Learning Driving Styles for Autonomous Vehicles for Demonstration

Markus Kuderer, Shilpa Gulati, Wolfram Burgard

Presented by: Marko Ilievski

Agenda

1. Problem definition
2. Background
    a. Important vocabulary
    b. Driving style
3. Reinforcement Learning Approach
4. Results
5. Issues with the approach
6. Discussion

Introduction

Problem

The authors claim that to **ensure comfort and acceptance** by passengers self-driving car must use **similar driving styles to** that of the **passengers in the car.**

Fig. 1. A Bosch highly automated driving development vehicle.

Proposed Solution

Learn the driving style of human drivers

Methodology

Feature-based inverse reinforcement learning to create at **continues reliable trajectory**

Background (Definitions)

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

- **Feasible -** Given the environment is the trajectory possible to execute
  - Car dynamics
  - Road dynamics and conditions
- **Ensure Safety -** there is no collision with any static or dynamic object
- **Passenger Comfort** (ex. Hard braking with fast accelerations)
- **Obey Road Regulation** (ex. Running a red light, no signaling between lane changes)

Driving Style - a method of selecting similar trajectories given a driver preferences

Background (Driving Style)

What comprises driving style, and how do you find similarities between trajectories (as defined by the authors)?

- Velocity Selections
- Acceleration Profile
- Jerk
- Curvature of path
- Lane keeping
- Collision avoidance with other vehicles
- Following distance

Calculating Similarities between trajectories

- Velocity Selections $f_{\mathrm{v}} = \int_t \|\mathbf{v}_{\mathrm{des}} - \dot{\mathbf{r}}(t)\| \; dt,$
- Acceleration Profile $f_{\mathrm{a}} = \int_t \|\ddot{\mathbf{r}}(t)\|^2 \; dt.$ Lateral $f_{\mathrm{a}_n} = \int_t (d_x(t)\ddot{\mathbf{r}}_y(t) - d_y(t)\ddot{\mathbf{r}}_x(t))^2 \; dt,$
- Jerk $f_{\mathrm{j}} = \int_t \|\dddot{\mathbf{r}}(t)\|^2 \; dt.$ Lateral $f_{\mathrm{j}_n} = \int_t (d_x(t)\dddot{\mathbf{r}}_y(t) - d_y(t)\dddot{\mathbf{r}}_x(t))^2 \; dt,$
- Curvature of path $f_{\kappa} = \int_t \|\kappa(t)\|^2 \; dt.$
- Lane keeping $f_{\mathrm{l}} = \int_t \|\mathbf{l}(t) - \mathbf{r}(t)\| \; dt.$
- 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 $f_{\mathrm{fd}} = \int_t \max\left(0, \hat{d} - d(t)\right) \; dt,$

All features are then merged into a single feature vector.

Learning from Demonstration (Algorithm)

- **Trajectories** are quantized as a set of 2D **quintic polynomials**.
- Maximum Entropy Inverse Reinforcement Learning Loop
  - Given observed trajectories $\{\tilde{\mathbf{r}}_1, \ldots, \tilde{\mathbf{r}}_N\}$
  - Calculate an average feature vector using the set of features defined above of all observed trajectories $\tilde{\mathbf{f}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{f}(\tilde{\mathbf{r}}_i)$
  - Try to find a set of parameters θ such that $\mathbb{E}_{p(\mathbf{r}|\boldsymbol{\theta})}[\mathbf{f}] \approx \mathbf{f}_{\mathrm{ML}}^{\boldsymbol{\theta}} := \frac{1}{N}\sum_{i=1}^{N} \mathbf{f}(\mathbf{r}_i^{\boldsymbol{\theta}})$. representing the difference between the current trajectory and the goal trajectory
  - Update the parameters of θ such that the gradient of $\mathbf{f}_{\mathrm{ML}}^{\boldsymbol{\theta}} - \tilde{\mathbf{f}}$, is optimized

