# Towards a Unified Framework for Learning from Observation

Santiago Ontañón (IIIA-CSIC, Spain)
José L. Montaña (Universidad de Cantabria, Spain)
Avelino J. Gonzalez (University of Central Florida, USA)

# Motivation

- Many disconnected approaches in the literature
- Lack of a common framework to compare

# Outline

- Learning from Observation
- A Unified Framework
- Levels of Difficulty of LFO
- Statistical Formulation
- Conclusions

# Outline

- Learning from Observation
- A Unified Framework
- Levels of Difficulty of LFO
- Statistical Formulation
- Conclusions

# Learning from Observation

- Learn to perform a task solely by observing the external behavior of another agent

# Learning from Observation

- Supervised learning: learning a mapping from input variables to output variables
- LfO: learning a control function (which might have internal state)

# Many Approaches

- Can be traced back to 1979, with different names:
  - Learning from Observation
  - Learning from Demonstration
  - Imitation Learning
  - Apprenticeship Learning
  - Programming by Demonstration

# Many Approaches

- Reinforcement Learning Techniques
- Case-based Reasoning
- Decision Trees, Neural Networks, etc.
- Generic Algorithms
- Inductive Logic Programming
- Cognitive Architectures (SOAR, etc.)
- etc.

[Argall et al. 2009] "A survey of robot learning from demonstration"

# Applications

- Domains with complex behaviors:
  - Robotics
  - Computer games
  - Training and simulation
  - Automated programming
  - etc.

# Related Problems

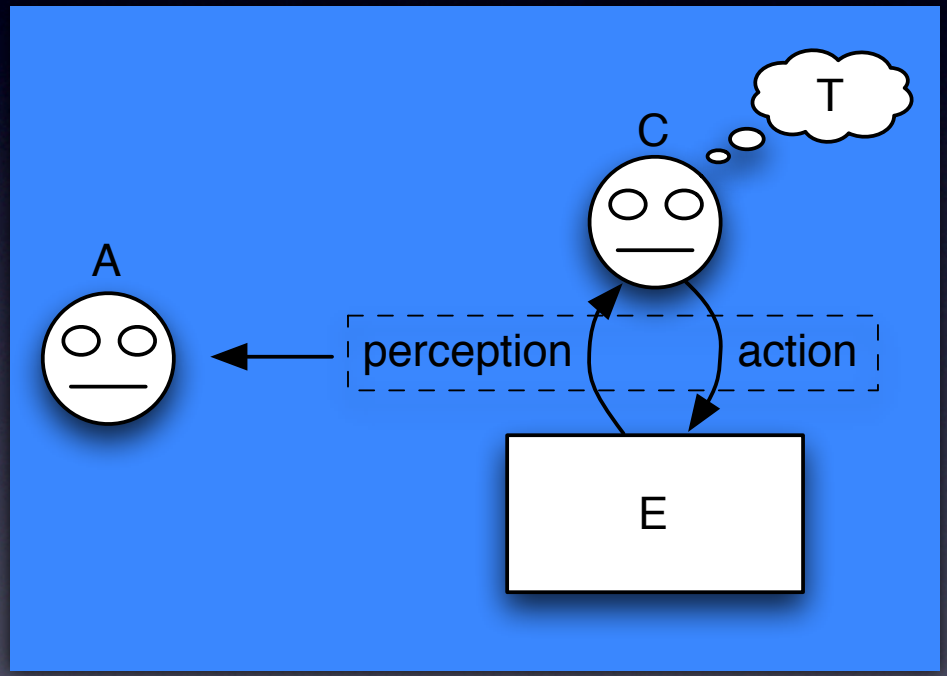
- Inverse Reinforcement Learning:
  - Given behavior (optimal policy, or trajectories), learn the reward function
- Workflow reconstruction / Automata discovery

# Outline

- Learning from Observation
- A Unified Framework
- Levels of Difficulty of LFO
- Statistical Formulation
- Conclusions

# Vocabulary

- An environment E
- An expert (or actor) C
- A task T
- A learning agent A



# Learning Traces

- The learning agent A can only observe the interaction of the expert C with the environment, E, not the internal state of C:
  - perceptions (state of E by A): X
  - actions: Y

$$LT = [(t_1, x_1, y_1), ..., (t_n, x_n, y_n)]$$

# LFO Task

- Given:
  - A set of learning traces $LT_1$, ..., $LT_k$
  - An environment E (characterized by a set of input variables X, and a set of control variables Y)
  - Optionally, a description of the task T
- Learn:
  - A behavior B that "behaves like" C in achieving task T in E

