Imitating Trajectories in Aviation

Imitating Trajectories with Generative Adversarial Imitation Learning

George A. Vouros, T. Kravaris, C. Spatharis, K. Blekas,
A. Bastas, G. Santipantakis

AI-Lab, University of Piraeus
Greece

Outlook of presentation

- Motivation
- Background knowledge
- Problem specification
- ILTP Framework
- Generative Adversarial Imitation Learning (GAIL)
  and Apprenticeship Learning (AppLearn) algorithms
- Evaluation Results
- Conclusions & future work

Motivation

- Imitate experts shaping the flown trajectories, by learning models that incorporate their preferences, strategies in an aggregated way (i.e. their joint policy).
- Do this with (a small number of) historical trajectories in a data-driven way, and with no transformation of trajectories.
- Advance predictions at the pre-tactical and tactical phases.
- This paves the way
  - To detect important stakeholders' features that affect trajectory evolution
  - Compare alternative models based on assumptions on stakeholders' (e.g. airlines) "whims"
  - Detect and distinguish "latent" features that determine modes of behaviour

Background knowledge

- Trajectory (enriched)

An *enriched trajectory state* or *enriched trajectory point* of a trajectory of length $|T|$, is defined to be a triplet $s_{r,i}$=<p_i, t_i, v_i>, where where p_i is a point in the 3D space, v_i is a vector consisting of categorical and/or numerical variables and t_i is a timestamp, with $i \in [0, |T| - 1]$. An *enriched trajectory* T is defined to be a sequence of enriched states $s_{r,i}$=<p_i, t_i, v_i>, $i \in [0, |T| - 1]$.

- Data-driven Trajectory prediction

Assuming a set $\mathbf{T}_E = \{T_{E,i} | i = 1, 2, 3, ...\}$ of historical, demonstrated enriched trajectories, the *trajectory prediction problem* can be defined as follows: Given $\mathbf{T}_E$ and a cost function c, the objective is to predict a trajectory T_{π^*}, generated by a policy

$$\pi^* = \underset{\pi}{\operatorname{argmin}} \ \mathbb{E}_\pi [c(\langle p, t, v \rangle, a)] \tag{1}$$

Background knowledge

- Imitation Learning

Imitation learning studies the problem of learning to perform a specific task in a setting, where the learner has access to expert demonstrations (trajectories).

- Behavioral Cloning : solves **a regression** problem **minimizing the error between the actions demonstrated and the policy actions, over the states of the historical trajectories**.
- Inverse Reinforcement Learning: Two-steps process.

(a) **derive a cost function *c*** that assigns minimal cost to trajectories demonstrated by experts and maximal cost to trajectories generated by other policies and

(b) **find a policy** that minimizes the expected cumulative cost using the cost function.

Background knowledge

- Imitation Learning

Inverse Reinforcement Learning:

Apprenticeship learning: Assumes that the cost function is a linear combination of basis functions, which result to feature vectors over states and actions.

Generative adversarial imitation learning (GAIL): Does not apply restrictive assumptions on the cost function and scales to large, continuous state-action spaces.

Background knowledge

- Imitation Learning

Inverse Reinforcement Learning

Generative adversarial imitation learning (GAIL)

GAIL directly learns the optimal policy from expert demonstrations, quite efficiently, **aiming to bring the distribution of the state-action pairs of the imitator as close as possible to the trajectories demonstrated by the expert**.

GAIL objective:

Discriminator: Maximizes the cost of non-expert trajectories

Discriminator: … and minimizes the cost of expert trajectories

while maximizing the entropy of the policy

$$\min_{\pi} \max_{D \in (0,1)^{SxA}} \mathbb{E}_{\pi}[logD(s,a)] + \mathbb{E}_{\pi_E}[log(1 - D(s,a))] - \lambda_H H(\pi)$$

Problem specification (trajectory prediction)

Given a set of expert (demonstrated) trajectories: $\mathbf{T}_E$

Consider:

