Behavior Programming

Group Members:
Daniel Biskup, Mohammad Asif Khan, Nilesh Chakraborty, Nofel Mahmood, Saptarishi Bhattacharya, Saurav Ganguli

Overview

Connect 4

Fuzzy Control for Breakout

Self Organizing Map

Bayesian Learning & Trajectory Planning

Connect 4

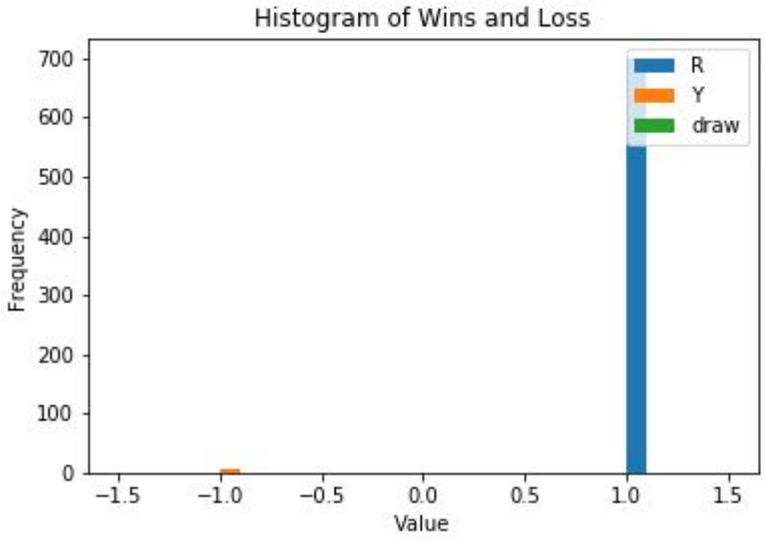
- Three experiments ran in parallel.
- Connect4 board size: 19x19
- Algorithm: Recursive DFS MinMax with depth restriction
- Total System RAM usage at any time: 3.4GB
- System:

  16GB RAM

  Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz

  (two cores, four threads)

Connect 4

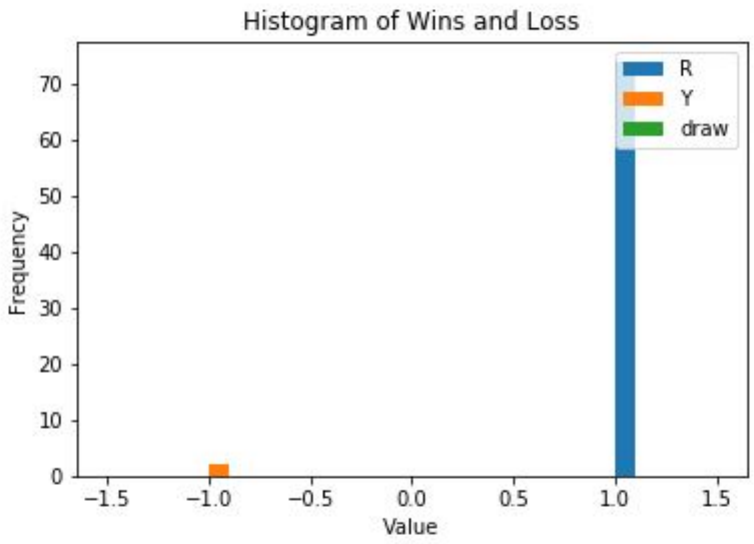


3 vs Random
Red plays MinMax with a depth of three.
Yellow moves at random.

