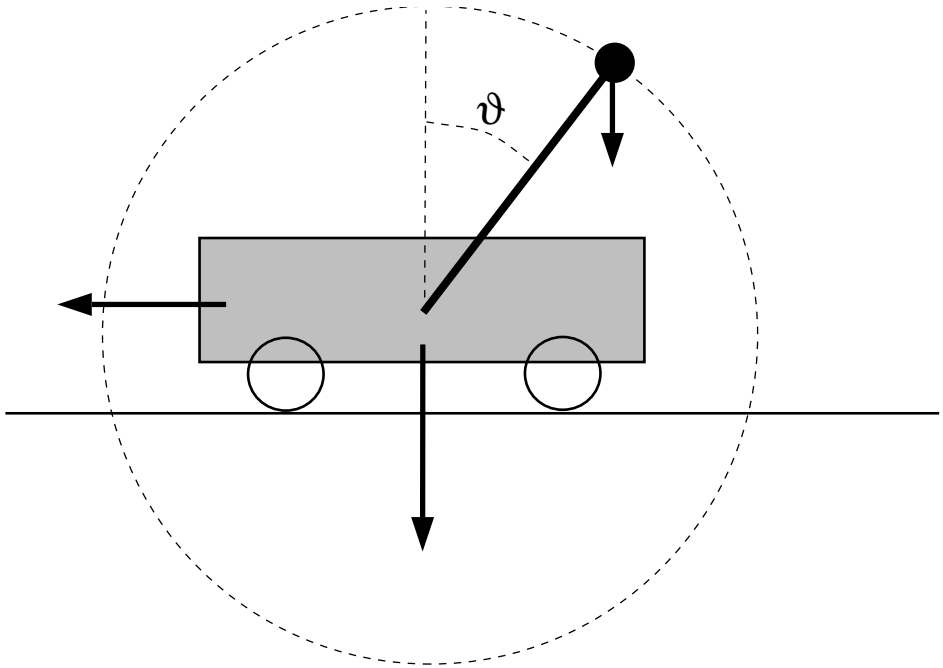
Localized Policy Iteration

Given π, and a simulator,
let's try to find the s with the biggest Δ.

- Action values differ a lot in $s \rightarrow$ important to identify the best one.
- Yields big $B\big(s, \rho(s)\big)$ balls $\rightarrow$ paves the set of states while minimizing the sampling.

$\rightarrow$ Define a bandit-based algorithm that identifies such states quickly (uses MC simulations to estimate Δ).
$\Rightarrow$ Active sampling algorithm.

LPI in action

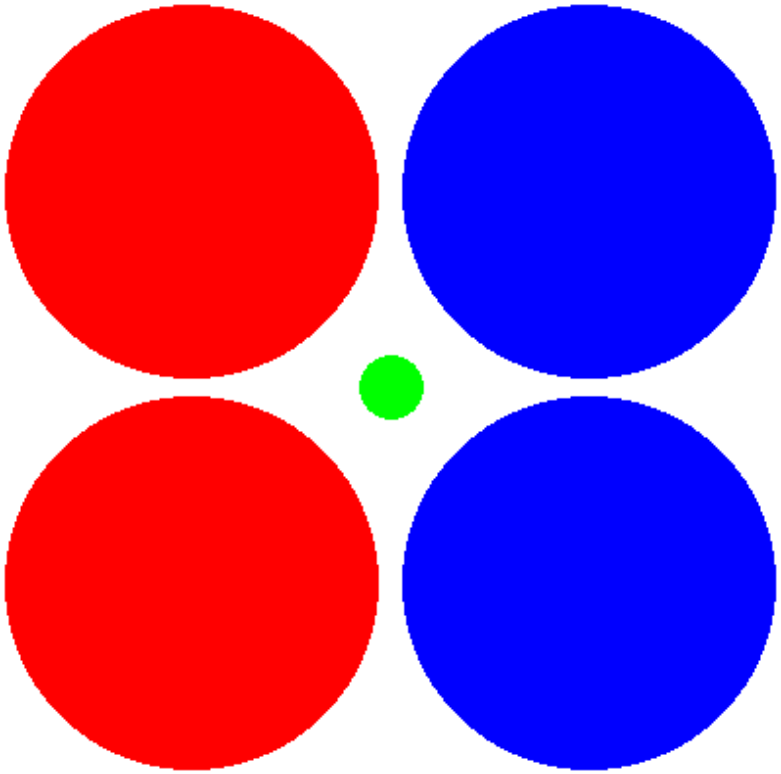


Goal: move left/right to balance the pendulum.

