Search Pathology in the Game of Go

Markus Enzenberger

Go Seminar, University of Alberta

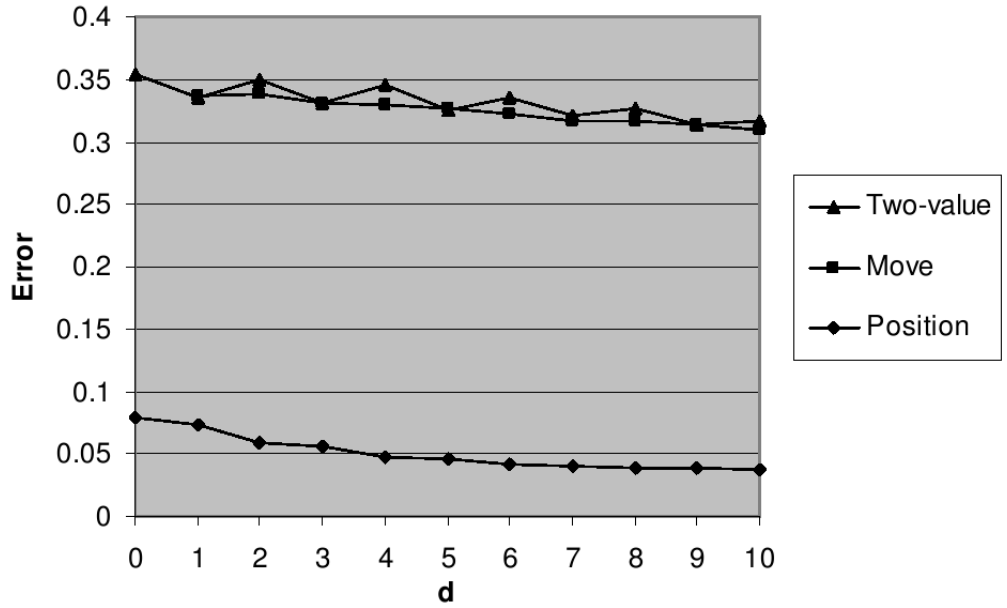
Januar 31, 2006

Contents

Search Pathology

- Minimax search in game trees can degrade quality of the backed-up heuristic evaluation (Nau, 1979; Beal, 1980)
- Usually not observed in real game-playing programs
- Absence was explained by
  - Similarity of values in sibling nodes (Beal, 1982; Bratko, 1982)
  - Early terminal positions (Pearl, 1984)
  - Optimistic evaluation function (Schrüfer, 1986)

Recent paper (Luštrek, Bratko, Gams, 2005):



- Disappears in model with real-valued evaluation function
- Win/loss error shows an odd-even effect

An odd-even pathology was also observed by (Nau, 1986) using minimax for a certain class of games

- Studies use generated noisy evaluation function applied to small artificial game trees.
- It can be expected that some games are more susceptible to pathology than others
- Experimental data by real game-playing programs rare

Computer Go

- Misjudging status of a group introduces error, which can both over- and underestimate real value by a large amount
- Many Go programs do not use global search at all
- Go game trees have a nearly uniform branching factor

NeuroGo

- Neural network based
- Learns by self-play
- Uses TD-learning for prediction of local connectivity and local reward
- Tournament version uses an extended set of input features
- Nega-max implementation of alpha-beta search
  Standard move ordering techniques; iterative deepening
- All legal moves are generated, apart from safe territory (threshold)

Goal

- Examine statistical properties of NeuroGo's evaluation function
- Study performance depending on search depth

Error of evaluation function is unknown,
but higher error will cause fewer wins

Game Playing

- Opponent programs supporting Go Text Protocol
- 9×9 board
- Komi 6.5
- 21 balanced four-move opening positions
- 200 games against each opponent / for each depth 1–4
- Alternate color every 2nd game
- Exclude duplicates
- Use GNU Go for scoring
- Athlon XP 2800+/512 MB (≈300 evaluations/sec)

