Solving nonogram puzzles by reinforcement learning

Frédéric Dandurand†, Denis Cousineau‡, and Thomas R. Shultz*

† Department of Psychology, Université de Montréal, 90, ave. Vincent-d’Indy, Montréal, QC, H2V 2S9, Canada
‡ École de psychologie, Pavillon Vanier, Université d’Ottawa, 136 Jean Jacques Lussier, Ottawa, Ontario, K1N 6N5, Canada
*Department of Psychology and School of Computer Science, McGill University, 1205 Penfield Avenue, Montreal, QC, H3A 1B1, Canada

Abstract

We study solvers of nonogram puzzles. Given an optimal solving module for solving a given line, we compare performance of three algorithmic solvers used to select the order in which to solve lines with reinforcement learning. The reinforcement-learning (RL) solver uses a measure of reduction of distance to goal as a reward. We compare two methods for storing qualities (Q values) of state-action pairs, a lookup table and a connectionist function approximator. We find that RL solvers learn near-optimal solutions that also outperform a heuristic solver based on explicit, general rules often given to nonogram players. Only RL solvers that use a connectionist function approximator generalize their knowledge to generate good solutions on about half of unseen problems; RL solvers based on lookup tables do not generalize.

Methods

1. Surveyed online nonogram web sites
2. Found two classes of advice
   • Solving a line: many explicit strategies and rules (e.g., Wikipedia, under Nonogram)
   • Selecting which line to solve: heuristic advice given for solved examples; few explicit rules

Computational model

Hybrid system (explicit and implicit)

To solve a line: rule-based solver
To select lines: compare 4 solvers: Random, Heuristic, Optimal and Reinforcement learning

Reinforcement-learning solver

- Learns expected value (Q) of selecting this line (a_i) in its present state (s_i)
- Higher reward (r_{t+2}) for fewer steps

| Q(s_{t+1},a_t) = Q(s_{t+1},a_t) + a [ r_{t+1} + y \cdot Q(s_{t+2},a_{t+1}) - Q(s_{t+1},a_t)] |

- Two methods for storing/computing Q
  (1) Lookup table
  (2) Cascade-correlation neural network function approximator

- Inputs: concatenation of a line’s current state and its constraints
- Output: predicted (estimated) Q
- Training: 3 nonograms puzzles
- Testing: an unseen nonogram

Future work

- Collect human data for cognitive modeling
- Develop a universal solver which works on any nonogram puzzle, of any size

References


Contact: frederic.dandurand@gmail.com

Research supported by a grant and a fellowship from the Natural Sciences and Engineering Research Council of Canada.