Feature Ensemble Method for the Game of Go

FASTAR/Espresso Workshop 2012

Francois van Niekerk
francoisvn@ml.sun.ac.za

October 2012

Outline

1 Introduction

2 Background
- Game of Go
- Monte-Carlo Tree Search for Computer Go
- Patterns in Go

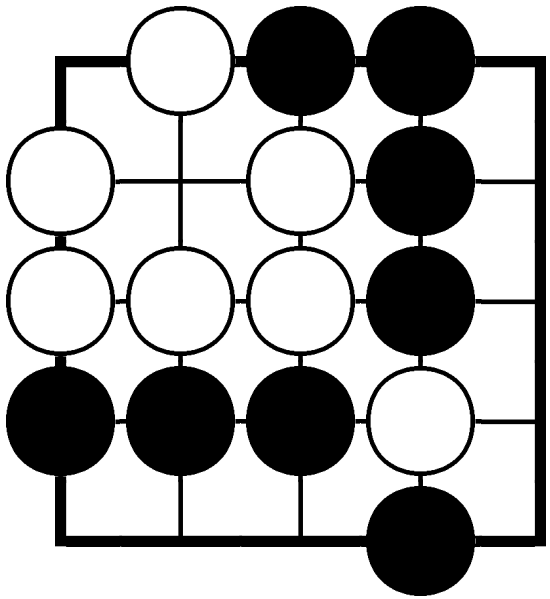
3 Feature Ensemble Method
- Process
- Decision Trees
- Pattern Methods

4 Conclusion

Introduction

- Computer Go: development of computer programs able to play Go.
- Monte-Carlo Tree Search (MCTS) is the dominant Computer Go algorithm.
- MCTS doesn't learn — no improvement from playing many games.

Game of Go



- Ancient combinatorial board game.
- Simple rules.
- Very deep tactically and strategically.

Monte-Carlo Tree Search for Computer Go

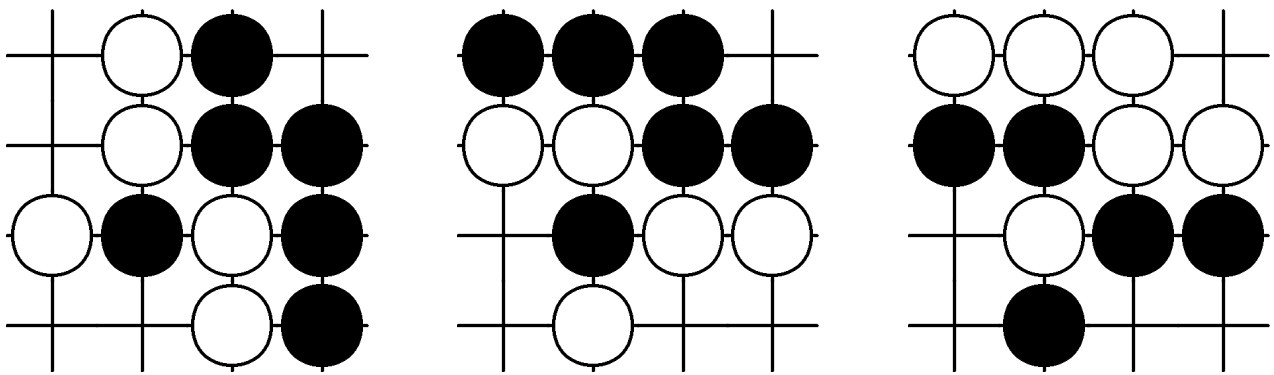
- Tree for moves and their follow-ups.
- Exponential tree growth means brute-force is infeasible.
- Monte-Carlo methods — stochastic simulations (playouts).
- Winrate of playouts starting from a position is the value of the position.
- Playout results guide tree exploration yielding Monte-Carlo Tree Search (MCTS).
- Various heuristics used to bias the search and playouts.