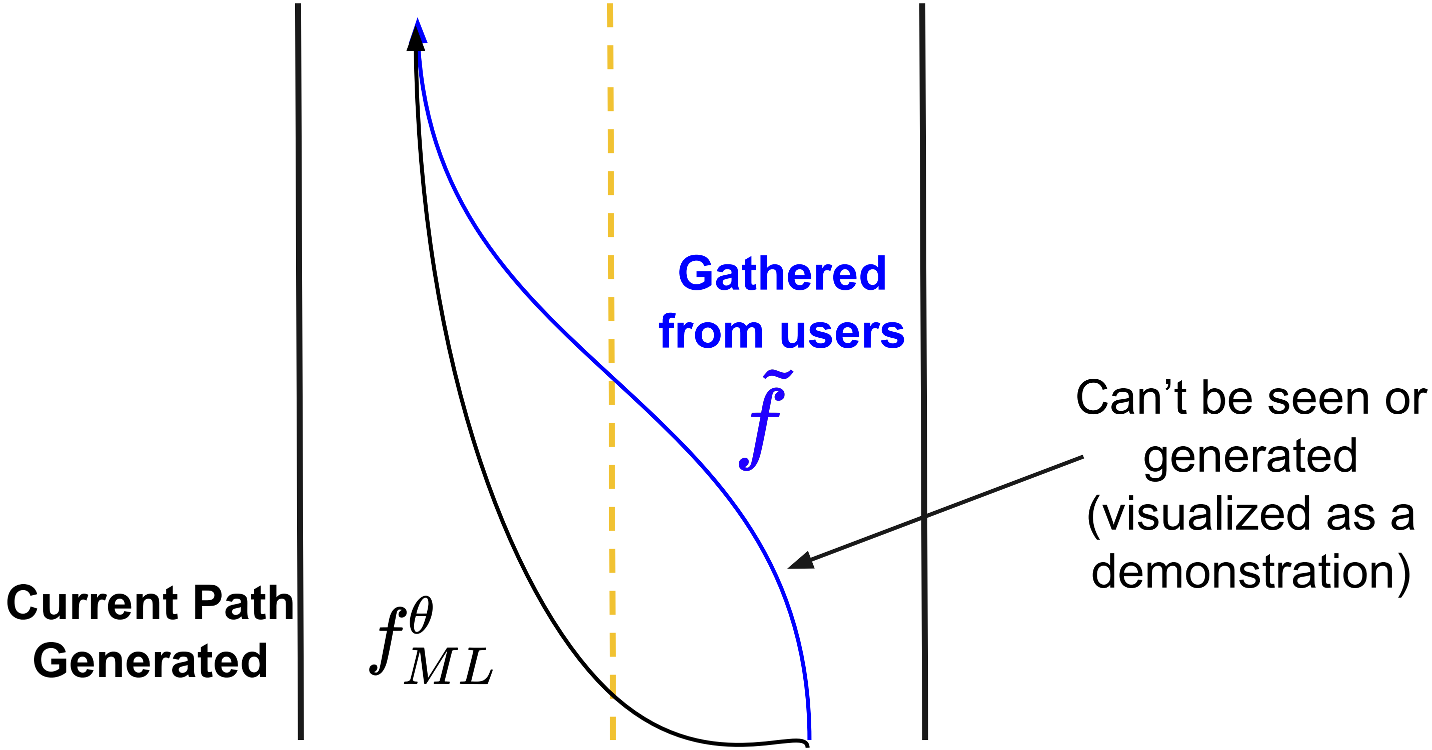
Learning from Demonstration (Algorithm)

Ultimately using this algorithm the **current trajectory** will **converge** **toward** the **demonstrated trajectory**

Now let's look at an example.

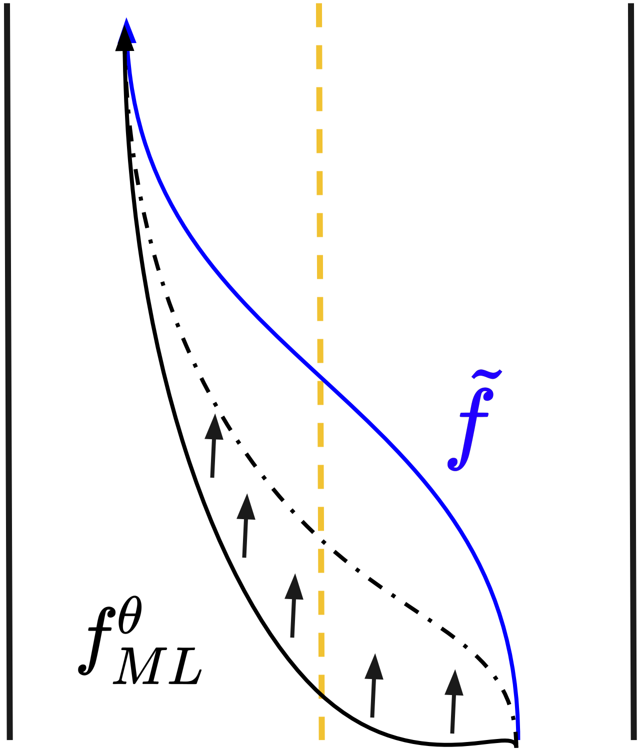
Learning from Demonstration (Example)