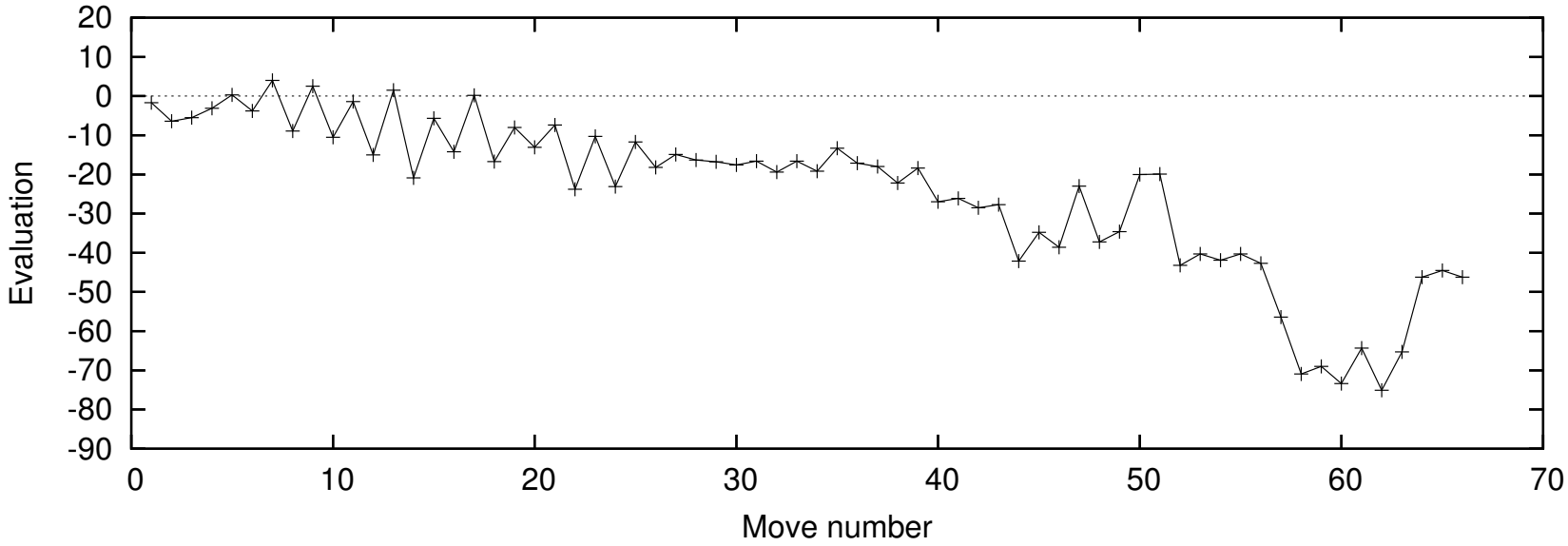
Opponent Programs

- Aya 5.53, uses selective global search on 9×9
- Crazy Stone 0001-19, Monte Carlo
- Explorer, 6.6.x/Nov 3 2005, no global search
- GNU Go 3.6, no global search
- NeuroGo (depth 1)

Statistical Properties

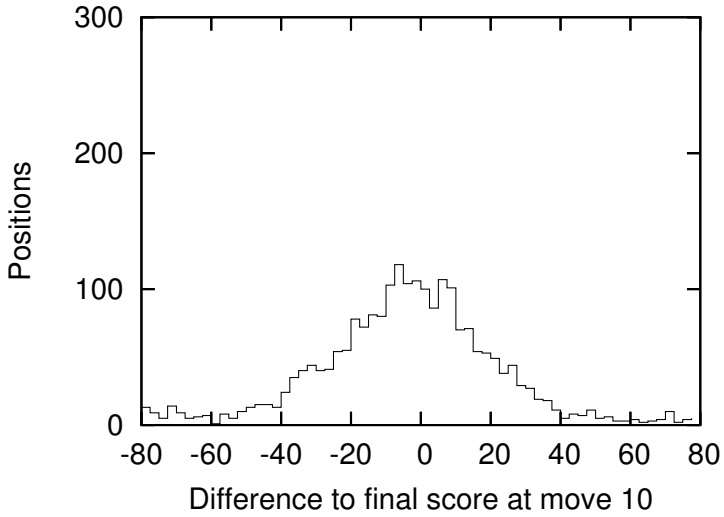
- All positions in games
- Black counts positive
- Black odd / White even move numbers
- NeuroGo automatically adds komi