Patterns in Go: Use in MCTS

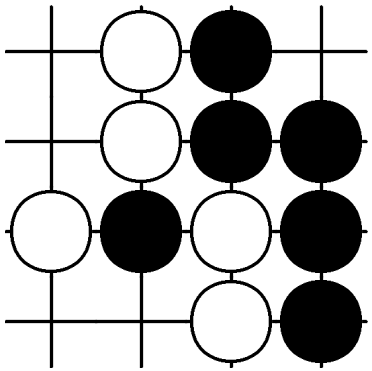
- Commonly occurring patterns are harvested from collections of top-level games.
- Patterns as well as other simple metrics (e.g. atari, distance to border) are used as features.
- Weights are trained for these features.
- Feature weights are combined to form a move weight.
- These weights are used to bias MCTS.

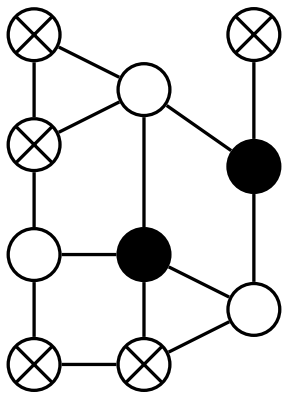
Patterns in Go: Naïve Patterns

- Patterns of black, white and empty points in an area, around a potential move.
- Patterns must account for changes in colour, rotation and reflection.
- Simple to use.
- Functionally equivalent patterns often don't match.



Patterns in Go: Common Fate Graphs





- Board intersections form nodes in a graph.
- Adjacent non-empty nodes of the same colour are merged.
- More concise than naïve patterns.
- More difficult to store and match.
- Current use is limited.
- Other graph pattern representations exist.

Feature Ensemble Method

- Patterns are very important in Go.
- Our aim: discover valuable patterns which cannot be represented by naïve patterns.
- We propose pattern methods generating a graph centered around the evaluated move.
- Feature Ensemble Method (FEM) uses many pattern methods (features) in combination.
- Incremental decision trees used to explore potential patterns.

Process

- Offline (before and after play):
  - Patterns are harvested from games (seeded with top-level games).
  - Decision trees are grown with new patterns.
  - Weights for patterns are trained incrementally.
- Online (during play):
  - Decision trees are used to match patterns.
  - Weights of matched patterns are combined to form a move weight.
  - Move weights are used to bias the tree search or playouts.

Decision Trees

- One incremental decision tree per pattern method.
- Decision trees used to guide graph exploration for the pattern methods.
- Each internal node has a question that partitions the patterns.
- Each decision tree node represents a pattern.
- As we descend down the tree, patterns increase in size and specificity.
- Tree is grown by gathering statistics and considering different questions.
- Questions aren't permanently fixed — these decision trees can also shrink to change questions (incremental decision trees).

Pattern Methods

- Pattern method groups with parameters to differentiate.
- Naïve-CFG:
  - Similar to naïve patterns and CFG.
  - Parameters: Compress stones? Compress empty areas? Partially compress some areas?
  - Nodes store number of intersections compressed.
  - Edges store number of connections.
  - Most relevant: middle and end of game.
- Sparse:
  - Store the closest stones.
  - Parameters: Compress stone chains?
  - Edges store min and max distances between nodes.
  - Most relevant: opening and start of middle of game.
- Region-Compress:
  - Compress regions of stones and empty intersections.
  - Maintains information of shape and orientation.
  - Most relevant: local tactics and end of game.

Conclusion

- Our aim: discover valuable patterns which cannot be represented by naïve patterns.
- We proposed Feature Ensemble Method (FEM) in this talk.
- FEM uses many pattern methods in combination.
- Incremental decision trees used to explore potential patterns.
- Uncertain how best to structure questions in decision trees so that they are able to efficiently match arbitrary graph portions, while staying invariant to rotation and reflection.

Thanks

Thank you for your time.

Any questions or suggestions?

Email: francoisvn@ml.sun.ac.za.