Games: 708
Red: 700
Yellow: 8
Draw: 0

Total time: 18h
Average time per Game: 92s = 1.5min

Connect 4

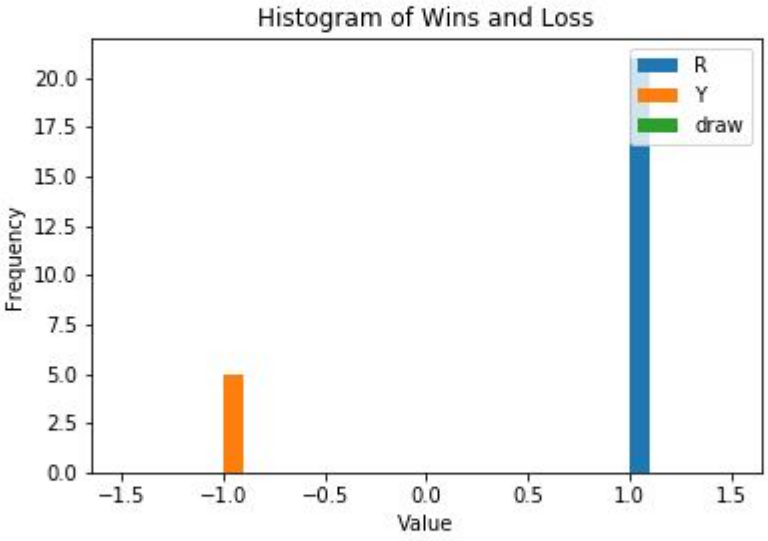


3 vs 1
Red plays MinMax with a depth of three.
Yellow plays MinMax with a depth of one.

Games: 76
Red: 74
Yellow: 2
Draw: 0

Total time: 18h
Average time per Game: 855s = 14.25min

Connect 4



3 vs 3

Red and Yellow play MinMax with a depth of three.

Games: 26

Red: 21

Yellow: 5

Draw: 0

Total time: 17.7h

Average time per Game: 2455s = 41min

Breakout

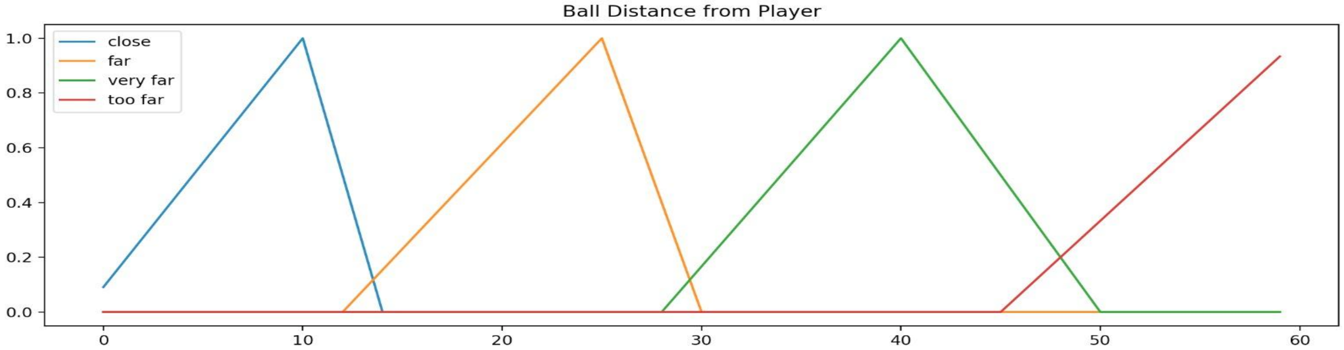
- Rules fire with certain degree of acceptance. Define membership functions.
    - Antecedent: Distance of the ball from the paddle.
    - Consequent: Movement of Paddle.

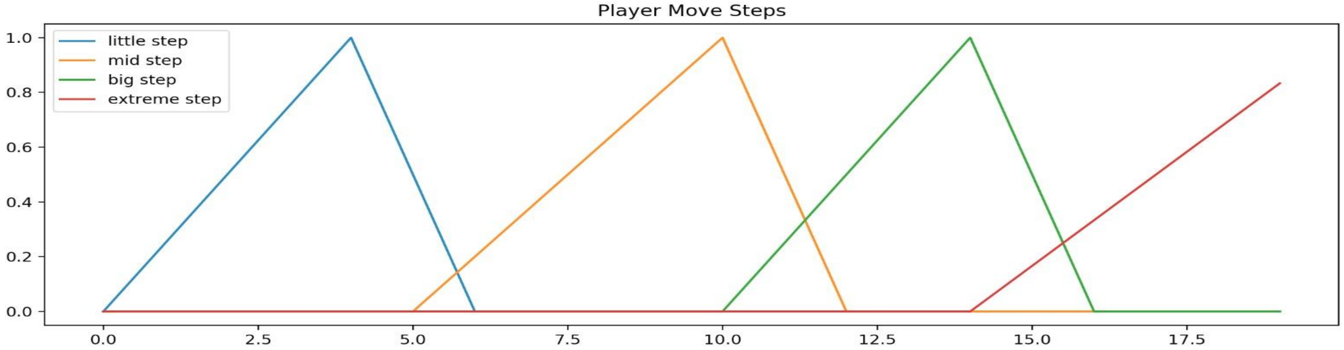
- SciKit-Fuzzy (https://pythonhosted.org/scikit-fuzzy/)

