# Learning Driving Styles for Autonomous Vehicles from Demonstration [1]

**Markus Kuderer, Shilpa Gulati, Wolfram Burgard**

## Reinforcement Learning

**Presented by:**
Damoun Ayman
Akash Malhotra

**Jury:**
Michèle Sebag
Herilalaina
Rakotoarison

**17/12/2020**

universite
PARIS-SACLAY

# Contents

1. Introduction and paper presentation
   - Problem and Proposed Solution

# Contents

1. Introduction and paper presentation
   - Problem and Proposed Solution

2. Reinforcement Learning Approach
   - Setting and Trajectory
   - Learning from Demonstration Methodology

université PARIS-SACLAY

# Contents


1. Introduction and paper presentation
   - Problem and Proposed Solution

2. Reinforcement Learning Approach
   - Setting and Trajectory
   - Learning from Demonstration Methodology

3. Experiments and results
   - Data Acquisition
   - Simulation results


université PARIS-SACLAY

# Contents

1. Introduction and paper presentation
   - Problem and Proposed Solution

2. Reinforcement Learning Approach
   - Setting and Trajectory
   - Learning from Demonstration Methodology

3. Experiments and results
   - Data Acquisition
   - Simulation results

4. Discussion

# Introduction and paper presentation

# Introduction


Figure 1: Pony.ai’s self-driving car in Guangzhou [2]

> Autonomous Vehicles or self-driving car contains thousands of sensors, and the way the vehicle is driven is strictly based on algorithms that analyse the inputs from sensors with little or no human intervention. [3]

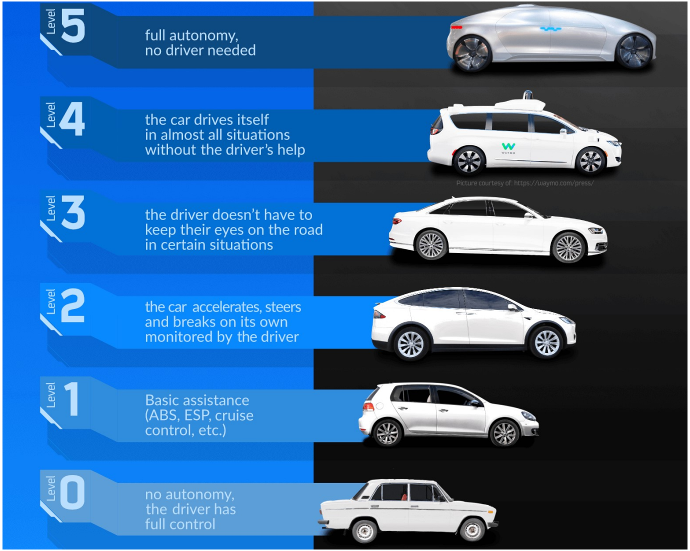



Figure 2: Levels of Vehicle Autonomy *SAE*

# Paper presentation

**Problem**

The authors affirm that in order to ensure comfort and acceptance by passengers, the self-driving car must use a driving style similar to that of the car's passengers.

# Paper presentation

**Problem**

The authors affirm that in order to ensure comfort and acceptance by passengers, the self-driving car must use a driving style similar to that of the car's passengers.

**Proposed Solution**

Learn the driving style of human drivers (Imitation Learning)

# Paper presentation



### Problem

The authors affirm that in order to ensure comfort and acceptance by passengers, the self-driving car must use a driving style similar to that of the car's passengers.

### Proposed Solution

Learn the driving style of human drivers (Imitation Learning)

### Methodology

Feature-based inverse reinforcement learning to create a continues reliable trajectory

# Reinforcement Learning Approach

# Forward Vs Inverse reinforcement learning

**"forward" reinforcement learning**
given:
states $\mathbf{s} \in \mathcal{S}$, actions $\mathbf{a} \in \mathcal{A}$
(sometimes) transitions $p(\mathbf{s}' \mid \mathbf{s}, \mathbf{a})$
reward function $r(\mathbf{s}, \mathbf{a})$
$\text{learn } \pi^{\star}(\mathbf{a} \mid \mathbf{s})$

