project 18: Bayesian imitation learning

task 18.1: from locations to velocities

Recall that the file `q3dm1-path2.csv` contains human trajectory data, i.e. a sequence

$$\boldsymbol{x}[0] : \boldsymbol{x}[1] : \boldsymbol{x}[2] : \boldsymbol{x}[3] : \cdots : \boldsymbol{x}[n]$$

of 3D locations the avatar of a human player was seen at while moving around the Quake III map *q3dm1*.

Given the location data in `q3dm1-path2.csv`, compute a sequence of velocities

$$\boldsymbol{v}[0] : \boldsymbol{v}[1] : \boldsymbol{v}[2] : \boldsymbol{v}[3] : \cdots : \boldsymbol{v}[n-1]$$

where

$$\boldsymbol{v}[t] = \boldsymbol{x}[t+1] - \boldsymbol{x}[t]$$

task 18.2: from velocities to action primitives

Note that the locations $\boldsymbol{x}[t]$ can be understood as states the player was in and that the velocities $\boldsymbol{v}[t]$ you just computed can be understood as moves or actions the player performed in each state. Under this interpretation, the equation

$$\boldsymbol{x}[t+1] = \boldsymbol{x}[t] + \boldsymbol{v}[t]$$

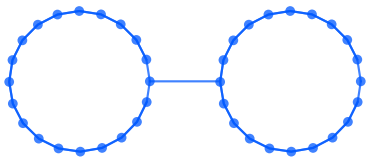
can be read a state transition.

Previously, you already used SOMs to determine a set $S = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{k_S}\}$ of prototypical states.

Something similar can be done to determine a set $A = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{k_A}\}$ of prototypical actions or *action primitives*. Simply run k_A-means clustering on the velocity vectors $\boldsymbol{v}[t]$ and consider the k_A resulting cluster centroids to be the action primitives $\boldsymbol{a}_i$.

task 18.3: simple Bayesian imitation learning

Fit a self organizing map (SOM) of k_S neurons to the $\boldsymbol{x}[t]$. This SOM should have the two-ring topology



from the last project. Set the number of SOM neurons to $k_S = 20$ and consider its trained weights as the states of a state space $S = \{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{k_S}\}$.

Run k-means clustering on the $\boldsymbol{v}[t]$ to obtain a set $A = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{k_A}\}$ of action primitives. Experiment with different choices of k_A, but, when in doubt, choose $k_A = 9$.

Given S and A, map the $\boldsymbol{x}[t]$ and $\boldsymbol{v}[t]$ to (the index of) their corresponding prototypes or primitives. To this end use

$$s[t] = \underset{j}{\operatorname{argmin}} \left\|\boldsymbol{x}[t] - \boldsymbol{s}_j\right\|^2$$

$$a[t] = \underset{i}{\operatorname{argmin}} \left\|\boldsymbol{v}[t] - \boldsymbol{a}_i\right\|^2$$

Given the $s[t]$ and $a[t]$ compute a $k_S \times k_A$ matrix $\boldsymbol{P}$ with entries

$$p_{sa} = p\big(s, a\big) = \frac{n_{sa}}{\sum_i \sum_j n_{ij}}$$

where n_{sa} counts occurrences of the pair $\big(s, a\big)$ in the set of pairs $\Big\{\big(s[t], a[t]\big)\Big\}_{t=0}^{n-1}$

Note that, once you have estimated all the joint probabilities $p\big(s, a\big)$, you can compute the conditional probability

$$p\big(a \mid s\big) = \frac{p\big(s, a\big)}{p\big(s\big)}$$

This, in turn can be used to automatically generate a sequence of locations

$$\hat{\boldsymbol{x}}[0] : \hat{\boldsymbol{x}}[1] : \hat{\boldsymbol{x}}[2] : \hat{\boldsymbol{x}}[3] : \cdots : \hat{\boldsymbol{x}}[T]$$

that resembles the original sequence of locations. In other words, it can be used to create human-like movements for an NPC or game bot.

In the lecture, we discussed that the following algorithm can accomplish this simple form of Bayesian imitation:

---

initialize a location $\hat{\boldsymbol{x}}[0]$, for instance

$$\hat{\boldsymbol{x}}[0] = \boldsymbol{x}[0]$$

for $t = 1, \ldots, T$

$$s[t] = \operatorname*{argmin}_j \left\| \hat{\boldsymbol{x}}[t-1] - \boldsymbol{s}_j \right\|^2$$

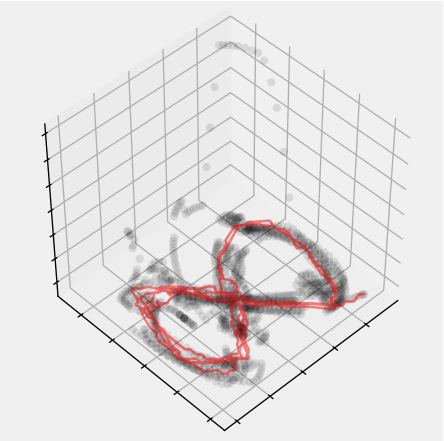
$$a[t] \sim p\big(a \;\big|\; s[t]\big)$$

$$\hat{\boldsymbol{v}}_1[t] = \boldsymbol{a}_{a[t]}$$

$$\hat{\boldsymbol{v}}_2[t] = \boldsymbol{s}_{s[t]} - \hat{\boldsymbol{x}}[t-1]$$

$$\hat{\boldsymbol{x}}[t] = \hat{\boldsymbol{x}}[t-1] + \eta \cdot \hat{\boldsymbol{v}}_1[t] + \big(1-\eta\big) \cdot \hat{\boldsymbol{v}}_2[t]$$

Implement this algorithm and plot the trajectories it produces. If you plot these trajectories together with the original location data, your results may ideally look like this



Experiment with different choices of the parameter $0 < \eta < 1$ and discuss what you observe.