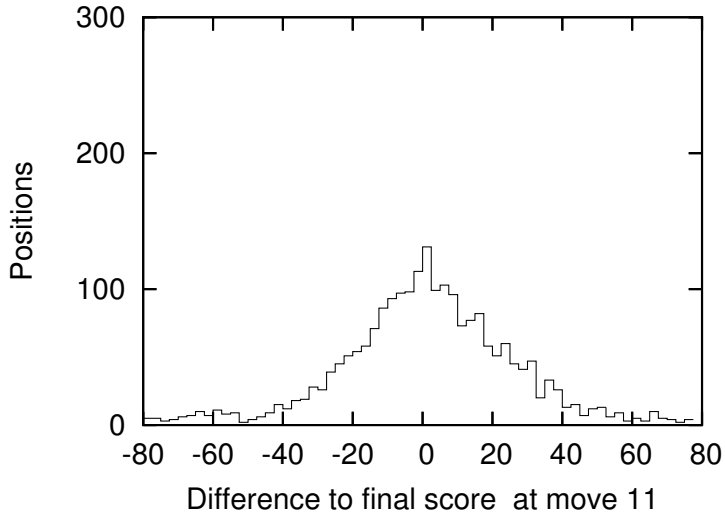
Example Game (NeuroGo vs Aya)

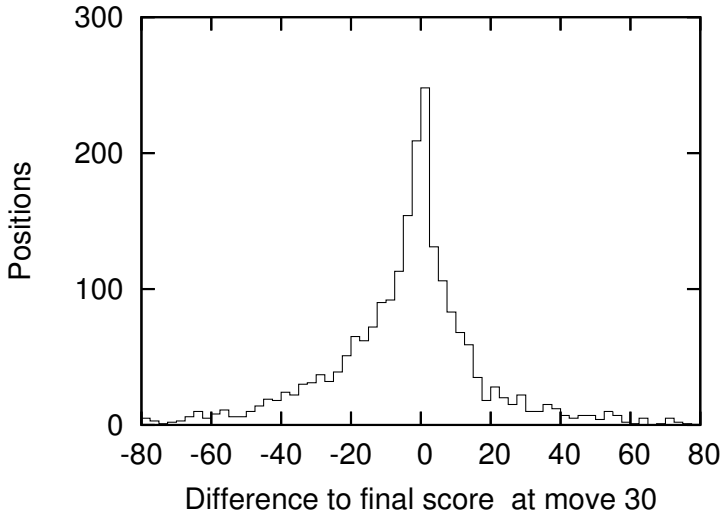


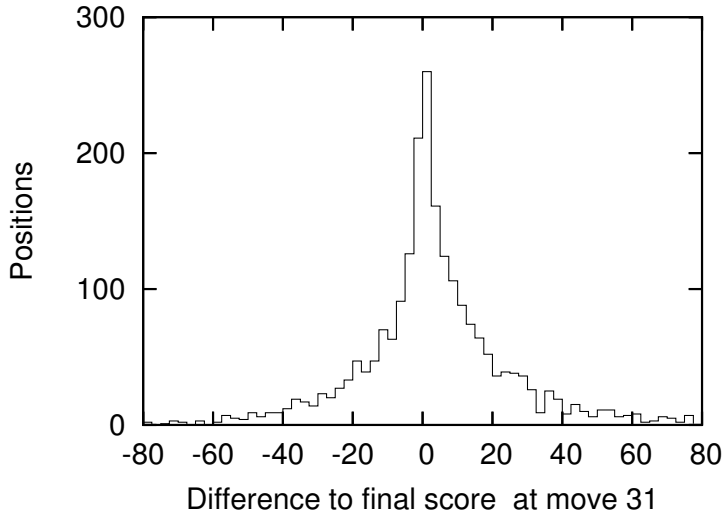
- Stable periods
- Slow changes
- Rapid changes
- Oscillations