Breakout

Rules	Antecedent (Distance)	Consequent (Paddle movement)
R0	Close	Little steps
R1	Far	Medium steps
R2	Very Far	Big steps
R3	Too Far	Extreme steps

Membership Functions

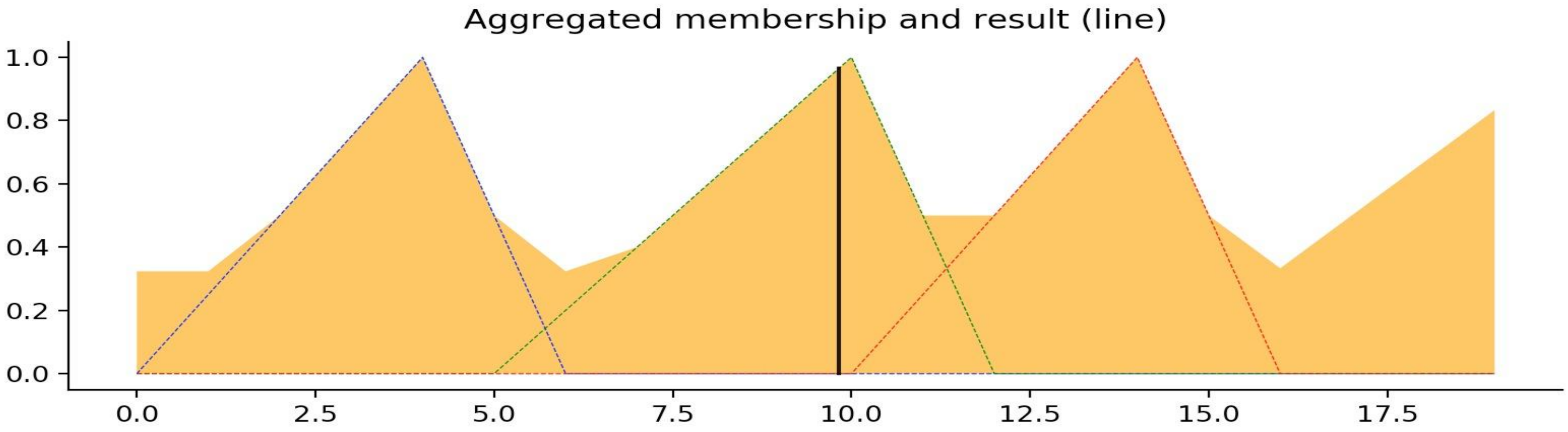




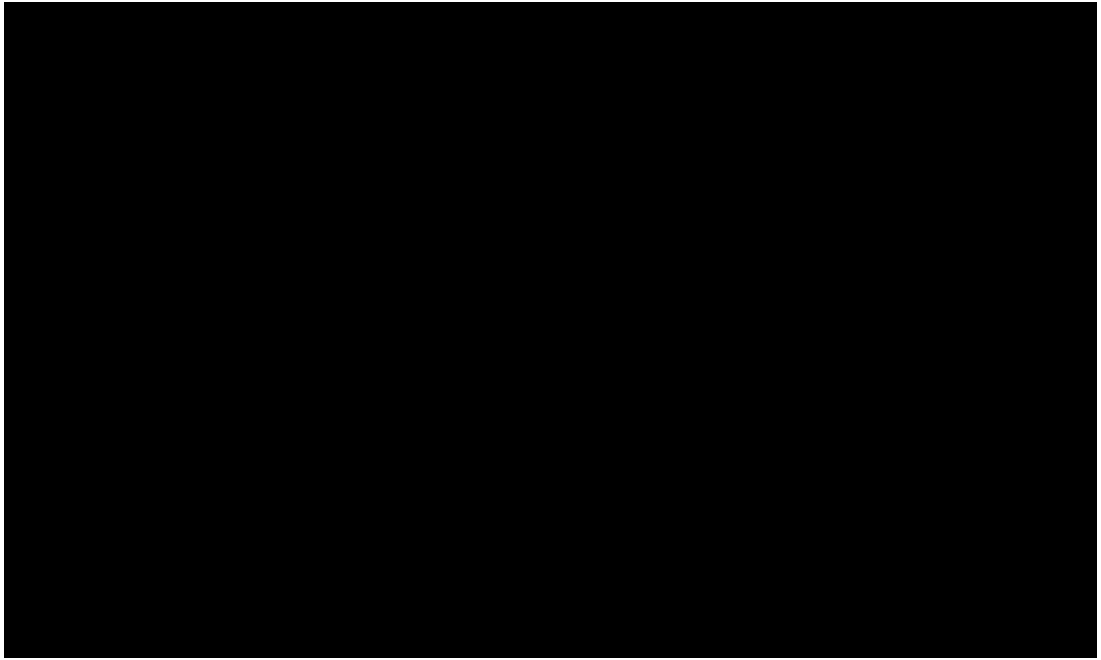
Defuzzification

- Used centroid method to perform defuzzification.

- For an input value of 2.56 as distance of ball from paddle, the movement of paddle to the left or right would be 9.8 steps



Video

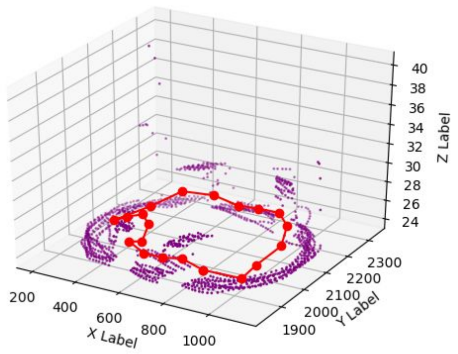


SOM:Self Organizing Map

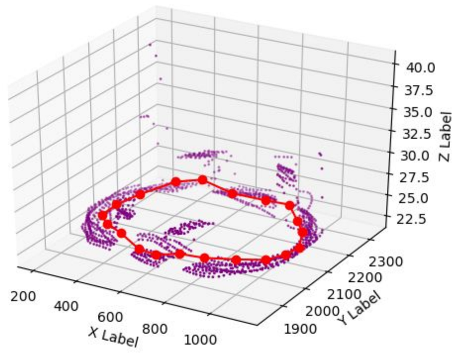
