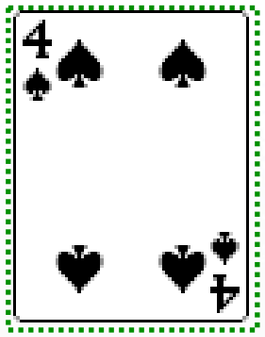
Learning to Play Hearts

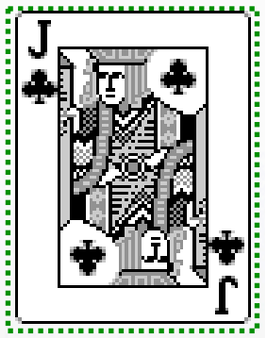
Nathan Sturtevant
University of Alberta - AI Seminar
February 3, 2006

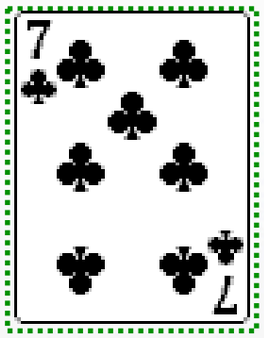
Hearts

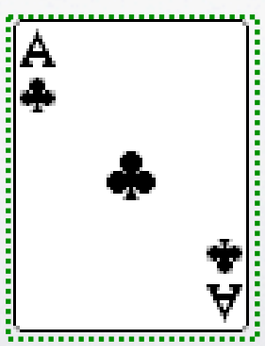
- Trick-based card game

Sample Trick

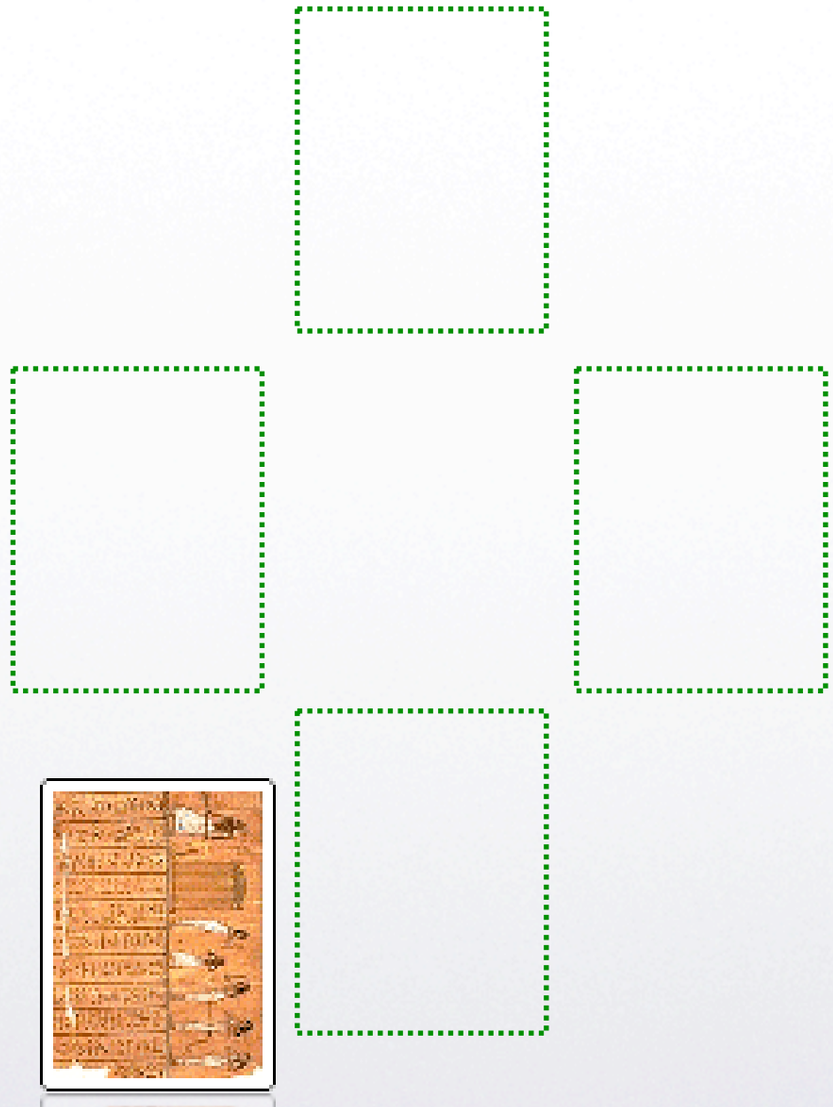








Sample Trick



Hearts

- Trick-based card game
- Want to *minimize* your points
  - One point for every heart (♥)
  - 13 points for Q♠
  - If one player takes all 26 points