Difference to final score





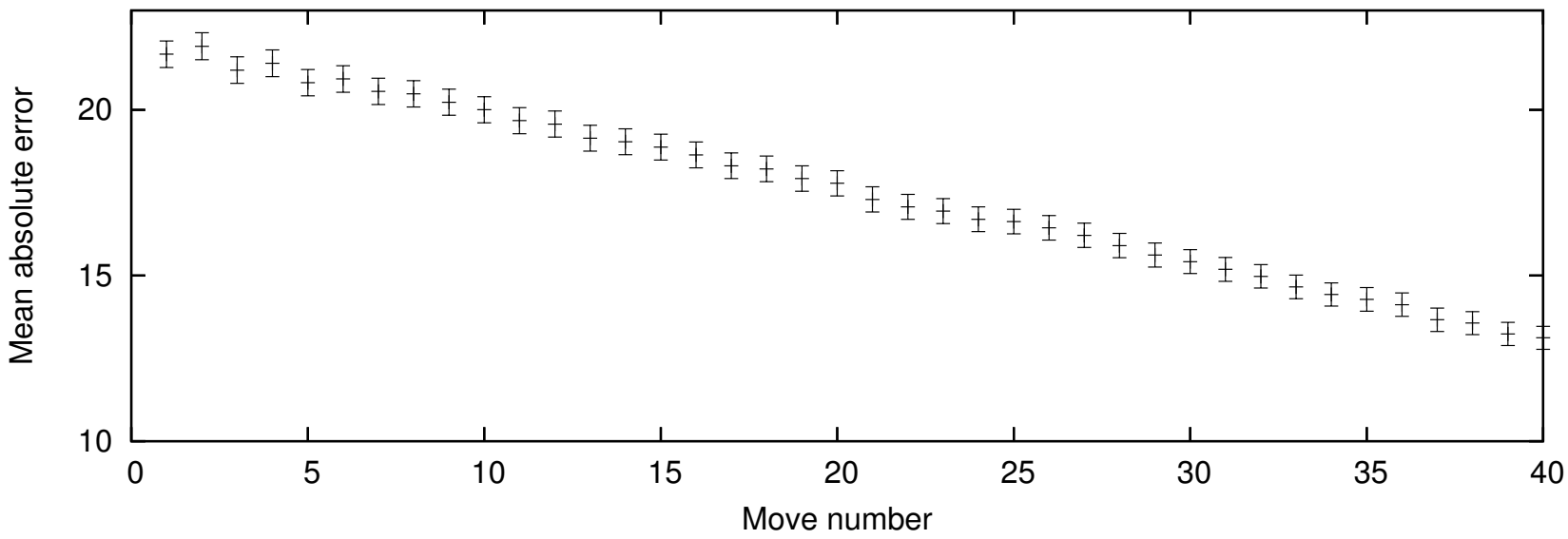




Difference to final score

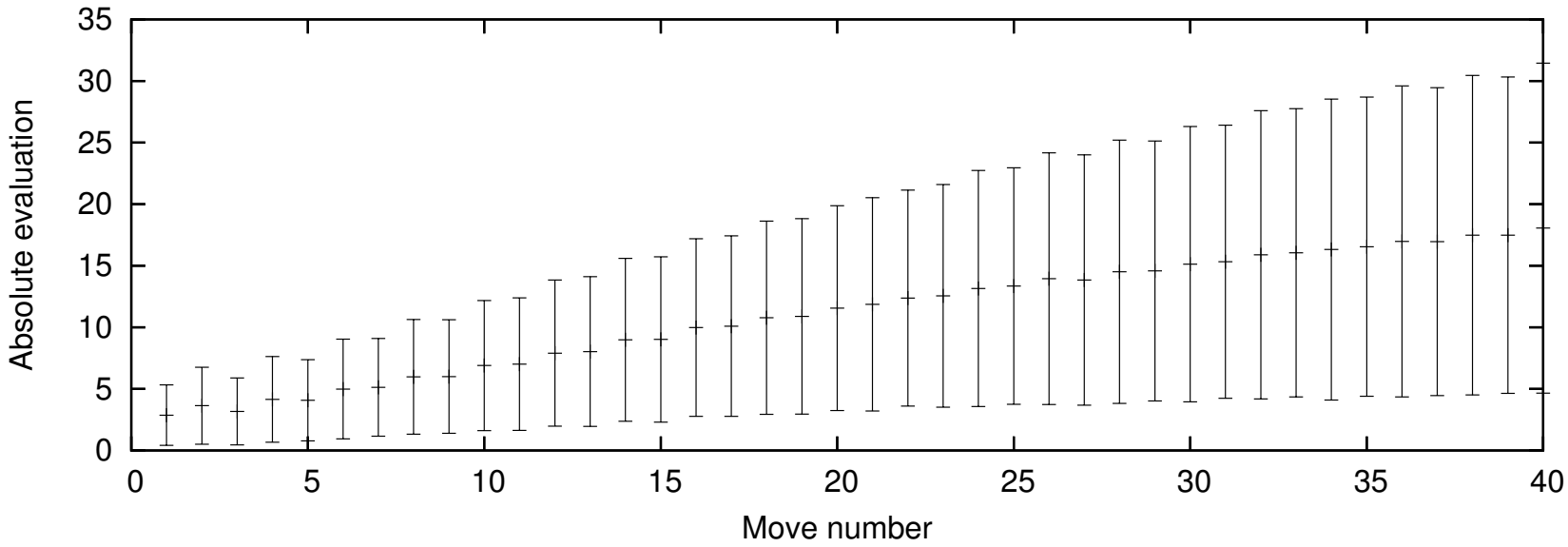
- Approximation of the error in evaluation
- Gaussian only early in game
- Peak around zero increases with move number