- Time step: Δt
- Actions (continuous): $(\Delta l, \Delta f, \Delta h)$

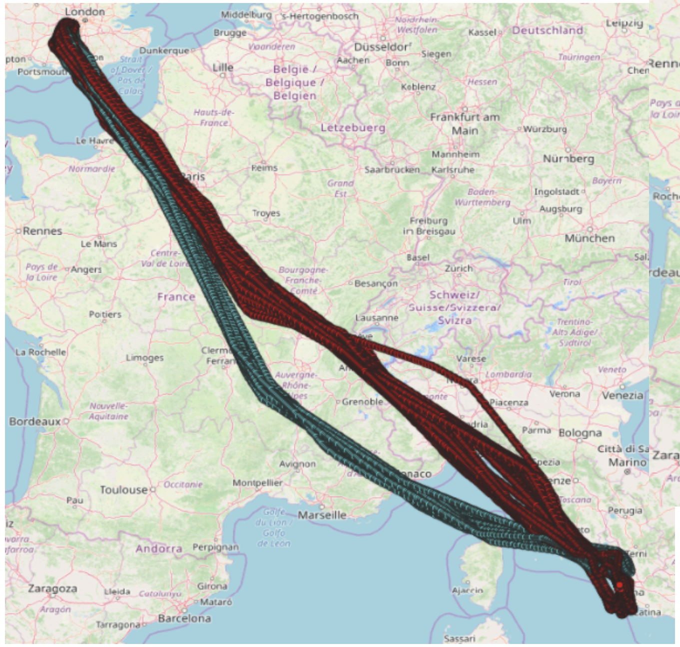
Data Driven Aircraft Trajectory Prediction Problem:

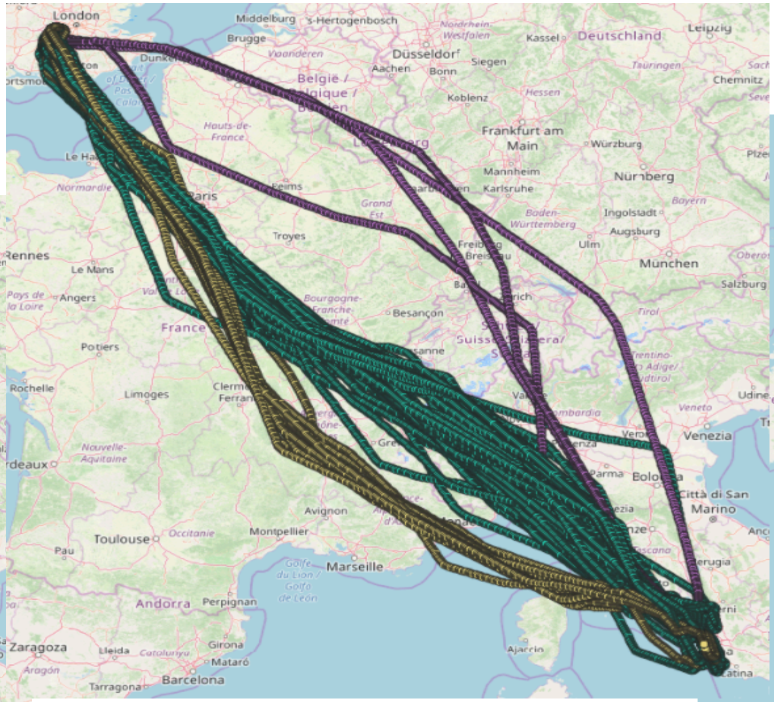
Given a set $\mathbf{T}_E = \{T_{E,i}, i = 1, \ldots N\}$ of historical aircraft trajectories, and a time step Δt, we need to determine a policy $\pi \in \Pi$ which optimizes the GAIL objective.

This policy, given the initial state of aircraft s_0, determines the evolution of the trajectory at any time instant $t = t_0 + (\Delta t * i)$, $i = 1, 2, 3\ldots$. Specifically, it determines $\pi(a|s, \Delta t)$, i.e. the action to be performed for the evolution of the aircraft position at state $s = \langle p, t, v \rangle$.