# "Behaves like"

- If no T is specified:
  - LFO is equivalent to learning to predict C's actions
- If T is specified:
  - LFO's performance must take into account both predicting C's actions and accomplishing T

# Measuring Performance

- In traditional ML, performance is measured by leaving some examples out of the training set: test set
  - In LFO, test set would be a set of traces
  - Comparing traces is not trivial
- Achievement of task T must be taken into account

# Measuring Performance

- Evaluate performance: how well is T achieved
- Evaluate output: how well the model predicts expert actions (like traditional ML)
- Evaluate model: inspect the learned model (typically by human inspection)

# Outline

- Learning from Observation
- A Unified Framework
- Levels of Difficulty of LFO
- Statistical Formulation
- Conclusions

# Types of LFO Problems

- Not all LFO algorithms work for all LFO problems
- Common differences:
  - Continuous/discreet variables
  - Observable environment or not
  - etc.

# Types of LFO Problems

- LFO problems can be characterized depending on whether:
  - They require generalization or not
  - They require planning or not
  - Do we have a model of the environment

# Types of LFO Problems

| Generalization? | Planning? | Known Env.? | Level |
|---|---|---|---|
| no | no | - | Level 1: Strict Imitation |
| yes | no | - | Level 2: Reactive Behavior |
| yes | yes | yes | Level 3: Tactical Behavior |
| yes | yes | no | Level 4: Tactical Behavior in unknown environment |

# Level 1: Strict Imitation

- No feedback required from environment
- No need for generalization nor planning
- The learned behavior is a strict function of time
- Algorithms required: pure memorization
- Example: robots in factories

# Level 2: Reactive Behavior

- Behavior is a ”perception to action mapping”
- No need for planning
- Standard (classification/regression) machine learning algorithms can be used in this level
- Example: simple complete information games like pong or space invaders

# Level 3: Tactical Behavior

- Perception is not enough to determine behavior:
  - Behavior to be learned has internal state
- Standard (classification/regression) machine learning algorithms cannot be used directly
- Example: driving a car, or complex games (e.g. Stratego)

# Outline

- Learning from Observation
- A Unified Framework
- Levels of Difficulty of LFO
- Statistical Formulation
- Conclusions

# Statistical Formulation of LFO

- Behavior as a stochastic process

$$I = \{I_1, ..., I_n\}$$

$$I_k = (X_k, Y_k)$$

- LFO consists on estimating the probability distribution of the stochastic process

$$\rho(Y_k | x_k, i_{k-1}, ..., i_1)$$

# Level 1: Strict Imitation

- Only the sequence of actions in the training trace has non 0 probability:

$$\rho(I_1 = (x_1, y_1), ..., I_n = (x_n, y_n)) = 1$$

$$BT = [(x_1, y_1), ..., (x_n, y_n)]$$

# Level 2: Reactive Behavior

- Reactive behavior only depends on perceptions:

$$\rho(Y_k|x_k, i_{k-1}, ..., i_1) = \rho(Y_k|x_k)$$

- In this case, LFO is equivalent to the traditional supervised learning problem, and each entry in a trace is one training example

# Level 3: Tactical Behavior

- The behavior needs some internal state (i.e. memory). Assuming only a finite amount of memory is required to learn a task:

$$\rho(Y_k|x_k, i_{k-1}, ..., i_1) = \rho(Y_k|x_k, i_{k-1}, ..., i_{k-l})$$

- Where $l$ plays a similar role as the order in a Markov process

# Level 3: Tactical Behavior

- Given a fixed $l$:
  - Markov process of order $l$ can be reduced to one of order $1$
  - We could use supervised learning algorithms
  - With an explosion in the set of input features

# Outline

- Learning from Observation
- A Unified Framework
- Levels of Difficulty of LFO
- Statistical Formulation
- Conclusions

# Conclusions

- Large amount of existing work in LFO
- Each author uses a different framework and vocabulary
- Need for unification for easy comparison of research and results

# Conclusions

- We presented a proposal for unified vocabulary
- Classification of LFO tasks in a series of levels:
  - Our goal was to classify the types of algorithms needed for different types of tasks

# Future Work

- Performance evaluation methodology
- Standard testbeds for comparison:
  - E.g. computer games?

Thank you!