- Large tails
- Similar, but slight asymmetry for Black/White to move

Mean absolute difference to final score



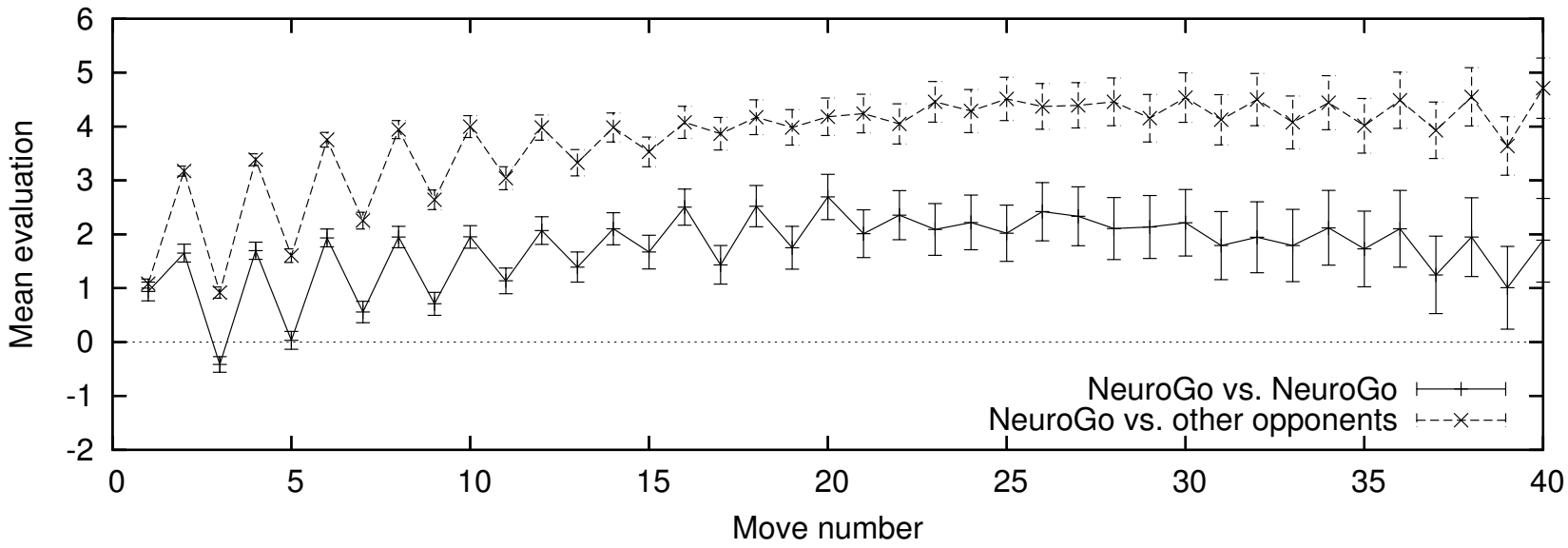
- Decreases nearly linearly
- Apart from very early positions, no dependency on color to move

