Midterm 4

Ziebart et al., Maximum Entropy Inverse Reinforcement Learning, AAAI 2008

Mario Correddu

May 31, 2022

Introduction to the problem

- Objective: learn to predict the behaviour an agent would choose, given some demonstrated behaviour
- difficult for general purpose statistical machine learning algorithms: need to reason far into the future
- An approach is to structure the space of learned policies to be solutions of Markov Decision Problems
- The agent tries to optimize an unknown reward function: we then just need to recover a reward function that induces the demonstrated behaviour
- Problem: demonstrated behaviour is prone to noise and imperfect behaviour
- Proposed solution: use the principle of maximum entropy to solve the ambiguity

Model description: Imitation learning setting:

- Agents behaviour: a path of states s_i and actions a_i is observed
- Assume the agent is trying to optimize a function that linearly maps the features of each state to a state reward value

$$reward(f_\zeta) = \theta^T f_\zeta = \sum_{s_j \in \zeta} \theta^T f_{s_j}$$

- Abbeel e Ng (2004) proved that, assuming a linear reward, in order to achieve the same performance as the observed policy, a necessary and sufficient condition is to match feature expectations:

$$\sum_{Path \zeta_i} P(\zeta_i) f_{\zeta_i} = \tilde{f}$$

Where ζ_i are trajectories demonstrated by the agent and $\tilde{f} = \frac{1}{m} \sum_{\zeta_i} f_{\zeta_i}$.

Model description: deterministic case

- Consider distributions of all possible behaviours (paths)
- distributions may exhibit preference for some paths over others that is not implied by path features
- resolve this ambiguity by maximum entropy: we choose the distribution that doesn't exhibit any additional preferences beyond matching feature expectations
- the resulting distribution is

$$P(\zeta_i) = \frac{1}{Z(\theta)} e^{\theta^T f_{s_j}}$$

.

Model description: non-deterministic case

- Now the distribution has to take into account the randomness of the MDP given by some state transition distribution T
- apply maximum entropy to the paths conditioned on T and constrainted to match feature expectations. So by definition of conditional expectation over discrete values we have:

$$P(\zeta|\theta, T) = \sum_{o \in \mathcal{T}} P_T(o) \frac{1}{Z(\theta, o)} e^{\theta^T f_{s_j}} I_{\zeta \in o}$$

where $\mathcal{T}$ is the space of all possible action outcomes, and $I_{\zeta \in o}$ is the indicator function that is 1 when ζ is compatible with o.

- Simplify the formula: we assume that transition randomness has a limited effect on behaviour and that the partition function is constant for all $o \in T$ and get:

$$P(\zeta|\theta, T) = \frac{e^{\theta^T f_{s_j}}}{Z(\theta, T)} \prod_{s_{t+1}, a_t, s_t \in \zeta} P_T(s_{t+1}|a_t, s_t)$$

.

Model description: Key formula

The formula we derived:

$$P(\zeta|\theta, T) = \frac{e^{\theta^T f_{s_j}}}{Z(\theta, T)} \prod_{s_{t+1}, a_t, s_t \in \zeta} P_T(s_{t+1}|a_t, s_t)$$

Under this model, as in deterministic one, plans with equivalent rewards have equal probabilities, and plans with higher rewards are exponentially more preferred, but now everything is weighted by the probability of the transitions. This way we get the dependence only on the rewards function, given by the Boltzmann distribution, as well as the influence of the transition distribution.

Model description: Learning

To maximize the entropy we maximize the likelihood wrt to the distribution we derived:

$$\arg\max_{\theta} \sum_{examples} log(P(\tilde{\zeta}, \theta, T)$$

And we notice that its gradient has a very interesting form:

$$L(\theta) = \tilde{f} - \sum_{\zeta} P(\zeta|\theta, T) f_{\zeta}$$

meaning that that the learner will perform equivalently to the agent demonstrated behaviour. We can then rewrite the whole thing also as:

$$L(\theta) = \tilde{f} - \sum_{\zeta} D_{s_i} f_{s_i}$$

where D_{s_i} is the expected visitation frequency of s_i, which can be calculated efficiently with an algorithm similar to the forward backward algorithm for Markov random fields.

Empirical results

The method described so far was applied to the problem of imitation learning of driver route choices. We use 3 different measures of performance:

- amount of distance shared between model's most likely path and demonstrated one
- percentage of testing path that is matched for at least 90 % of its length
- average log probability of training set under the model

	Matching	**90% Match**	**Log Prob**
Time-based	72.38%	43.12%	N/A
Max Margin	75.29%	46.56%	N/A
Action	77.30%	50.37%	-7.91
Action (costs)	77.74%	50.75%	N/A
MaxEnt paths	**78.79%**	**52.98%**	**-6.85**

And see that our model outperforms every other competitor in every aspect.

Final comments

The idea of defining a distributions over all paths and apply the maximum entropy principle based on the rewards, seems quite natural, but while the derived distribution is easy to handle, it seems also to imitate the agent's demonstrated behaviour better the other kind of models.