# Forward Vs Inverse reinforcement learning

### "forward" reinforcement learning

given:
states $\mathbf{s} \in \mathcal{S}$, actions $\mathbf{a} \in \mathcal{A}$
(sometimes) transitions $p(\mathbf{s}' \mid \mathbf{s}, \mathbf{a})$
reward function $r(\mathbf{s}, \mathbf{a})$
$\text{learn } \pi^\star(\mathbf{a} \mid \mathbf{s})$

### Inverse reinforcement learnin

given:
states $\mathbf{s} \in \mathcal{S}$, actions $\mathbf{a} \in \mathcal{A}$
(sometimes) transitions $p(\mathbf{s}' \mid \mathbf{s}, \mathbf{a})$
samples $\{\tau_i\}$ sampled from $\pi^\star(\tau)$
$\text{learn } r_\theta(\mathbf{s}, \mathbf{a})$
Then use it to learn $\pi^\star(\mathbf{a} \mid \mathbf{s})$

# Forward Vs Inverse reinforcement learning

**"forward" reinforcement learning**

given:

states $\mathbf{s} \in \mathcal{S}$, actions $\mathbf{a} \in \mathcal{A}$

(sometimes) transitions $p\left(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}\right)$

reward function $r(\mathbf{s}, \mathbf{a})$

$\text{learn } \pi^{\star}(\mathbf{a} \mid \mathbf{s})$

**Inverse reinforcement learnin**

given:

states $\mathbf{s} \in \mathcal{S}$, actions $\mathbf{a} \in \mathcal{A}$

(sometimes) transitions $p\left(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}\right)$

samples $\{\tau_i\}$ sampled from $\pi^{\star}(\tau)$

$\text{learn } r_{\theta}(\mathbf{s}, \mathbf{a})$

Then use it to learn $\pi^{\star}(\mathbf{a} \mid \mathbf{s})$

$\theta$ reward parameters

linear reward function: $r_{\theta}(\mathbf{s}, \mathbf{a}) = \sum_i \theta_i f_i(\mathbf{s}, \mathbf{a}) = \theta^T \mathbf{f}(\mathbf{s}, \mathbf{a})$

# Trajectory

**Trajectory**
is a path that a vehicle should follow with a given velocity profile.

# Trajectory

**Trajectory**
is a path that a vehicle should follow with a given velocity profile.

**Trajectory Representation**
Use quintic (5th order) spline $s_j : [t_j, t_{j+1}] \to \mathbb{R}^2$.

$$\mathbf{r}(t) = \mathbf{s}_j(t) \quad \text{for} \quad t \quad \in \quad [t_j, t_{j+1}]$$

# The features of driving styles

The author defines driving style and the similarity between trajectories by:

- **Velocity Selections:** $f_v = \int_t \|\mathbf{v}_{\text{des}} - \dot{\mathbf{r}}(t)\| \, dt$

# The features of driving styles

The author defines driving style and the similarity between trajectories by:

- **Velocity Selections:** $f_v = \int_t \|\mathbf{v}_{\text{des}} - \dot{\mathbf{r}}(t)\| \, dt$
- **Acceleration Profile:** $f_a = \int_t \|\ddot{\mathbf{r}}(t)\|^2 dt$
- **Acceleration Lateral:** $f_{a_n} = \int_t \left(d_x(t)\ddot{\mathbf{r}}_y(t) - d_y(t)\ddot{\mathbf{r}}_x(t)\right)^2 dt$

# The features of driving styles

The author defines driving style and the similarity between trajectories by:

- **Velocity Selections:** $f_v = \int_t \|\mathbf{v}_{\text{des}} - \dot{\mathbf{r}}(t)\| \, dt$
- **Acceleration Profile:** $f_a = \int_t \|\ddot{\mathbf{r}}(t)\|^2 dt$
- **Acceleration Lateral:** $f_{a_n} = \int_t \left(d_x(t)\ddot{\mathbf{r}}_y(t) - d_y(t)\ddot{\mathbf{r}}_x(t)\right)^2 dt$
- **Jerk:** $f_{\mathrm{j}} = \int_t \|\dddot{\mathrm{r}}(t)\|^2 dt$
- **Normal Jerk** $f_{\mathrm{j}_n} = \int_t \left(d_x(t)\dddot{\mathbf{r}}_y(t) - d_y(t)\dddot{\mathbf{r}}_x(t)\right)^2 dt$

