Reinforcement Learning in Board Games

George Tucker
Paper Background

- Reinforcement learning in board games
  - Imran Ghory
  - 2004
- Surveys progress in last decade
- Suggests improvements
- Formalizes key game properties
- Develops a TD-learning game system
Why board games?

- Regarded as a sign of intelligence and learning
  - Chess
- Games as simplified models
  - Battleship
- Existing methods of comparison
  - Rating systems
What is reinforcement learning?

- After a sequence of actions get a reward
  - Positive or negative
- Temporal credit assignment problem
  - Determine credit for the reward
  - Temporal Difference Methods
    - TD-lambda
History

- Basics developed by Arthur Samuel
  - Checkers
- Richard Sutton introduced TD-lambda
- Gerald Tesauro creates TD-Gammon
- Chess and Go
  - Worse than conventional AI
History

- Othello
  - Contradictory results
- Substantial growth since then
- TD-lambda has potential to learn game variants
Conventional Strategies

- Most methods use an evaluation function
- Use minimax/alpha-beta search
- Hand-designed feature detectors
  - Evaluation function is a weighted sum

- So why TD learning?
  - Does not need hand coded features
  - Generalization
Temporal Difference Learning

$$Output = \sum_{k=1}^{H} f\left(\sum_{j=1}^{N} I_{j,k} W_{j}^{I}\right) W_{k}^{O}$$

- $N$ is the number of input nodes.
- $H$ is the number of hidden nodes.
- $f()$ is our non-linear function.
\[
\Delta W_t = \alpha \sum_{k=1}^{t} \lambda^{t-k} \nabla_w Y_k d_t
\]

- \(t\) is time (in our case move number).
- \(T\) is the final time (total number of moves).
- \(Y_t\) is the evaluation of the board at time \(t\) when \(t \neq T\).
- \(Y_T\) is the true reward (i.e. win, loss or draw).
- \(\alpha\) is the learning rate.
- \(\nabla_w Y_k\) is the partial derivative of the weights with respect to the output.
- \(d_T\) is the temporal difference.
Disadvantage

- Requires lots of training
- Self-play
  - Short-term pathologies
  - Randomization
TD Algorithm Variants

- **TD-Leaf**
  - Evaluation function search
- **TD-Directed**
  - Minimax search
- **TD-Mu**
  - Fixed opponent
  - Use evaluation function on opponent’s moves
Current State

- **Many improvements**
  - Sparse and dubious validation
  - Hard to check

- **Tuning weights**
  - Nonlinear combinations
  - Differentiate between effective and ineffective

- **Automated evolution method of feature generation**
  - Turian
Important Game Properties

- **Board Smoothness**
  - Capabilities tied to smoothness
  - Based on the board representation

- **Divergence rate**
  - Measure how a single move changes the board
  - Backgammon and Chess – low to medium
  - Othello – high

- **Forced exploration**

- **State space complexity**
  - Longer training
  - Possibly the most important factor
Importance of State space complexity
Training Data

- Random play
  - Limited use
- Fixed opponent
  - Game environment and opponent are one
- Database play
  - Speed
- Self-play
  - No outside sources for data
  - Slow
  - Learns what works
- Hybrid methods
Improvement: General

- Reward size
  - Fixed value
  - Based on end board
- Board encoding
- When to learn?
  - Every move?
  - Random moves?
- Repetitive learning
- Board inversion
- Batch learning
Improvement: Neural Network

- Functions in Neural Network
  - Radial Basis Functions
- Training algorithm
  - RPROP
- Random weight initialization
  - Significance
Improvement: Self-play

- **Asymmetry**
  - Game-tree + function approximator

- **Player handling**
  - Tesauro adds an extra unit
  - Negate score (zero-sum game)
  - Reverse colors

- **Random moves**
  - Algorithm

- **Informed final board evaluation**
Evaluation

- Tic-tac-toe and Connect 4
  - Amenable to TD-learning
  - Human board encoding is near optimal
- Networks across multiple games
  - A general game player
    - Plays perfectly near end game
    - Randomly otherwise
  - Random-decay handicap
    - % of moves are random
    - Common system
Random Initializations

- Significant impact on learning
Inverted Board

- Speeds up initial training
Random Move Selection

- More sophisticated techniques are required
Reversed Color Evaluation
Batch Learning

- Similar to control

![Graph showing percentage of games won over 100's of training games]
Repetitive learning

- No advantage
Informed Final Board Evaluation

- Extremely significant

![Graph showing percentage of games won over hundreds of training games with two lines representing different conditions: Control and No-Final Guidance Agent.](image)
Conclusion

- Inverted boards and reverse color evaluation
- Initialization is important
- Biased randomization techniques
- Batch learning has promise
- Informed final board evaluation is important