State space: $(\theta, \dot{\theta})$

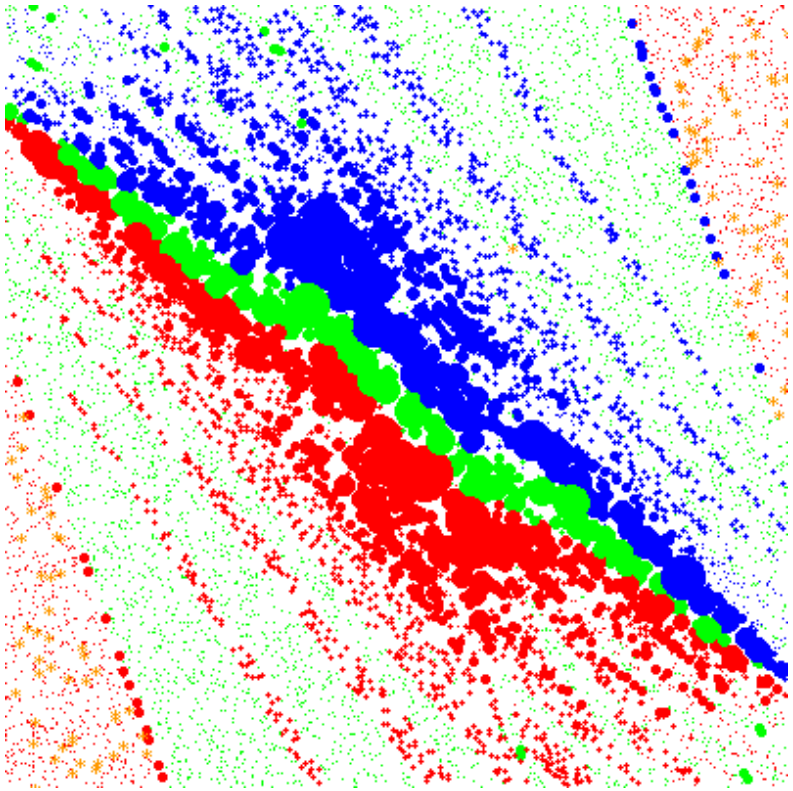
Game over if $|\theta| > \pi/2$

LPI in action



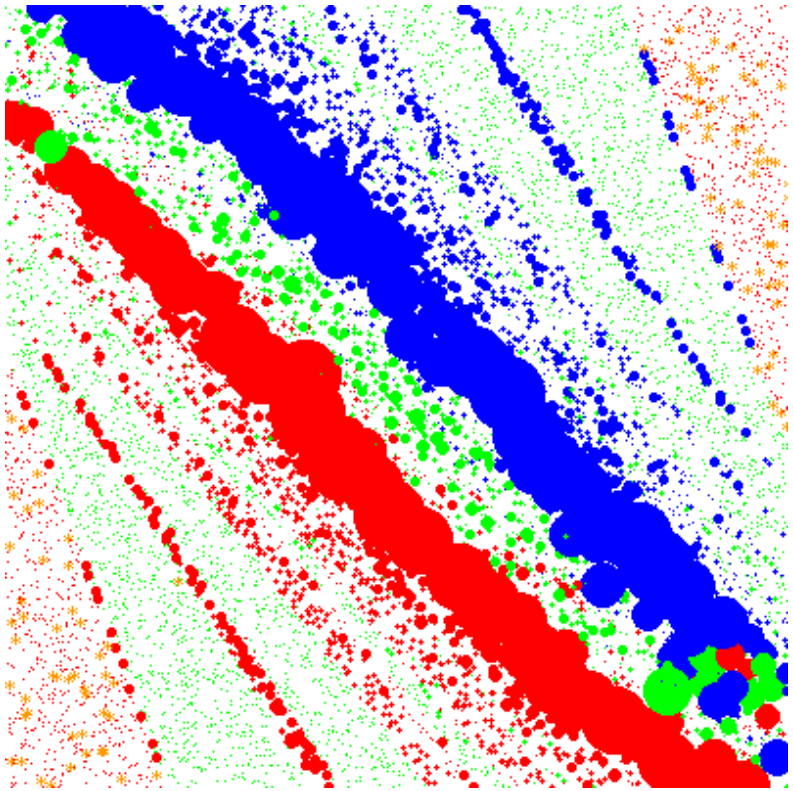
π_0

LPI in action



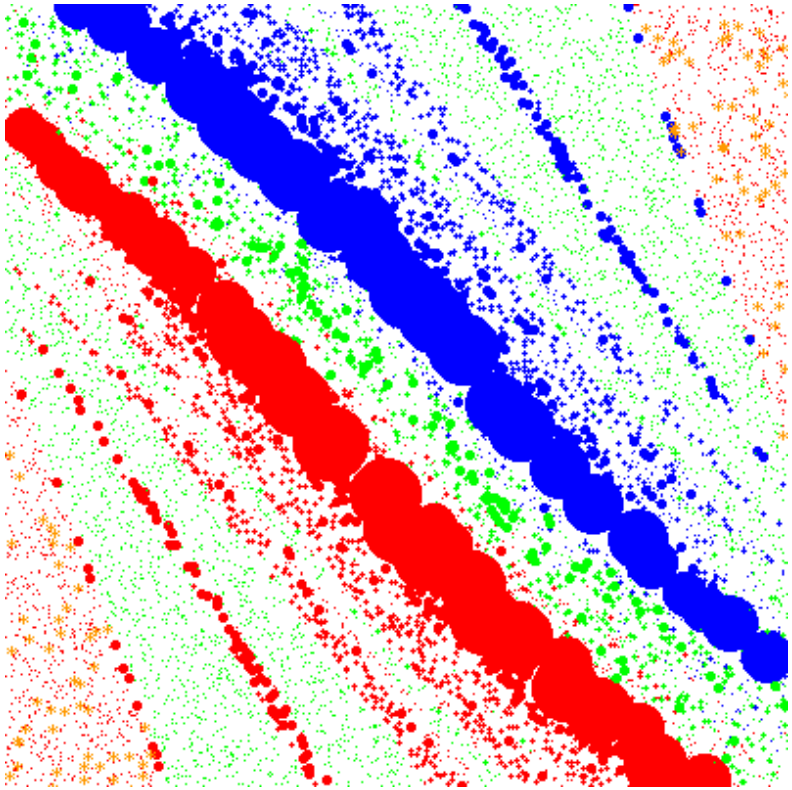
π_1

LPI in action



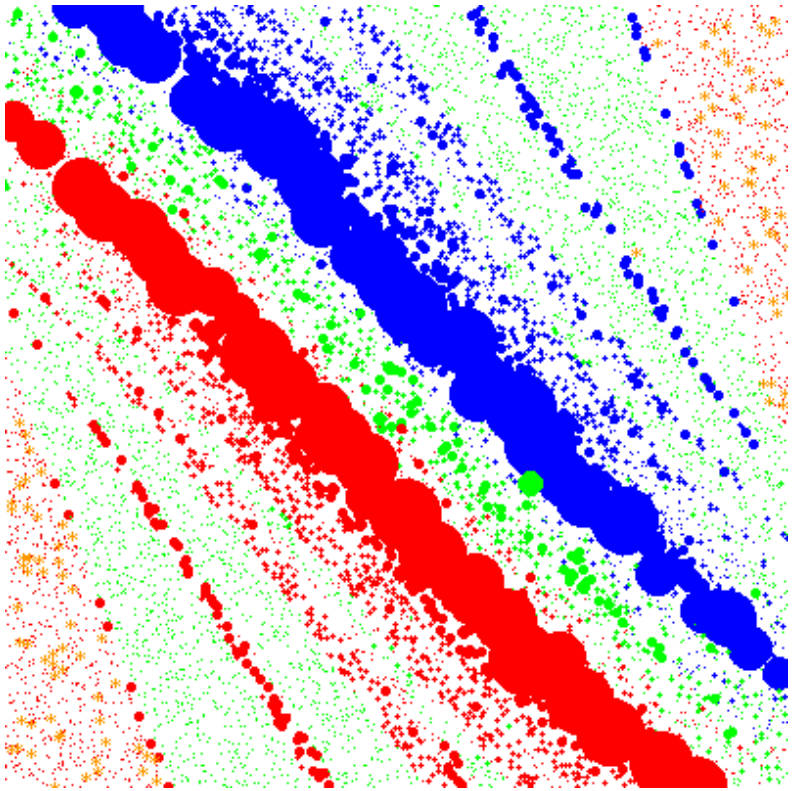
π_2

LPI in action



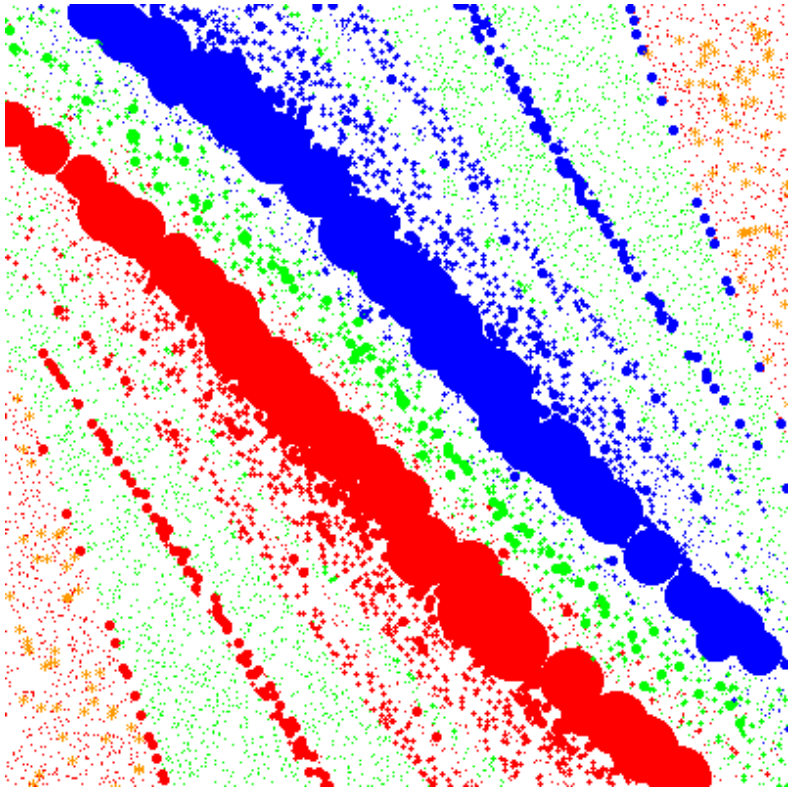
π_3

LPI in action



π_4

LPI in action



π_5

Conclusion

- Original question:
  How *local* is the info gathered in one state about the dominating action?

Conclusion

- Original question:
  How *local* is the info gathered in one state about the dominating action?
- Formalize the notion of *smoothness* for the environment's underlying model:
  Kantorovich distance, Lipschitz continuity $\rightarrow$ MDP smoothness.

Other metrics? Other continuity criterion?
Other similarity measure?

Conclusion

- Original question:
  How *local* is the info gathered in one state about the dominating action?
- Formalize the notion of *smoothness* for the environment's underlying model:
  Kantorovich distance, Lipschitz continuity $\rightarrow$ MDP smoothness.

  Other metrics? Other continuity criterion?
  Other similarity measure?
- Derive properties for the *policies* and *value functions*:
  LC of the (optimal) value functions, influence radius of a sample.

Conclusion

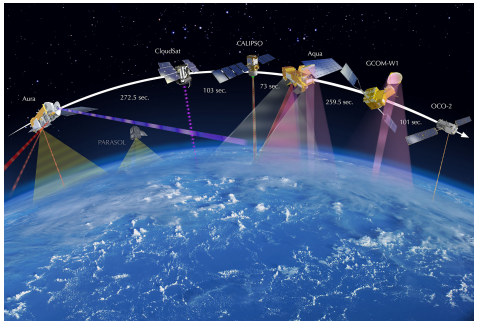
- Original question:
  How *local* is the info gathered in one state about the dominating action?
- Formalize the notion of *smoothness* for the environment's underlying model:
  Kantorovich distance, Lipschitz continuity $\rightarrow$ MDP smoothness.
  