Mean absolute evaluation (with deviation)



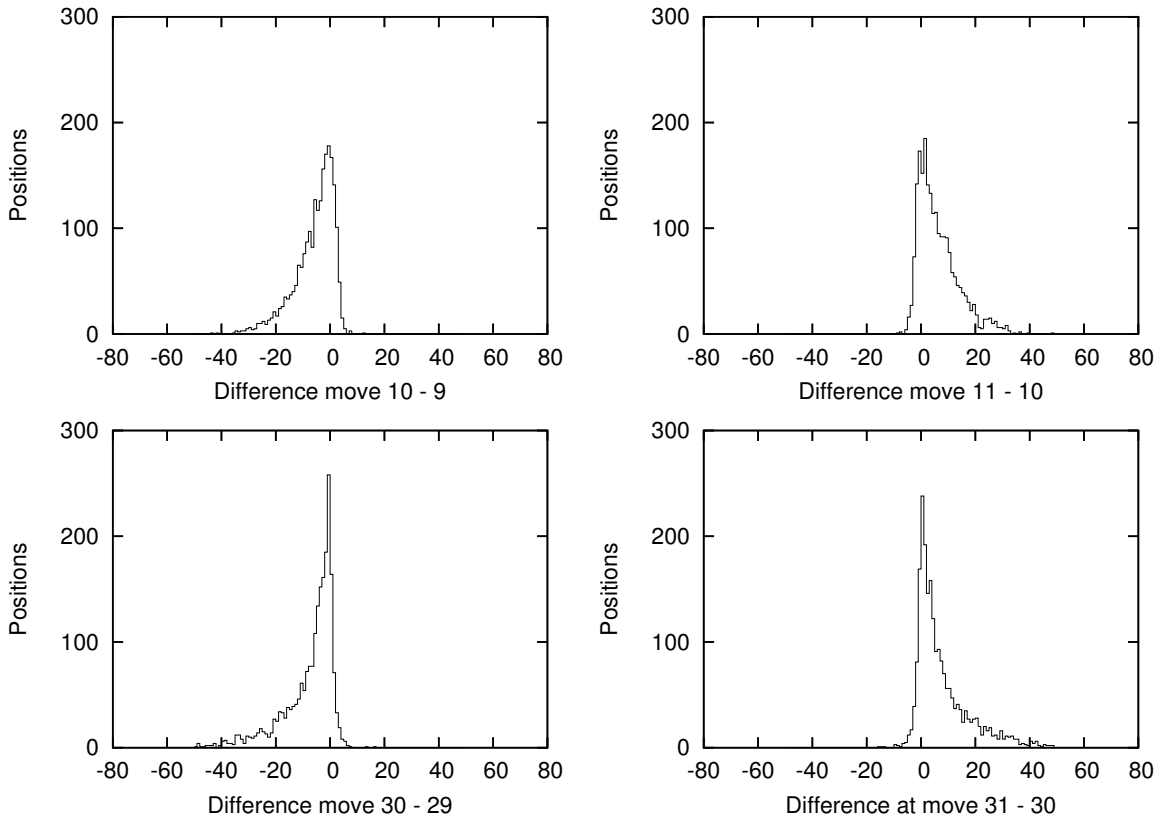
- Increases nearly linearly after move 5
- Smaller slope after move 20
- Variance increasing with move number

Average evaluation for player to move



- On average optimistic for player to move (especially in middle game)
- Favours positions with White to move (komi over-compensates)

