Play Selection in American Football

Daniel Bestard, Hans-Peter Hoellwirth, Akhil Lohia, Michael Cameron

Barcelona Graduate School of Economics

April 5, 2017

Motivation

- Subconsciously, sport players compute probabilities

Motivation

- Subconsciously, sport players compute probabilities
- They use estimates of distances, angles and probabilities of events occurring

Motivation

- Subconsciously, sport players compute probabilities
- They use estimates of distances, angles and probabilities of events occurring
- Aim: provide a framework for optimising decisions in American football

Motivation

- Subconsciously, sport players compute probabilities
- They use estimates of distances, angles and probabilities of events occurring
- Aim: provide a framework for optimising decisions in American football
- Use of DP to come up with the optimal policies. Two different implementations:

Motivation

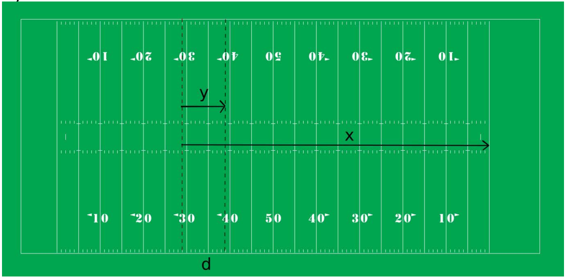
- Subconsciously, sport players compute probabilities
- They use estimates of distances, angles and probabilities of events occurring
- Aim: provide a framework for optimising decisions in American football
- Use of DP to come up with the optimal policies. Two different implementations:
    - Exact solution is feasible under some assumptions

Motivation

- Subconsciously, sport players compute probabilities
- They use estimates of distances, angles and probabilities of events occurring
- Aim: provide a framework for optimising decisions in American football
- Use of DP to come up with the optimal policies. Two different implementations:
  - Exact solution is feasible under some assumptions
  - For more general cases, approximations of the expected reward-to-go function are provided (API and OPI)

Parameters of the dynamic programming algorithm

- State of the system:
  - x_i: yards to the goal line
  - y_i: yards to the first down
  - d: number of downs
- Policies or actions that players can take:
  - P: pass
  - R: run
  - U: punt
  - K: kick
- Rewards:
  - Touchdown: 6.8
  - Field goal: 3
  - Safety: -2
  - Opposition score $= -\frac{6.8x}{100}$



Exact Dynamic Programming

DP Equation

$$\mu^k(i) = arg \max_{u \in U} \left[\sum_{j \in S} p_{ij}(u)(g(i,u,j) + J^{\mu^{k-1}}(j)) \right] \ \forall i \in S$$

- $p_{ij}(u)$: transition probabilities
- $g(i,u,j)$: reward function
- $J^{\mu^{k-1}}(j)$: reward-to-go function

J is computed exactly using the 15250 possible states of the system.

Simulations

- We create a reasonable class of policies and implement it.
- Policies are compared by calculating the points from one drive.
- Simulations for an optimal heuristic policy are run from the starting state of $(x_i, y_i, d) = (80, 10, 1)$.
- Example of a simulation:

$$\begin{bmatrix}25\\10\\1\end{bmatrix} \longrightarrow_0^P \begin{bmatrix}17\\2\\2\end{bmatrix} \longrightarrow_0^R \begin{bmatrix}14\\10\\1\end{bmatrix} \longrightarrow_0^P \begin{bmatrix}10\\6\\2\end{bmatrix} \longrightarrow_0^P \begin{bmatrix}10\\6\\3\end{bmatrix} \longrightarrow_0^R \begin{bmatrix}8\\4\\4\end{bmatrix} \longrightarrow_3^K T$$