    (shoots the moon) others get 26 each

Challenge

- Learn to play the game of Hearts well:
  - Multi-Player Game
  - Imperfect Information
  - Learning

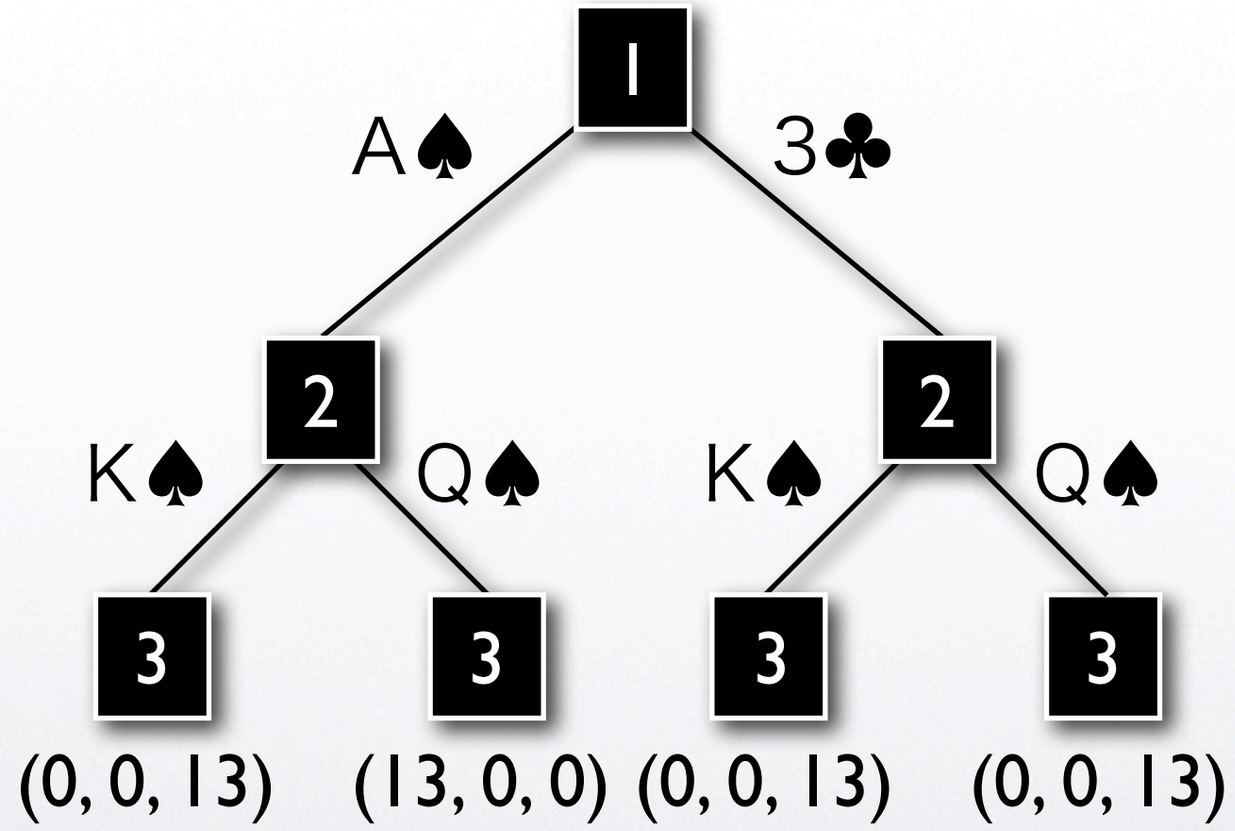
Multi-Player Games

- A lot of work in two-player games:
  - Checkers, chess, backgammon, scrabble, othello, go…
- Much less in multi-player games

Multi-Player Games

- Differences:
  - Max^n algorithm; generalization of minimax
  - Less pruning possible
  - Weaker theoretical properties

Hearts Example



Imperfect Information

- In practice we can't see opponents cards
- Monte-Carlo Sampling
  - Generate perfect-information sample hands for opponents
  - Analyze samples
  - Combine results

Previous Work

- Hearts program based on previous ideas
  - Hand-tuned evaluation function
  - Non-linear evaluation function
  - Somewhat slow
- Plays as well (better than) best computers?