# The features of driving styles

The author defines driving style and the similarity between trajectories by:

- **Velocity Selections:** $f_v = \int_t \|\mathbf{v}_{\text{des}} - \dot{\mathbf{r}}(t)\| \, dt$
- **Acceleration Profile:** $f_a = \int_t \|\ddot{\mathbf{r}}(t)\|^2 dt$
- **Acceleration Lateral:** $f_{a_n} = \int_t \left(d_x(t)\ddot{\mathbf{r}}_y(t) - d_y(t)\ddot{\mathbf{r}}_x(t)\right)^2 dt$
- **Jerk:** $f_{\text{j}} = \int_t \|\dddot{\mathbf{r}}(t)\|^2 dt$
- **Normal Jerk** $f_{\text{j}_n} = \int_t \left(d_x(t)\dddot{\mathbf{r}}_y(t) - d_y(t)\dddot{\mathbf{r}}_x(t)\right)^2 dt$
- **Curvature of path** $f_\kappa = \int_t \|\kappa(t)\|^2 dt$
- **Lane keeping** $f_1 = \int_t \|\mathbf{l}(t) - \mathbf{r}(t)\| dt$
- **Collision avoidance with other vehicles:** $f_{\text{d}} = \sum_c \int_t \frac{1}{\|\mathbf{r}(t) - \mathbf{o}_c(t)\|^2} dt$
- **Following distance** $f_{\text{ft}} = \int \max(0, \hat{d} - d(t)) dt$

université PARIS-SACLAY

# The features of driving styles

The author defines driving style and the similarity between trajectories by:

- **Velocity Selections:** $f_v = \int_t \|\mathbf{v}_{\text{des}} - \dot{\mathbf{r}}(t)\| \, dt$
- **Acceleration Profile:** $f_a = \int_t \|\ddot{\mathbf{r}}(t)\|^2 dt$
- **Acceleration Lateral:** $f_{a_n} = \int_t \left(d_x(t)\ddot{\mathbf{r}}_y(t) - d_y(t)\ddot{\mathbf{r}}_x(t)\right)^2 dt$
- **Jerk:** $f_{\mathrm{j}} = \int_t \|\dddot{\mathbf{r}}(t)\|^2 dt$
- **Normal Jerk** $f_{\mathrm{j}_n} = \int_t \left(d_x(t)\dddot{\mathbf{r}}_y(t) - d_y(t)\dddot{\mathbf{r}}_x(t)\right)^2 dt$
- **Curvature of path** $f_\kappa = \int_t \|\kappa(t)\|^2 dt$
- **Lane keeping** $f_1 = \int_t \|\mathbf{l}(t) - \mathbf{r}(t)\| dt$
- **Collision avoidance with other vehicles:** $f_{\mathrm{d}} = \sum_c \int_t \frac{1}{\|\mathbf{r}(t) - \mathbf{o}_c(t)\|^2} dt$
- **Following distance** $f_{\mathrm{ft}} = \int \max(0, \hat{d} - d(t)) dt$

All features are then merged into a single feature vector.

# Maximum Entropy IRL

$$f : r \leftarrow (f1(r), ..., f_n(r)) \in \mathbb{R}_n \quad \text{- (1)}$$

$$\widetilde{f} = \frac{1}{N}\sum_{i=1}^{N} f(\widetilde{r_i}) \quad \text{- (2)}$$

$$\mathbb{E}_{p(r|\theta)}[f] = \widetilde{f} \quad \text{- (3)}$$

$$p(r|\theta) = exp(-\theta^T f(r)) \quad \text{- (4)}$$

$$\mathbb{E}_{p(r|\theta)}[f] - \widetilde{f} \quad \text{- (5)}$$

# Maximum Entropy IRL

$$f : r \leftarrow (f1(r), ..., f_n(r)) \in \mathbb{R}_n \quad \text{- (1)}$$

$$\widetilde{f} = \frac{1}{N} \sum_{i=1}^{N} f(\widetilde{r_i}) \quad \text{- (2)}$$

$$\mathbb{E}_{p(r|\theta)}[f] = \widetilde{f} \quad \text{- (3)}$$

$$p(r|\theta) = exp(-\theta^T f(r)) \quad \text{- (4)}$$

$$\mathbb{E}_{p(r|\theta)}[f] - \widetilde{f} \quad \text{- (5)}$$

## Maximum Likelihood estimation:

- Finding expectation is not possible using analytical methods, so we estimate the distribution and find it using MLE.