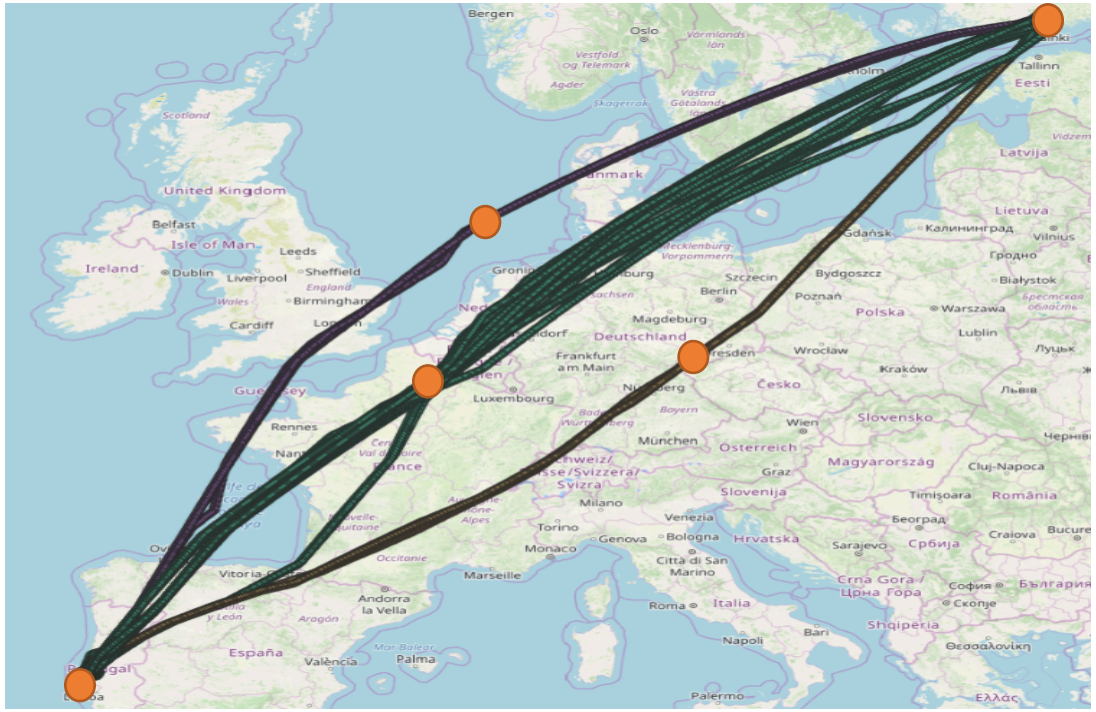
Problem specification (trajectory prediction)

- But… there are multiple modes of behavior…





Problem specification



Given a set of expert (demonstrated) trajectories: $\mathbf{T}_E$

Consider:

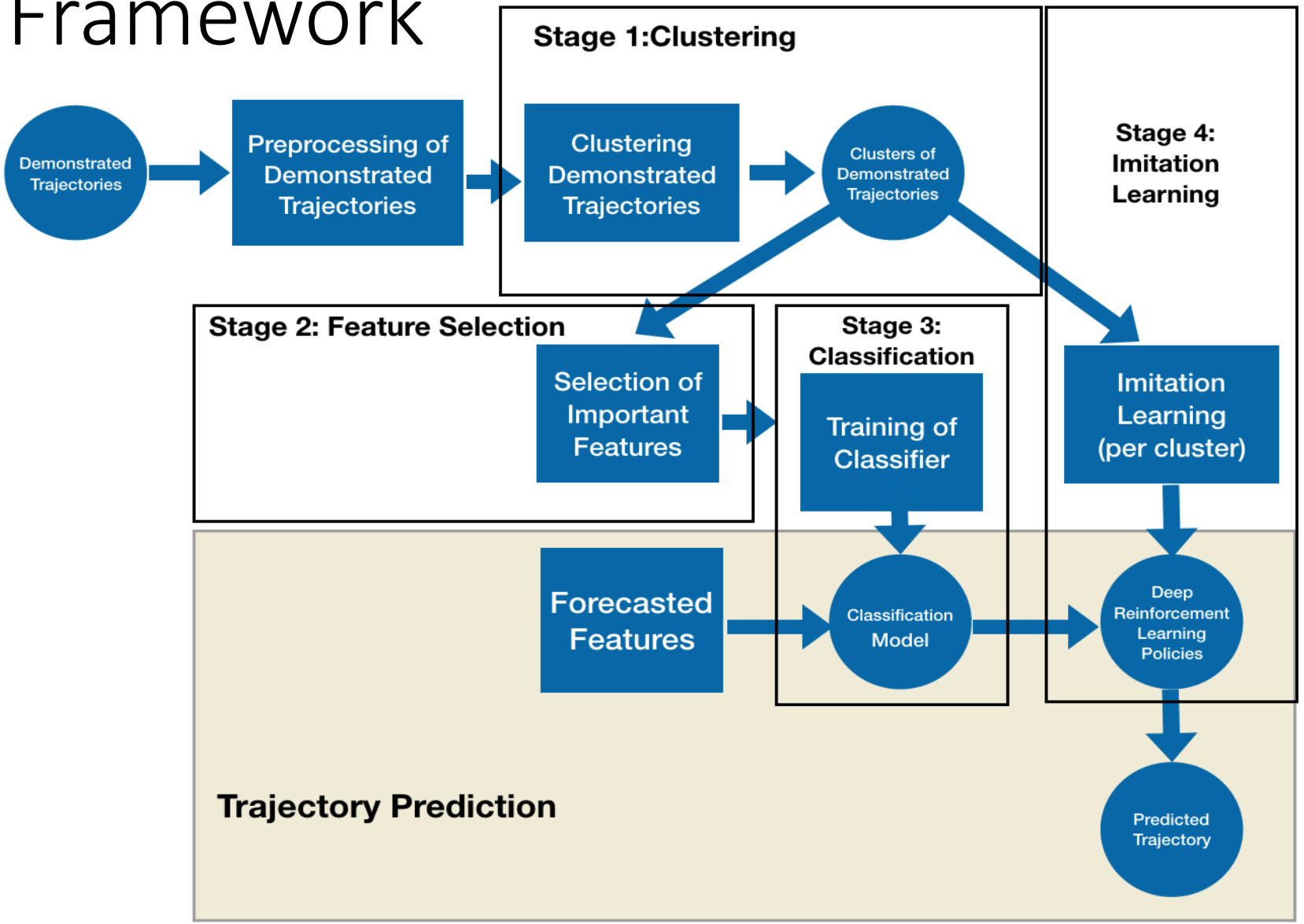
- Time step: Δt
- Actions (continuous): $(\Delta l, \Delta f, \Delta h)$
- K classes of trajectories $C_{E,l}$
- A set **S**^{f} of "landmark" states with forecast time and values of features $s^f_{fix_r} = \langle (l_{fix}, f_{fix}, h_{fix}), t^f, v^f \rangle$
- A set of forecast features T^f of the **future trajectory** (as a whole):