3D Hearts Deluxe 7.5 0506
2 3 4 9 5 6 A 2 A 3 6 K A
Your lead...

Average Scores

	Per Game	Per Hand
Expert Program	56.1	5.16
Opponent Avg.	76.3	6.97

Played 90 games, each to 100 points.

Learning in Hearts

- University of Mass. Course Project (Perkins, 1998)
- Operational Advice (Fürnkranz, et. al., 2000)
- State sampling with imperfect-information (Fujita and Ishii, 2005)

Our Approach

- Use search-based approach to train
  - Similar to what was used in Backgammon
  - TD-Gammon plays at the level of the best humans

Backgammon

- Why did learning in backgammon work well?
  - Already developed good neural networks
  - Stochastic element helps exploration
  - Good features

Hearts

- Promising domain for learning:
  - Game fixed length (13 moves)
  - Cards dealt randomly
  - Occasionally get good cards

Hearts Difficulty

- Cards have relative value
  - 5♣ is good when 2-4♣ already played
  - 5♣ is bad when 6-A♣ already played

Approach

- Define features for game
- Use perceptron (regression) to predict game score given features
- Use max^n to play given predicted score
- Use TD(λ) to train

Features

- What features to use for each player?
  - 52 cards they could have in their hand
  - 52 cards they could have taken
  - 104 features per player
  - 416 total features

Valuable Feature

- Interesting feature: P1 has the lowest ♥
  - [P1 has 2♥] or
  - [P1 has 3♥] and

    [[P1 has taken 2♥] or [P2 has taken 2♥]
    [[P3 has taken 2♥] or [P4 has taken 2♥]]
  - …

Features

- We defined basic ‘atomic’ features
- Only evaluate features on trick boundaries
- Sample Features
  - Which suits do we hold low/high cards
  - Which suits are we ‘short’
  - Which suits does the ‘leader’ have