$$\mathbb{E}_p[f] \approx f(\arg\max_r p(r|\theta)) \quad \text{- (1)}$$

# Learning algorithm

1. We have the observed trajectories. Calculate emipirical feature vector $\widetilde{f}$.

# Learning algorithm

1. We have the observed trajectories. Calculate emipirical feature vector $\widetilde{f}$.
2. Initialize the weight vector $\theta$. Initialize start state (pos, vel, acc, etc.) of the predicted trajectory same as that of observed trajectory.

# Learning algorithm

1. We have the observed trajectories. Calculate emipirical feature vector $\widetilde{f}$.
2. Initialize the weight vector $\theta$. Initialize start state (pos, vel, acc, etc.) of the predicted trajectory same as that of observed trajectory.
3. Find subsequent spline control points such that minimizes $\theta_T f(r)$.

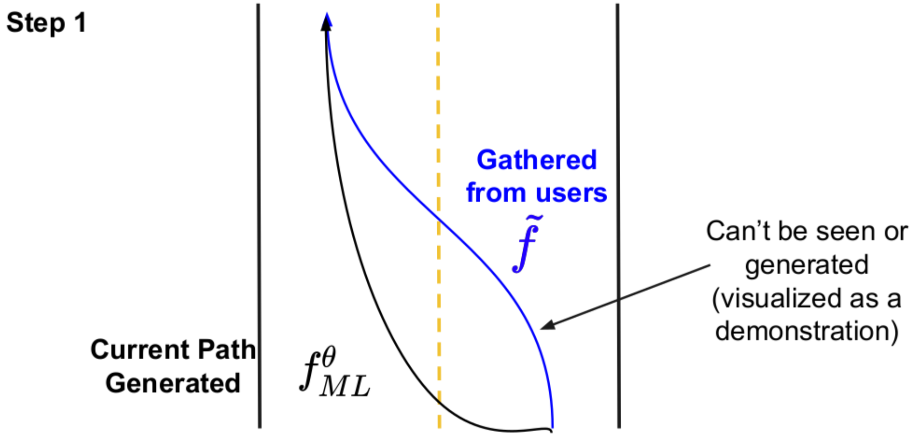
# Learning algorithm

1. We have the observed trajectories. Calculate emipirical feature vector $\widetilde{f}$.
2. Initialize the weight vector $\theta$. Initialize start state (pos, vel, acc, etc.) of the predicted trajectory same as that of observed trajectory.
3. Find subsequent spline control points such that minimizes $\theta_T f(r)$.
4. Calculate predicted feature vector and difference b/w predicted and observed feature vector $\mathbb{E}_{p(r|\theta)}[f] - \widetilde{f}$. Update the weights $\theta$ with this gradient.

# Learning algorithm

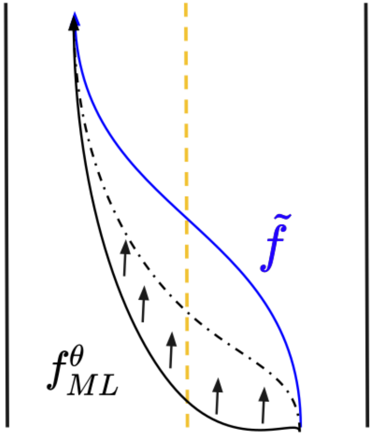
1. We have the observed trajectories. Calculate emipirical feature vector $\widetilde{f}$.
2. Initialize the weight vector $\theta$. Initialize start state (pos, vel, acc, etc.) of the predicted trajectory same as that of observed trajectory.
3. Find subsequent spline control points such that minimizes $\theta_T f(r)$.
4. Calculate predicted feature vector and difference b/w predicted and observed feature vector $\mathbb{E}_{p(r|\theta)}[f] - \widetilde{f}$. Update the weights $\theta$ with this gradient.
5. Repeat from step 3 until convergence.

