Learning Physically-Instantiated Game Play Through Visual Observation

Andrei Barbu, Siddharth Narayanaswamy, and Jeffrey Mark Siskind

School of Electrical and Computer Engineering

SHANGHAI JIAO TONG UNIVERSITY

Introduction

- goal is to emulate a 2-year-old child
- an integrated robotics system for learning to play board games
- learn a full set of rules; learn to play, not to play well
- learn the initial board, legal-move generator, and outcome predicate
- two robots play a board game, while a third watches and takes over
- fully automatic with no human intervention
- no communication between the robots

Why games, and why learning?

- games are an idealized version of the real world
- most AI cannot deal with real-world complexity
- children learn from observation
- not only when rules are unavailable, c.f. Social Learning Theory
- have you read the rules for board games you've played?

Experimental setup

Robots

- custom robots with a 4-DOF arm, two fingers, and two eyes
  1. shoulder pan
  2. shoulder tilt
- eyes on a 1-DOF pendulum arm that rotates around the game
- each eye can pan/tilt independently
- mounted in a custom housing
- parts primarily from Lynxmotion, enhanced with custom parts to provide greater support for the arm and eyes and increased efficacy of operation

Computer Vision

- reconstruct the game state from visual information
- must detect the board itself; this is a calibration step done once on startup where 9 ellipses arranged in a grid are found
- OpenCV ellipse finder is used with multiple thresholds and voting in order to detect Xs, Os, and empty board positions

Games

- off-the-shelf game hardware, but judiciously chosen to simplify robotic manipulation
- depressions in the board provide for easy piece placement
- large, easy-to-grab pieces
- Tic-Tac-Toe with standard rules learned
- Hexapawn; three pawns on opposing sides; win by queening, capture, or force an inability to move
- learned 5 variants of Hexapawn: regular, forward diagonal moves, forward and backward diagonal moves, vertical backward moves, vertical backward and sideways moves

Task

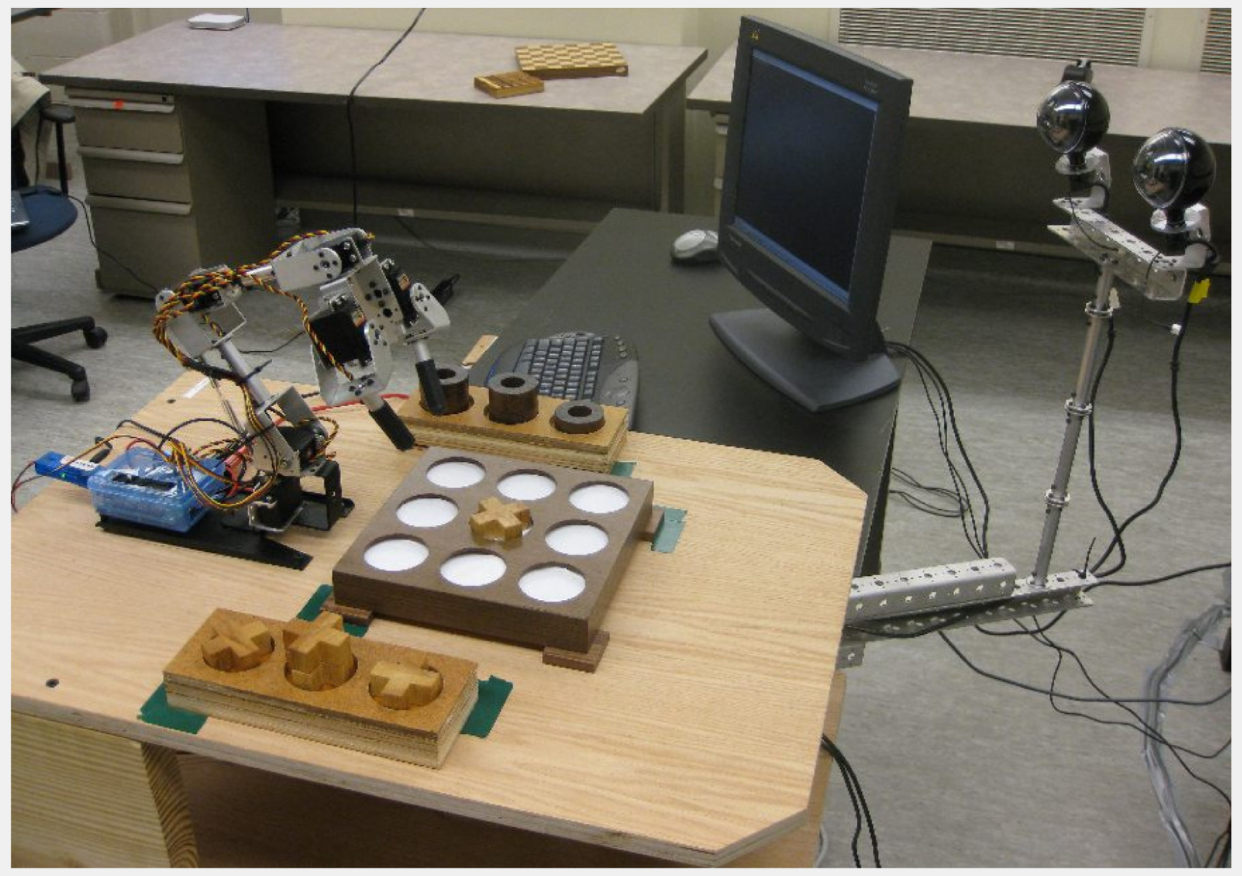

Figure: figure example

Rules

- PROGOL, an inductive-logic-programming (ILP) system, is used to learn the initial board, legal-move generator, and outcome predicate
- rules represented as logical formulae, i.e. Horn clauses
- learned rules are then directly executed with PROLOG
- system is given background knowledge about the world, such as: a board exists, pieces can be on the board, players own pieces, the concepts of linearity, forwards, sideways, and the frame axiom
- background knowledge is of the type any child would have
- search the space of possible initial board, legal-move generators, and outcome predicates for 3×3 games, given the evidence of n games, and find the most compressed rule set which best explains the observed games
- learn the initial board first, then the legal-move generator, and finally the outcome predicate
- use the previously acquired knowledge to learn the next item
- can learn a full game description in a modest number of games, typically 3–6

$$E = mc^2$$

Results

- reliable robust operation was achieved for 62 games, approximately 2000 pick-up and put-down operations with fewer than 20 interventions
- robotic manipulation based on dead-reckoning due to non-linear relationship between servo control and angular position
- error detection by interleaving manipulation and visual reconstruction of board states
- learned Hexapawn rule set:

```
number = 4
result = number * number
print(result)  # Outputs: 16
```

- learned similar rule sets for all 6 games

Lincoln Logs & language

- assembly task using assembly toys, e.g. Lincoln Logs
- novel computer-vision system to recognize block assemblies from grammars by extracting features, fitting them to known grammars, and searching for implied necessary or possible blocks
- novel language component to describe assemblies in terms of walls and windows, and reconstruct the same structure out of different assembly toys
- more advanced robotic system, with custom grippers, farther reach, tactile sensors, and a palm-mounted camera
- more robust robotic manipulation using visual servoing

Future work

- complete the Lincoln Log task and move on to other assembly toys
- expand upon the current game learning and scale up to games of higher complexity, e.g. checkers
- learn the mapping from world state to game state
- learn Lincoln Log and other assembly-toy grammars
- integrate more sensors, e.g. a laser pointer and ultrasonic range finder
- stochastic ILP for fault tolerance
- a custom ILP system with better heuristics for learning games