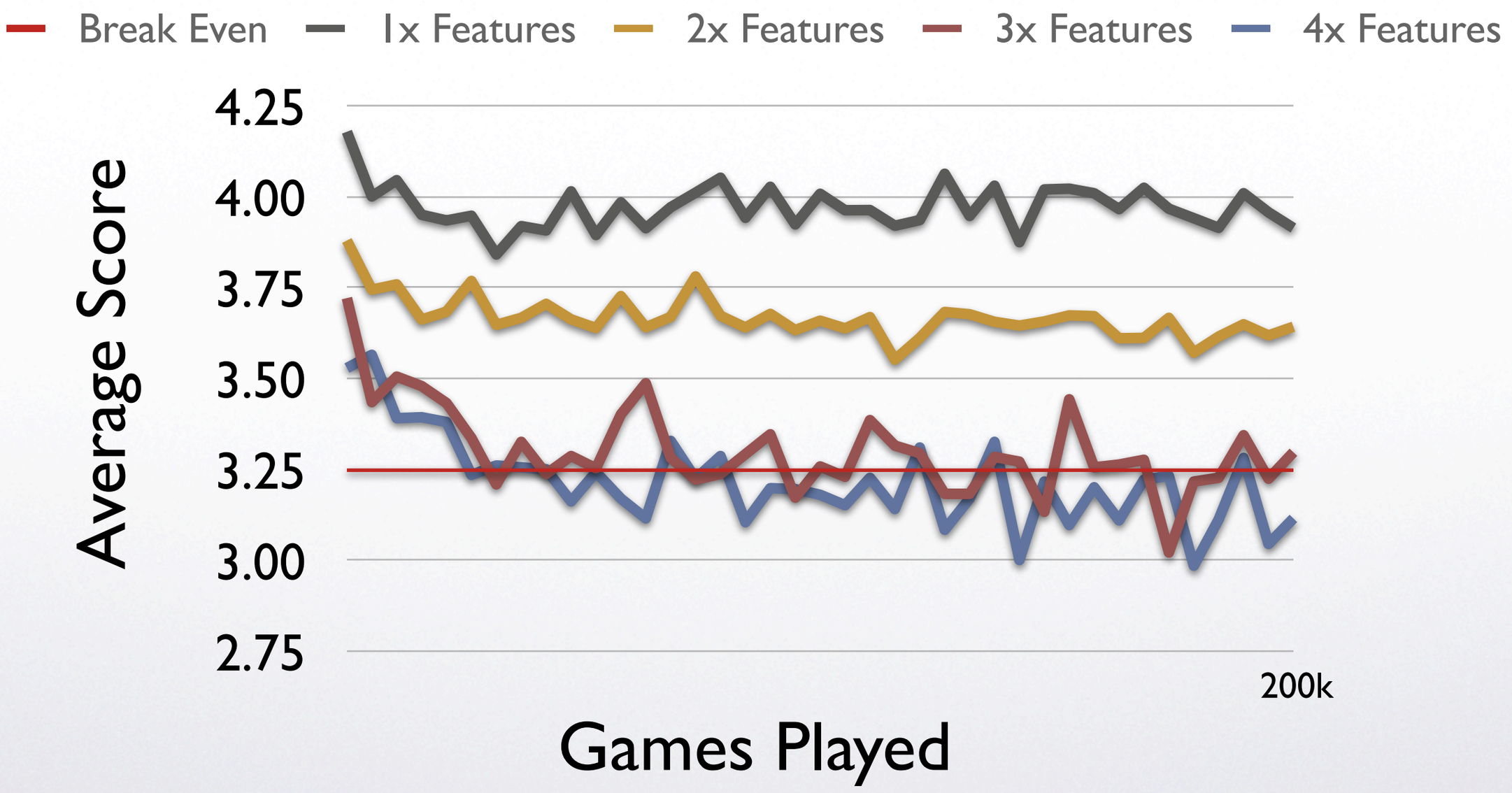
Features

- These features still inadequate
  - Combinations of features more interesting than ‘atomic’ features
  - Combine features using AND operator

Learning Part I

- Learn to avoid the Q♠
  - 60 'atomic features'
  - λ = 0.75
  - α = 1/[13 x # active features]
  - Predict expected points in game
  - Train against previous program
  - Randomly switch position each game

QoS Features



Analysis

- What is the network learning
- Easily understand by examining weights assigned to feature sets

Features - Avoid Q♠

Rank	Weight	We have	We have	We have	Opponent
1	-0.103	1 low ♠		Lead	Q♠ no ♠
2	-0.097	1 low ♠	No ♥	Lead	Q♠ no ♠
3	-0.096	2 low ♠	K♠		Q♠ two ♠
4	-0.093	1 low ♠	No ♣	Lead	Q♠ no ♠
5	-0.090	1 low ♠	No ♦	Lead	Q♠ no ♠
148	-0.040	1 low ♠	Q♠		Lead no ♠