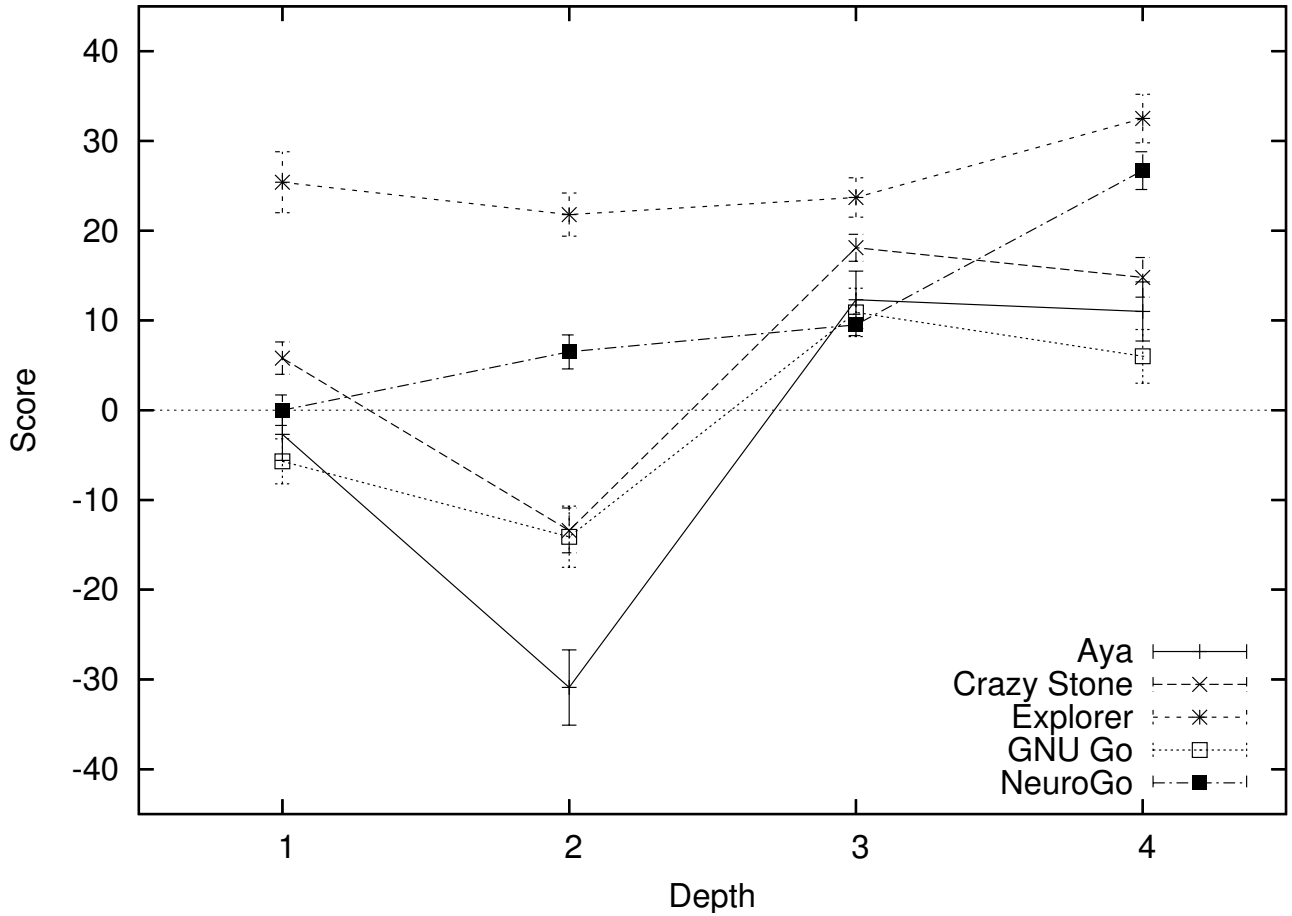
Difference between subsequent moves



- Large number of positions with stable evaluation
- If unstable, greatly favours player to move

Performance

Average score



- Results vs Aya, Crazy Stone, GNU Go show a pathology
- Depth 2 worse than 1; 4 worse than 3
- Not visible in games against Explorer
- Absent in games against itself

Conclusion

- Evaluation function of a medium-strength Go program was examined
- Evaluation was found optimistic for player to move, especially in unstable positions
- Difference to final score decreased linearly with move number
- Resembles Gaussian only for early positions
- Search pathology occurred
- Significant impact on playing strength at low depths
- Games vs fixed-depth version of same Go program are not sufficient to observe this effect