# Learning from Demonstration example



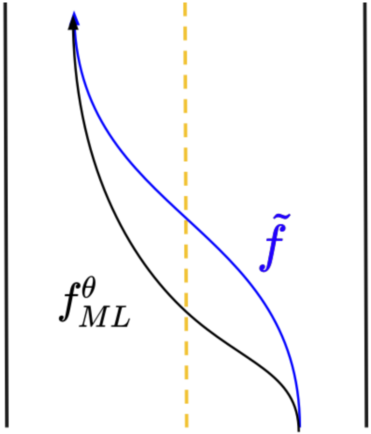
# Learning from Demonstration example

**Step 1**



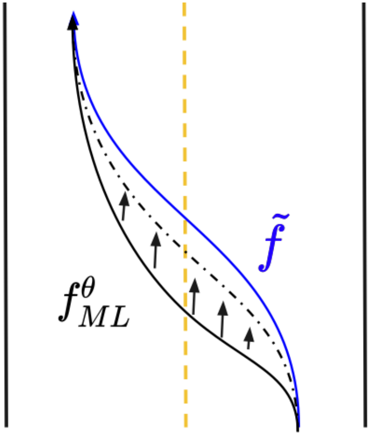
# Learning from Demonstration example

**Step 2**



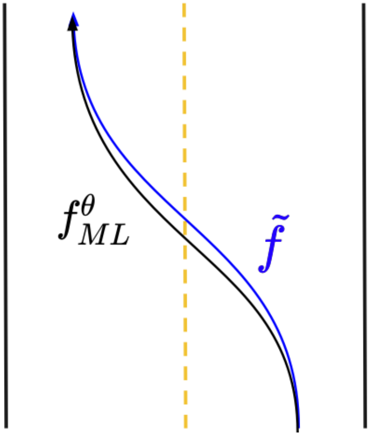
# Learning from Demonstration example

**Step 2**



# Learning from Demonstration example

**Step N**



# Experiments and results

# Navigation

1. Divide navigation into planning cycles.
2. Fix position, velocity and acceleration, $p_0, v_0, a_0$ at beginning of each planning cycle according to the trajectory of previous planning cycle.
3. Determine the next spline control, $p_j, v_j, a_j$ for $j \in 1, ..., S$ points w.r.t. to cost function $\theta^T f(r)$

# Experiments

1. **Data Acquisition:** Recorded Driving styles of different drivers on US highways.
   1.1 Acceleration maneuvers (23 m/s   30 m/s) and lane change maneuvers.
   1.2 Each segment of 8 mins long.

# Experiments

1. **Data Acquisition:** Recorded Driving styles of different drivers on US highways.
   1.1 Acceleration maneuvers (23 m/s  30 m/s) and lane change maneuvers.
   1.2 Each segment of 8 mins long.
2. **Learning:** individual styles. Learn $\theta$ that yield policies which explain observed trajectory.

# Experiments

1. **Data Acquisition:** Recorded Driving styles of different drivers on US highways.
   1.1 Acceleration maneuvers (23 m/s  30 m/s) and lane change maneuvers.
   1.2 Each segment of 8 mins long.
2. **Learning:** individual styles. Learn $\theta$ that yield policies which explain observed trajectory.
3. **Tested** in realistic simulation.

universite PARIS-SACLAY

# Results

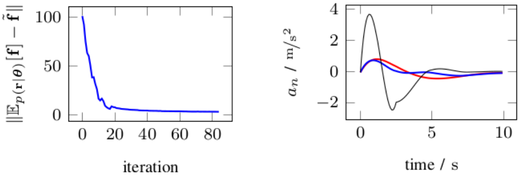



Figure 3: [1]

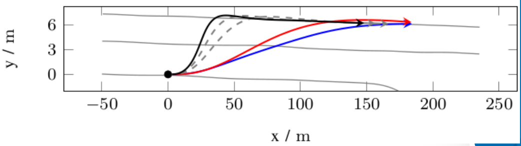



Figure 4: [1]

(Left) Norm of Expected and observed feature values. (Right) Normal acceleration during lane change. Initial Guess (black), Observed (blue), and Learned (red).