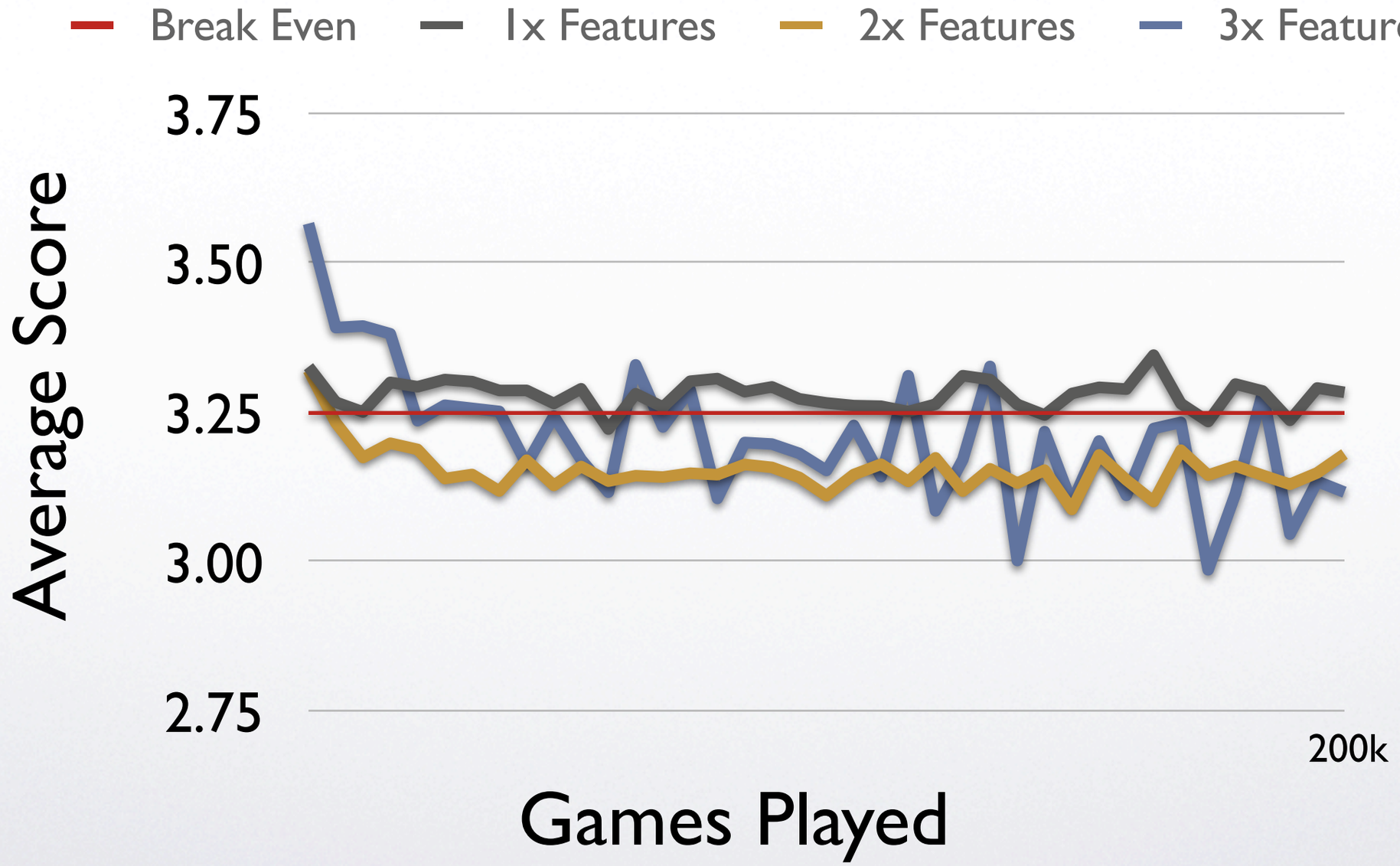
Features - Take Q♠

Rank	Weight	We Have	We have	We have	We have
1	0.125	Q♠	1 low ♠		Lead
2	0.123	Q♠	1 low ♠		
3	0.117	Q♠	No ♣	No ♥	Lead
4	0.116	A/K/Q♠			Lead
5	0.112	Q♠	No ♣	No ♥	No ♦

Learning Part II

- Learn to avoid taking ♥
  - Removed 14 Q♠-specific features
  - 42 new point (♥) related features (0-13)
  - Same learning parameters

Hearts Features



Learning Part III

- Learn to play the full game of Hearts
  - No ‘shooting the moon’
  - Take best 10,000 features from the Q♠
  - Take best 1,000 features from ♥ points
  - Train against expert and by self-play