  E.g. aircraft type, time/day, estimated TOW, average FIR en-route charges, etc.

Data Driven Aircraft Trajectory Prediction Problem:

Determine
(a) the specific class $C_{E,\sigma} \subseteq \mathbf{T_E}$ of expert trajectories that most probably the **future trajectory** (i.e the trajectory with forecasted features T^f crossing the points in **S**^{f}) belongs, and
(b) a policy $\pi_\sigma \in \mathbf{\Pi}$, which optimizes the GAIL objective, and which specifies the evolution of the trajectory at any time instant $t0 + (\Delta t * i)$, i = 1, 2, 3...

Imitation Learning for Trajectory Prediction ILTP Framework



Imitating trajectories: GAIL

GAIL employs

- a generative trajectory model $\mathbf{G}_\theta$ that models the policy
- a discriminative classifier $\mathbf{D}_w$ that distinguishes between the distribution of data (i.e. state action pairs) generated by the policy and the demonstrated data.

- GAIL alternates between an **Adam gradient step on w** to increase the GAIL objective with respect to D, and
- a step on using the **Trust Region Policy Optimization (TRPO) Deep RL algorithm to improve the policy** by tuning θ so as to decrease the GAIL objective.

A.Bastas, T.Kravaris, G.A.Vouros “Data Driven Aircraft Trajectory Prediction with Deep Imitation Learning”, In arXiv cs.LG, 2020.

Imitating trajectories: AppLearn

Applearn assumes a linear model of the reward function:

$$R(s) = \mathbf{w}^T \phi(s)$$

where ϕ are basis functions representing features of states (spatial and weather features).

AppLearn follows the 2 step process, iteratively:

- determine the proper weight vector in order to shape the expert's behaviour (solving a quadratic programming problem)
- Use
  - DQN exploiting the estimated reward function to determine a policy in 2D
  - A regression NN for predicting the altitude at each state.

C.Spatharis, K.Blekas, G.A.Vouros "Apprenticeship learning of flight trajectories prediction with inverse reinforcement learning", In Proc of SETN 2020.

Experiments: Cases

Short (single FIR) trajectories

- BCN-MAD (April 2016)
  - 528 trajectories
  - 2 clusters of 250 and 278 trajectories

Long (multi FIR) trajectories

- LHR-FCO (Jan & Jul 2019)
  - 218 & 242 trajectories
  - 2 clusters of 23 and 195 trajectories & 3 clusters of 4, 19 and 219 trajectories
- HEL-LIS (Jan & Jul 2019)
  - 49 & 55 trajectories
  - 3 clusters of 4,3 and 42 trajectories & 3 clusters of 2, 9 and 44 trajectories

Trajectories have been enriched with 4 meteo-features *(temperature, geopotential height, u-componentof wind, v-component of wind*) and *the aircraft type*.