  Other metrics? Other continuity criterion?
  Other similarity measure?
- Derive properties for the *policies* and *value functions*:
  LC of the (optimal) value functions, influence radius of a sample.
- Exploit these properties in an *algorithm* for RL problems:
  Localized Policy Iteration combines UCB-like methods with influence radii into an active learning method.
  
  Deeper study of incremental/asynchronous PI methods.

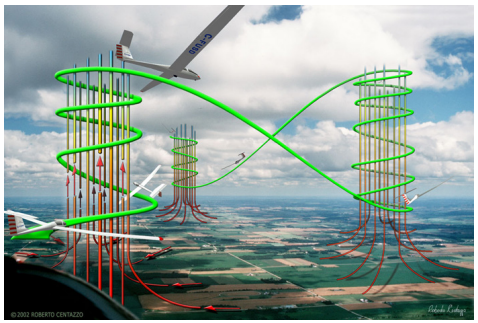
Why is it useful?

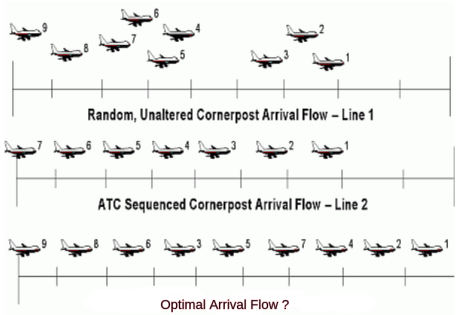
- Maybe it's not, but it's a formal explanation of an intuitive statement ($\gamma L_p < 1$)
- MCTS on continuous states: discretization
- Non-stationary problems: temporal validity, plan / replan tradeoff
- LWPR-like value function approximation.

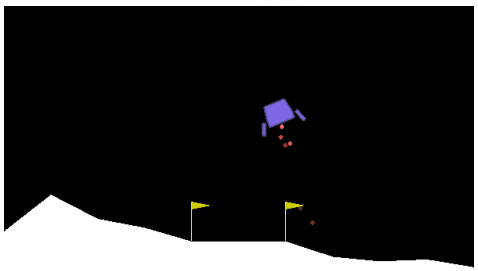
Back to reality

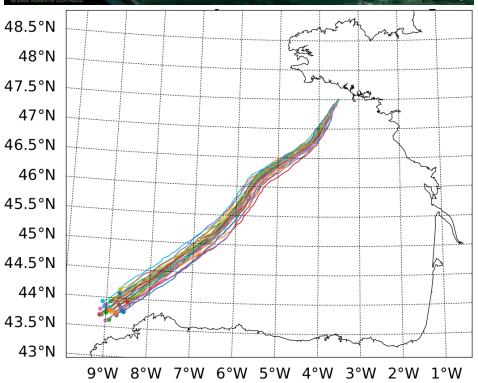
That was a lot of theory/philosophy. Back to research pictures.

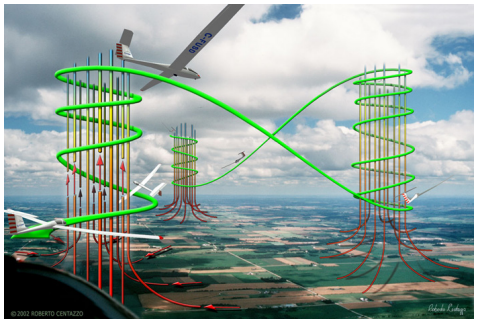


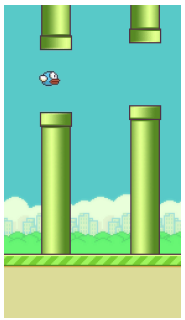


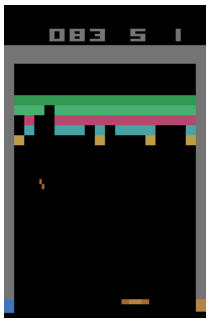












General conclusion

Now you should know a little more about:

- What I did in the last $\sim$ 10 years.
- Data and Decision Sciences @ ISAE-SUPAERO
- SuReLI
- RL research