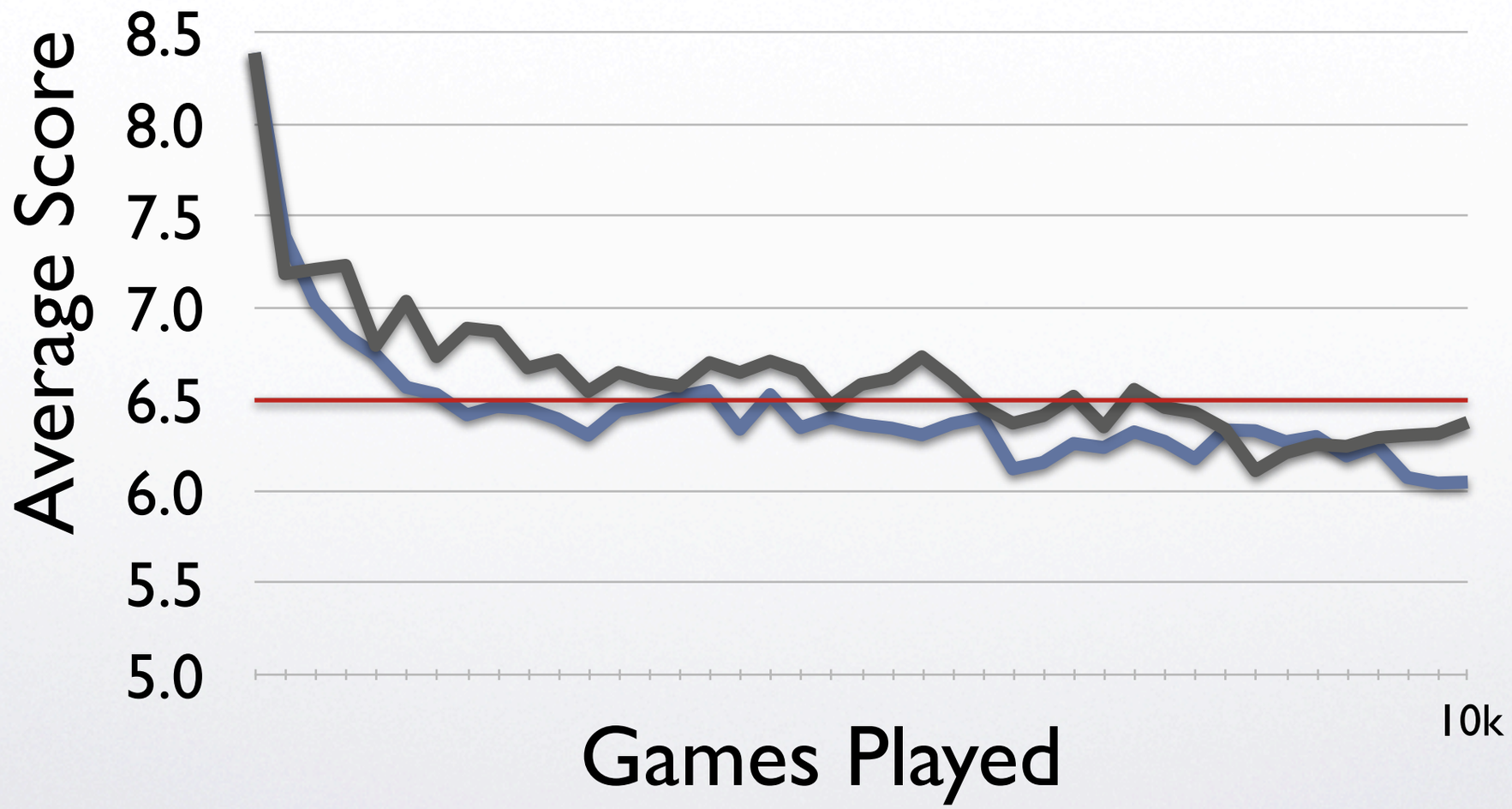
Steady-State Evaluation

- Test the learned networks
  - 4 players, 2 player types
  - 2^4 ways of assigning player types
  - Repeat each hand 2^4 - 2 times
  - Play 100 hands