Experiments: Measures

RMSE: computing trajectories corresponding points using DTW

- 1D

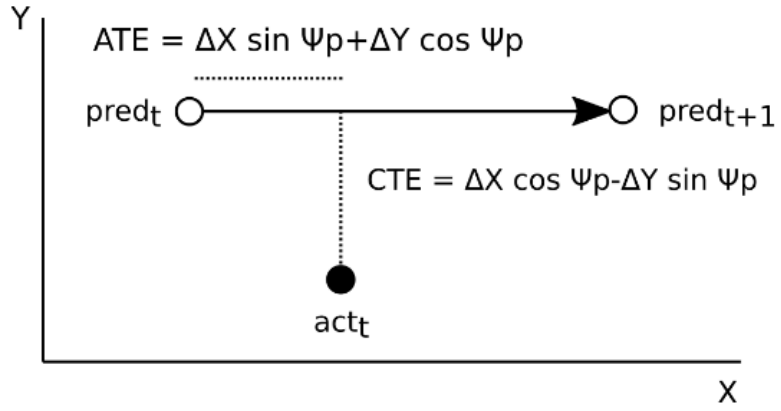
$$RMSE(var) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(var_{pred} - var_{actual})^2}$$

- 3D

$$RMSE_{3D} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\frac{\sqrt{\sum_{d=1}^{Dim}(var_{d_{pred}} - var_{d_{actual}})^2}}{Dim}}.$$

Experiments: Measures

Track Errors : computing trajectories corresponding points using timestamps



Along track (ATE) and cross track (CTE) errors w.r.t. the predicted trajectory's points at times t and $t+1$, denoted by $pred_t$ and $pred_{t+1}$, and the actual trajectory's point act_t at time t. $\Delta X = X_p - X_a$ is the difference between the X coordinate (longitude) of $pred_t$ (X_p) and the X coordinate of act_t (X_a). $\Delta Y = Y_p - Y_a$ is the difference between the Y coordinate (latitude) of $pred_t$ (Y_p) and the y coordinate of act_t (Y_a). Ψp denotes the course, in our case the bearing (i.e. the angle between the direction of the trajectory and the North), of the predicted trajectory.

Experiments: Results

- One vs Multiple Policies – GAIL vs AppLearn

GAIL

BCN-MAD (April 2016)

AppLearn

	OnePolicy				MultPolicies			
	Long	Lat	Alt	3D	Long	Lat	Alt	3D
0	14350	8347	457	17279	10932	5577	333	12652
0.2	13780	8311	550	16825	10252	5477	402	12048
0.5	9726	8847	427	14066	7490	6679	324	10565
0.7	5979	7059	246	9916	4430	6360	188	8033

	OnePolicy				MultPolicies			
	Long	Lat	Alt	3D	Long	Lat	Alt	3D
0	23822.7	10547.6	5217.5	26902.9	14693.9	5695.9	344.4	15975.2
20	16722.4	12775.6	5585.5	22257.5	13178.4	7020.0	285.1	14996.5
50	15213.1	12336.1	3971.2	20310.6	10753.4	4066.9	302.7	11548.9
70	12709.9	13438.7	2161.4	18893.1	11211.4	3893.9	265.0	11911.4

	OnePolicy				MultPolicies			
	ATE	CTE	V	ETA	ATE	CTE	V	ETA
0	-31.6	577.0	67.0	245.96	305.8	154.0	23.8	274.10
0.2	-99.4	808.5	121.3	288.65	454.5	391.8	57.6	268.70
0.5	984.9	1657.1	115.6	398.34	826.0	875.1	79.5	325.84
0.7	851.5	1540.7	25.8	460.56	1065.0	1133.0	5.1	369.03

	OnePolicy				MultPolicies			
	ATE	CTE	V	ETA	ATE	CTE	V	ETA
0	-4510.5	566.6	-4182.9	620.9	3111.2	2816.6	-26.2	1989.3
20	-5836.2	1236.1	-4665.9	248.3	-5007.2	-265.8	55.1	185.6
50	-5072.6	350.1	-3161.2	155.4	-3646.3	-13.3	141.3	88.4
70	-4886.3	136.2	-1738.8	93.1	-3063.8	-341.7	44.4	74.9

Experiments: Results

LHR-FCO (July 2019)

