Title: Apprenticeship learning

URL: https://en.wikipedia.org/wiki/Apprenticeship_learning

PageID: 19463198

Categories: Category:Machine learning

In artificial intelligence , apprenticeship learning (or learning from demonstration or imitation learning) is the process of learning by observing an expert. It can be viewed as a form of supervised learning , where the training dataset consists of task executions by a demonstration teacher.

Mapping function approach

Mapping methods try to mimic the expert by forming a direct mapping either from states to actions, or from states to reward values. For example, in 2002 researchers used such an approach to teach an AIBO robot basic soccer skills.

Inverse reinforcement learning approach

Inverse reinforcement learning (IRL) is the process of deriving a reward function from observed behavior. While ordinary "reinforcement learning" involves using rewards and punishments to learn behavior, in IRL the direction is reversed, and a robot observes a person's behavior to figure out what goal that behavior seems to be trying to achieve. The IRL problem can be defined as:

Given 1) measurements of an agent's behaviour over time, in a variety of circumstances; 2) measurements of the sensory inputs to that agent; 3) a model of the physical environment (including the agent's body): Determine the reward function that the agent is optimizing.

IRL researcher Stuart J. Russell proposes that IRL might be used to observe humans and attempt to codify their complex "ethical values", in an effort to create "ethical robots" that might someday know "not to cook your cat" without needing to be explicitly told. The scenario can be modeled as a "cooperative inverse reinforcement learning game", where a "person" player and a "robot" player cooperate to secure the person's implicit goals, despite these goals not being explicitly known by either the person nor the robot.

In 2017, OpenAI and DeepMind applied deep learning to the cooperative inverse reinforcement learning in simple domains such as Atari games and straightforward robot tasks such as backflips. The human role was limited to answering queries from the robot as to which of two different actions were preferred. The researchers found evidence that the techniques may be economically scalable to modern systems.

Apprenticeship via inverse reinforcement learning (AIRP) was developed by in 2004 Pieter Abbeel , Professor in Berkeley 's EE CS department, and Andrew Ng , Associate Professor in Stanford University 's Computer Science Department. AIRP deals with " Markov decision process where we are not explicitly given a reward function, but where instead we can observe an expert demonstrating the task that we want to learn to perform". AIRP has been used to model reward functions of highly dynamic scenarios where there is no obvious reward function intuitively. Take the task of driving for example, there are many different objectives working simultaneously - such as maintaining safe following distance, a good speed, not changing lanes too often, etc. This task, may seem easy at first glance, but a trivial reward function may not converge to the policy wanted.

One domain where AIRP has been used extensively is helicopter control. While simple trajectories can be intuitively derived, complicated tasks like aerobatics for shows has been successful. These include aerobatic maneuvers like - in-place flips, in-place rolls, loops, hurricanes and even auto-rotation landings. This work was developed by Pieter Abbeel, Adam Coates, and Andrew Ng - "Autonomous Helicopter Aerobatics through Apprenticeship Learning"

System model approach

System models try to mimic the expert by modeling world dynamics.

Plan approach

The system learns rules to associate preconditions and postconditions with each action. In one 1994 demonstration, a humanoid learns a generalized plan from only two demonstrations of a repetitive ball

collection task.

Example

Learning from demonstration is often explained from a perspective that the working Robot-control-system is available and the human-demonstrator is using it. And indeed, if the software works, the Human operator takes the robot-arm, makes a move with it, and the robot will reproduce the action later. For example, he teaches the robot-arm how to put a cup under a coffeemaker and press the start-button. In the replay phase, the robot is imitating this behavior 1:1. But that is not how the system works internally; it is only what the audience can observe. In reality, Learning from demonstration is much more complex. One of the first works on learning by robot apprentices (anthropomorphic robots learning by imitation) was Adrian Stoica's PhD thesis in 1995.

In 1997, robotics expert Stefan Schaal was working on the Sarcos robot-arm. The goal was simple: solve the pendulum swingup task . The robot itself can execute a movement, and as a result, the pendulum is moving. The problem is, that it is unclear what actions will result into which movement. It is an Optimal control -problem which can be described with mathematical formulas but is hard to solve. The idea from Schaal was, not to use a Brute-force solver but record the movements of a human-demonstration. The angle of the pendulum is logged over three seconds at the y-axis. This results into a diagram which produces a pattern.

In computer animation, the principle is called spline animation . That means, on the x-axis the time is given, for example 0.5 seconds, 1.0 seconds, 1.5 seconds, while on the y-axis is the variable given. In most cases it's the position of an object. In the inverted pendulum it is the angle.

The overall task consists of two parts: recording the angle over time and reproducing the recorded motion. The reproducing step is surprisingly simple. As an input we know, in which time step which angle the pendulum must have. Bringing the system to a state is called “Tracking control” or PID control . That means, we have a trajectory over time, and must find control actions to map the system to this trajectory. Other authors call the principle “steering behavior”, because the aim is to bring a robot to a given line.

See also

Inverse reinforcement learning

References