Arrangement

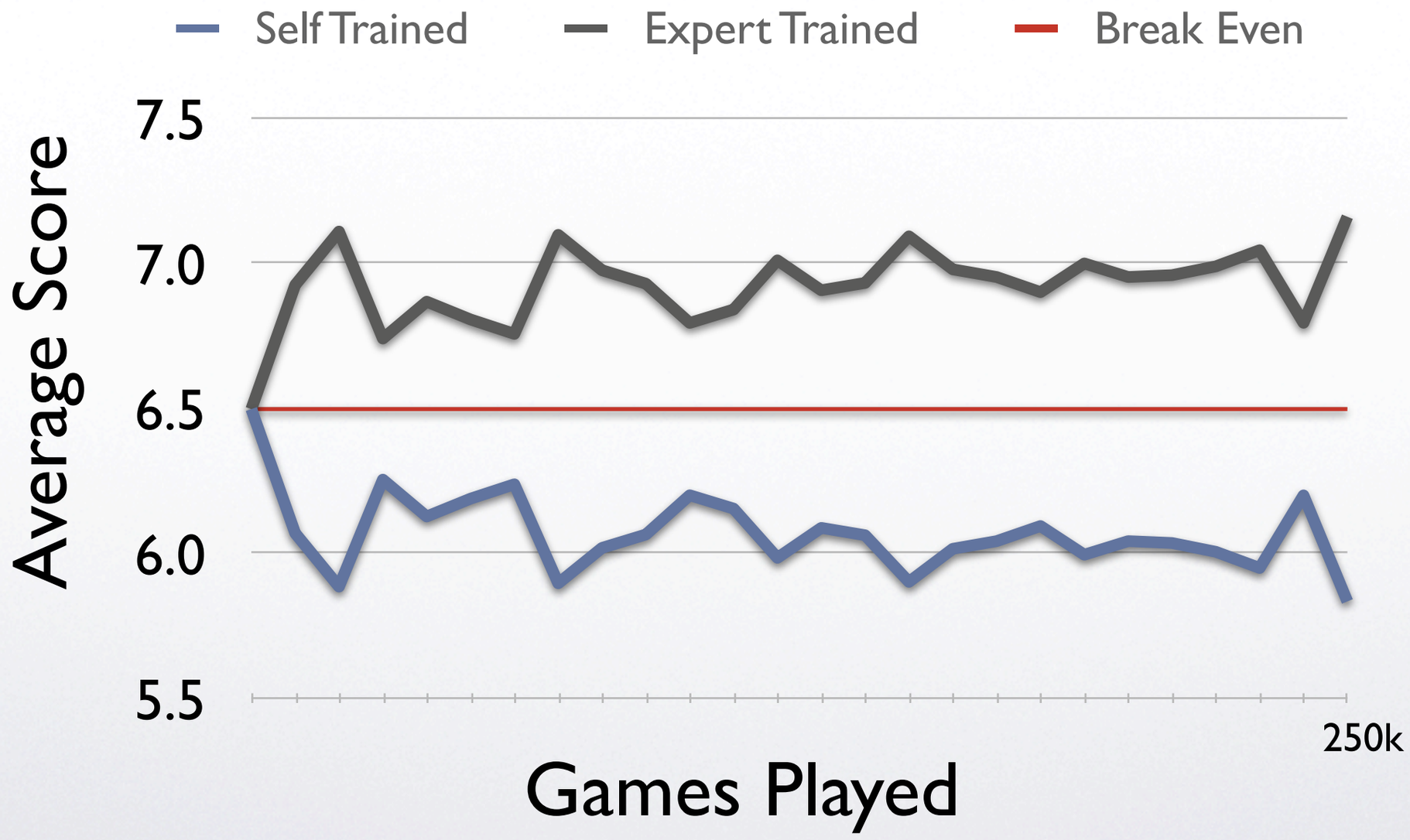
Player 1	Player 2	Player 3	Player 4
Expert	Trained	Trained	Trained
Trained	Expert	Trained	Trained
Expert	Expert	Trained	Trained
Trained	Trained	Expert	Trained
Expert	Trained	Expert	Trained
Trained	Expert	Expert	Trained
Expert	Expert	Expert	Trained

Games Against Expert



Games Against Expert

Trained Play



Summary

- Learned to beat 'expert' by a large margin
- Program plays well, but lacks deep analysis of game

Ongoing Work

- Learn with 'shooting the moon' turned on
  - Duplicate all features three times
    - Points are split
    - We have all the points
    - Someone else has all the points
- Compare steady-state play

Steady-State Results

(1)	(2)	Score (1)	Score (2)
Self-Trained 90k games	Expert	6.27	7.10
Expert-Trained 220k games	Expert	6.17	7.35
Self-Trained 90k games	Expert-Trained 220k games	6.35	7.03

Future Work

- Optimize code
- Different algorithm than max^n
- Better ways of combining features
- Play against humans
- Passing cards
- Imperfect information

Thank You

- Joint work with Adam White
- Thanks to Rich Sutton and Mark Ring