- GAIL vs AppLearn

GAIL (for the larger cluster)

M	Long	Lat	Alt	3D
0	23371.12	20888.65	372.33	18351.99
0.2	24427.65	20568.25	359.35	18733.01
0.5	20274.53	18497.25	370.98	16209.15
0.7	14313.57	14444.13	539.25	12126.93

ATE	CTE	VE	ETA
18689.42	17874.3	636.2	457.1
19273.79	19362.4	621.78	615.12
15758.31	16399.46	629.06	791.83
13205.18	11294.95	659.35	910.28

GAIL (for all clusters)

Long	Lat	Alt	3D
14586	13797	213	20583
20618	19412	316	29228
17644	17261	273	25384
18011	16576	497	25535

ATE	CTE	V	ETA
1541.5	-926.5	7.6	1064.0
1177.4	-757.8	71.2	1196.8
7454.3	-2785.5	83.8	1671.4
8735.7	-2924.9	213.7	1859.8

AppLearn (for all clusters)

Long	Lat	Alt	3D
22140.9	31579.4	3909.1	38873.4
14266.2	19281.8	624.3	24001.2
16242.0	19984.4	770.3	25789.5
14789.7	15939.0	374.3	21878.2

ATE	CTE	V	ETA
-12852.2	636.7	-1843.1	20789.4
-9374.0	768.4	209.3	-356.0
-10384.2	1866.8	400.2	-305.0
-6355.7	-359.1	41.5	162.3

Experiments: Results

HEL-LIS (July 2019)

- GAIL vs AppLearn

GAIL (for the larger cluster)

M	Long	Lat	Alt	3D
0	88448.14	95173.02	1096.41	75950.75
0.2	91184.7	100957.56	1062.17	79921.51
0.5	90334.3	92006.24	1090.32	76575.08
0.7	77587.38	76998.23	1966.88	64771.15

ATE	CTE	VE	ETA
77341.27	59731.61	1074.71	801.44
81309.09	52941.16	1052.04	978.19
81468.87	49669.47	1252.01	1080.75
81990.64	46206.48	1691.44	1113.12

GAIL (for all clusters)

	July 2019		
Long	Lat	Alt	3D
24828	23020	333	34585
37886	39584	324	56449
49978	50816	353	73469.5
50456	51543	598	74556

ATE	CTE	V	ETA
1837.0	-3712.4	78.5	715.9
12646.4	3352.5	76.4	1615.2
24161.9	4984.9	83.5	2344.4
31294.1	1871.5	87.2	2852.2

AppLearn (for all clusters)

Long	Lat	Alt	3D
32667.9	38158.7	5394.5	50605.5
37394.2	50782.1	166.4	63084.2
40214.2	49522.0	236.0	63821.2
39418.1	49193.3	352.7	63052.5

ATE	CTE	V	ETA
-11732.2	203.6	-2776.5	19499.0
-27355.4	686.6	13.1	262.0
-29390.1	-555.5	-10.7	1027.0
-24991.9	-1396.9	-135.8	430

Conclusions

- Imitation Learning potential:
  - **Effectiveness**: High (in accuracy) but not in stages with large variations (landing phase)
  - **Efficiency**: Large number of training episodes but with low-sampling efficiency
  - **Tradeoff between arbitrary and linear cost function**: It seems that a linear cost function is not appropriate. However, AppLearn scores better in landing phases compared to GAIL.
- Imitation learning provides a
  - "natural" (in terms of inherently dealing with trajectories),
  - simple (not requiring feature engineering),
  - unconstrained (without considering flight plans as constraints or fitting trajectories into a specific spatio-temporal resolution),
  - continuous (in terms of state and actions' features considered) and
  - generic (in terms of being able to apply to different types of trajectories and at any flight phase) approach,
  - Pure data-driven (not requiring any model-based mechanistic prediction method -although it would be interesting to investigate interactions)

  TP approach

Future work

(a) verifying the effectiveness of the methods for different origin-destination airports, while being trained using larger training datasets

(b) exploiting flight plans to constrain the prediction pipeline, and detecting deviations from intended routes

(c) extending the method to deal inherently with different modes of trajectory evolution, and

(d) generalizing beyond specific origin-destination pairs to deal with predictions at large scale.

Many Thanks to

This work has been done in the context of the
CF project
Data-Driven Trajectory Imitation with Reinforcement Learning

Imitating Trajectories
in Aviation

University of Piraeus Research Centre