Step 1



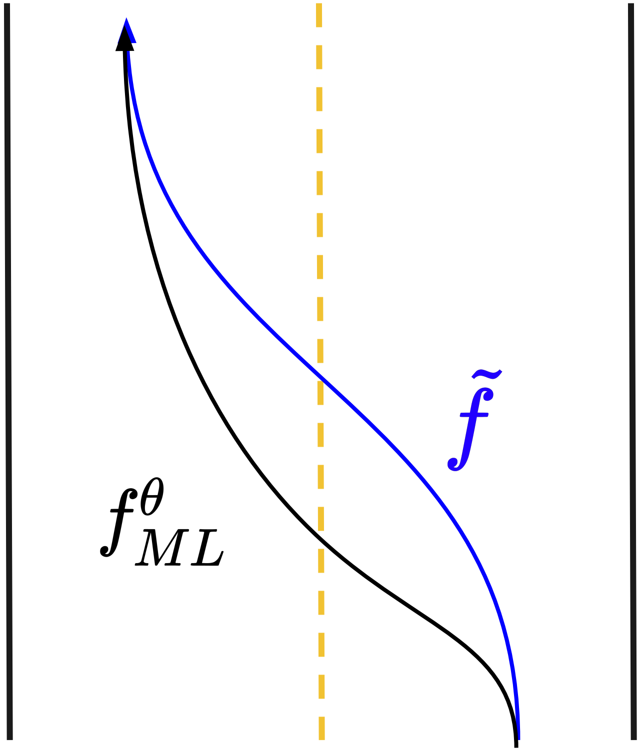
Learning from Demonstration (Example)

Step 1



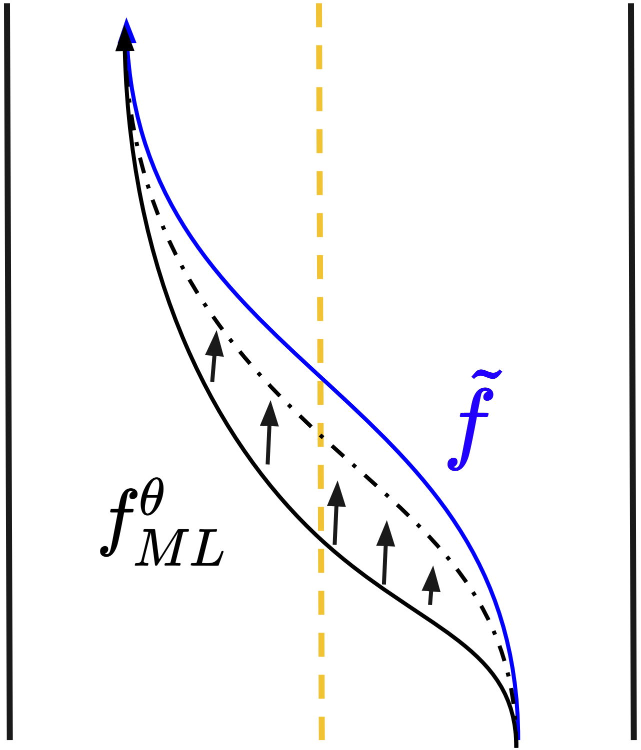
Learning from Demonstration (Example)

Step 2



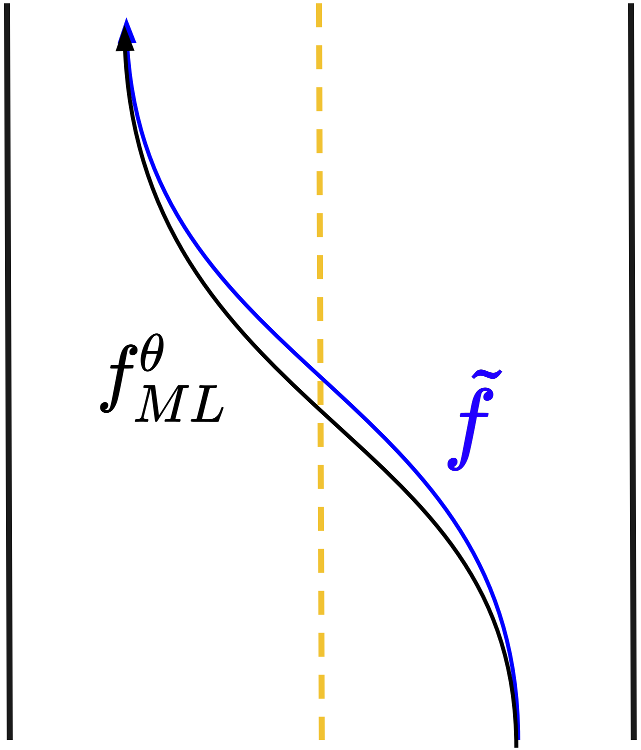
Learning from Demonstration (Example)

Step 2



Learning from Demonstration (Example)

Step N



Data Acquisition

- Used an existing map
- Drivers demonstrated acceleration

  In the velocity range of 20-30 m/s
- Lane changes were also performed
- In total 8 minutes of driving data were collected
- All data was then separated into “lane change” and “lane keeping”

Fig. 1. A Bosch highly automated driving development vehicle.