Policy Update

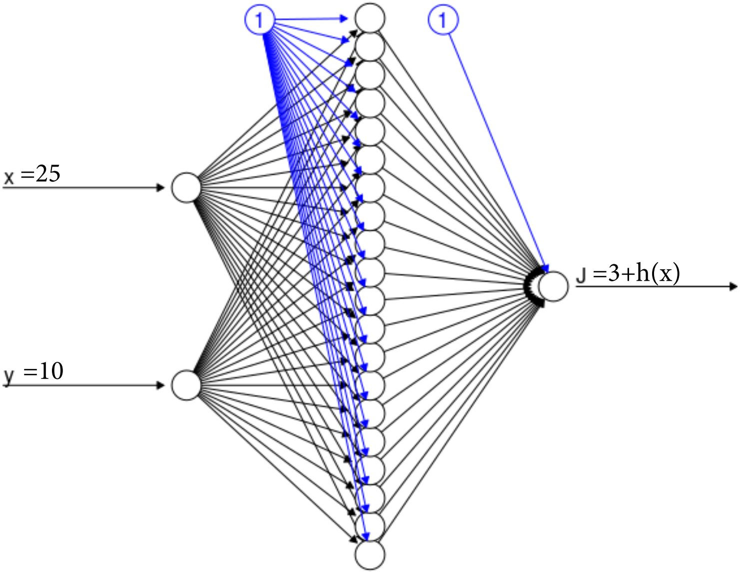
Approximated DP algorithm

$$\mu^k(i) = arg \max_{u \in U} \left[\sum_{j \in S} p_{ij}(u)(g(i, u, j) + \widetilde{J}^{\mu^{k-1}}(j)) \right]$$

API and OPI Algorithm

- Algorithm:
  1. Start an initial policy μ_0
  2. For each $k \in \{1, \ldots, K\}$:
     1. Given μ_{k-1}, simulate N_s sample trajectories
     2. Fit the neural network using the sample data to estimate the $J^{\mu^{k-1}}$
     3. Update policy to get μ^k
- Two different ways to make the approximations
  - API: Many training sample points, few iterations
  - OPI: Few training sample points, many iterations

Neural Network

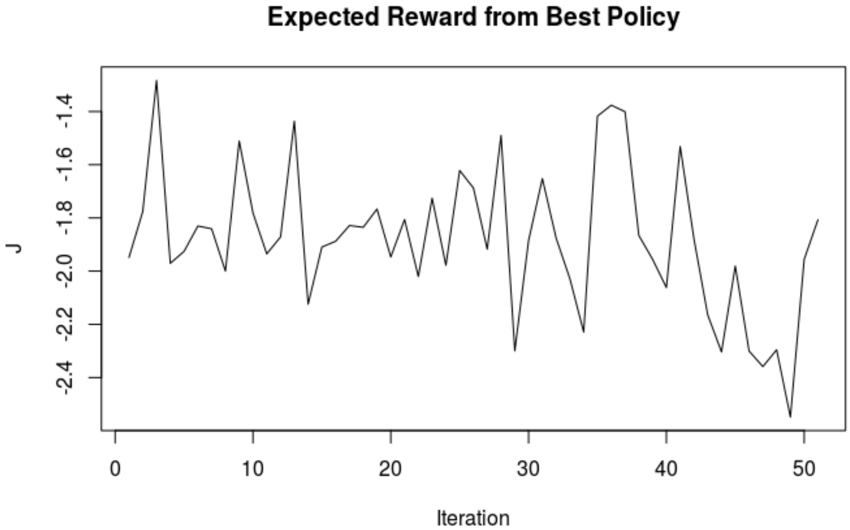


Policy Update

Approximated DP algorithm

$$\mu^k(i) = arg \max_{u \in U} \Big[\sum_{j \in S} p_{ij}(u)(g(i,u,j) + \widetilde{J}^{\mu^{k-1}}(j)) \Big]$$

Results



Expected Reward from Best Policy
J
-1.4
-1.6
-1.8
-2.0
-2.2
-2.4
0
10
20
30
40
50
Iteration

Super Bowl XLIX

Seahawks should have run!