Trajectory during lane change. Initial Guess (black), Observed (blue), and Learned (red).

# Results

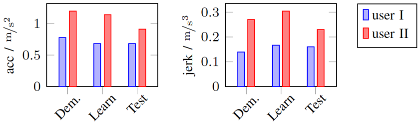



Figure 5: [1]

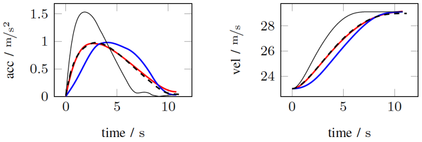



Figure 6: [1]

Comparison of mean observed and learned values for acc and jerk of driving style of two users.

Observed (blue), initial guess (black), learned (blue) and dotted (in realistic simulation) values for acceleration and velocity.

# Discussion and Related works

# Discussion

**Pros:**

- A novel approach. (Continuous domain, non MDP)
- Explainable features
- Simple to implement

**Cons:**

- Not enough info about experimental setup.
- No numerical quantification of "goodness". No benchmark is given.
- Tested only in simulation -> reality gap
- Risky: If a person drives badly, the algorithm will learn to drive badly
- 5 Hz is a concerningly slow planner to deal with all situations and speeds

# Related Works

- IRL for driving styles -> Abbeel and Ng[4], Ziebart et al.[5], Choi and Kim[6]
- New in this paper: Use IRL in continuous state spaces (can consider higher order properties such as latent jerk) and do not model dynamics as MDP in context of highway driving.
- Inspiration: learning pedestrian behavior. [7] [8]
- Inspired:
    - Cited by 257
    - [9] takes into account the response of other cars
    - [10] uses GANs for imitation learning from Human Driver (comparison with rule based and MLE).
    - [11] proposes algorithm for learning from humans in dynamical systems.

# References I


M. Kuderer, S. Gulati, and W. Burgard.
Learning driving styles for autonomous vehicles from demonstration.
In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2641–2646, 2015.

Pony.ai.
Guangzhou, china is a great place to test self driving cars.
`https://youtu.be/RDEMj1MGKjQ`.

Araz Taeihagh and Hazel Si Min Lim.
Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks.
*Transport Reviews*, 39(1):103–128, 2019.

# References II


Pieter Abbeel and Andrew Y Ng.
Apprenticeship learning via inverse reinforcement learning.
In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

Brian D Ziebart, Andrew L Maas, Anind K Dey, and J Andrew Bagnell.
Navigate like a cabbie: Probabilistic reasoning from observed context-aware behavior.
In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 322–331, 2008.

# References III


Jaedeug Choi and Kee-Eung Kim.
Bayesian nonparametric feature construction for inverse reinforcement learning.
In *Twenty-Third International Joint Conference on Artificial Intelligence*. Citeseer, 2013.

Markus Kuderer, Henrik Kretzschmar, Christoph Sprunk, and Wolfram Burgard.
Feature-based prediction of trajectories for socially compliant navigation.
In *Robotics: science and systems*, 2012.

Henrik Kretzschmar, Markus Kuderer, and Wolfram Burgard.
Learning to predict trajectories of cooperatively navigating agents.
In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 4015–4020. IEEE, 2014.

# References IV


Wilko Schwarting, Javier Alonso-Mora, and Daniela Rus.
Planning and decision-making for autonomous vehicles.
*Annual Review of Control, Robotics, and Autonomous Systems*, 2018.

Alex Kuefler, Jeremy Morton, Tim Wheeler, and Mykel Kochenderfer.
Imitating driver behavior with generative adversarial networks.
In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 204–211. IEEE, 2017.

Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia.
Active preference-based learning of reward functions.
In *Robotics: Science and Systems*, 2017.


universite PARIS-SACLAY

# Thank you for your attention
# questions ?

universite
PARIS-SACLAY

# Learning Driving Styles for Autonomous Vehicles from Demonstration [1]

**Markus Kuderer, Shilpa Gulati, Wolfram Burgard**

## Reinforcement Learning

**Presented by:**
Damoun Ayman
Akash Malhotra

**Jury:**
Michèle Sebag
Herilalaina
Rakotoarison

**17/12/2020**

universite
PARIS-SACLAY