Learning Individual Navigation Styles (Simulation

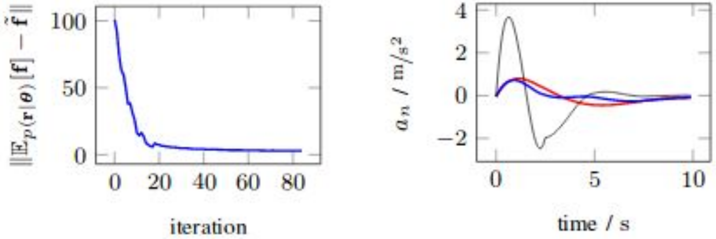


Fig. 2. Left: evolution of the norm of the difference between the empirical feature values and the expected feature values during learning on a dataset of 20 observed trajectories. Right: normal acceleration of a demonstrated trajectory (blue), the initial guess (black), and the optimized trajectory with the final learned policy (red).

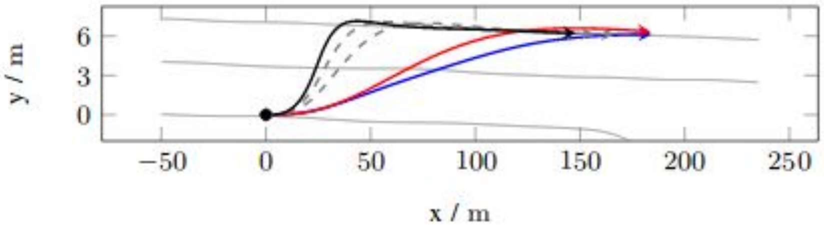
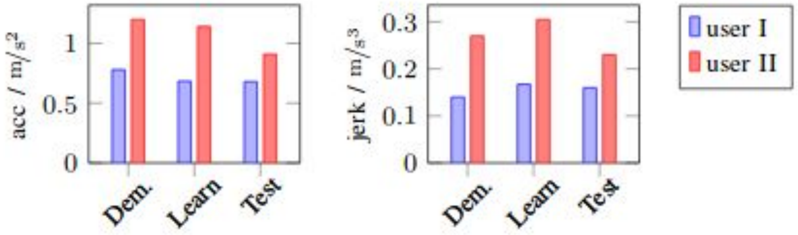


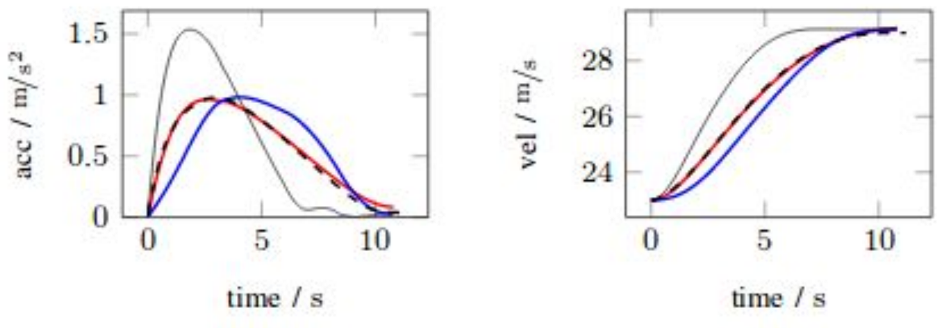
Fig. 3. Demonstrated trajectory (blue), the initial guess (black), and the optimized trajectory with the final learned policy (red). The trajectory shows a change of two lanes to the left. The dashed lines illustrate intermediate policies during the learning phase.



Simulation Testing

The authors ran this on a *realistic simulation environment* and claim:

- That the algorithm was able to run at 5Hz
    - No specs were provided regarding the computing capabilities of the system
- Learning policy is suitable to autonomously control a car



Issues with the approach

- Major safety concerns
  - How to extract **emergency maneuvers form a small set of demonstration** trajectory
  - **A finite number** of demonstrated trajectories may be **insufficient** to **solve an infinite number of situations**
  - Are the listed features sufficient for all cases
- Not guarantees that the selected trajectory is optimal in a given situation
- The set of features might change given the current surrounding
- Not really self-driving, rather lane keeping assistance

Issues with the results

- The testing done using this planner are unsatisfactory
  - No demonstration on autonomous driving in the real world
  - Two users are not sufficient to demonstrate the ability of the planner
  - No clear numeric representation of comfort
- 5 Hz is a concerningly slow planner to deal with all situations and speeds

Discussion

- Should we autonomous vehicle have different driving styles?
- What if the driving style programmed is far too aggressive to be deemed safe?
- What if different users have different comfort levels, can this method account for that?

Thanks!