Learning topology of data.

Performance Metric: Quantization Error, Topological Error

SOM

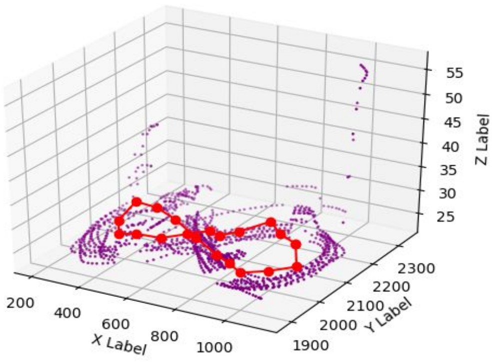


Random Initialization

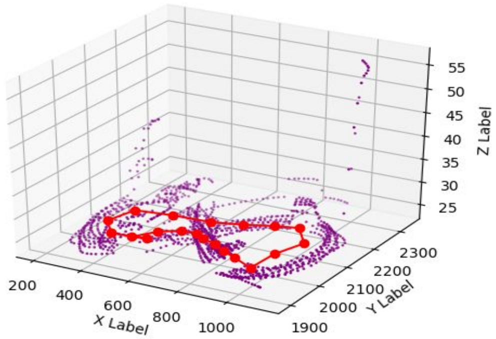


PCA Initialization

SOM



Random Initialization



PCA Initialization

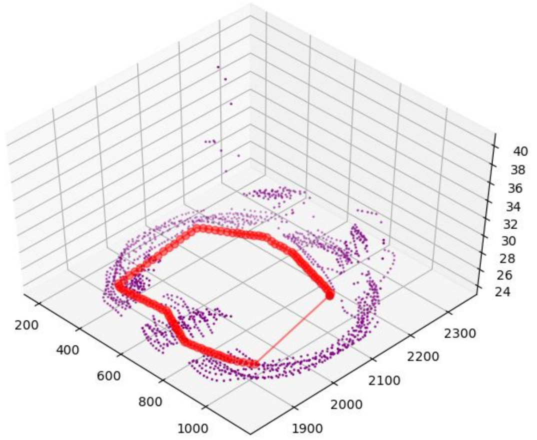
SOM

	Quantization Error	Topological Error
Path 1, Random Initialization	64.9778	0.0941
Path 1, PCA Initialization	**47.2564**	**0.0339**
Path 2, Random Initialization	64.5626	0.1343
Path 2, PCA Initialization	**59.2096**	**0.1166**

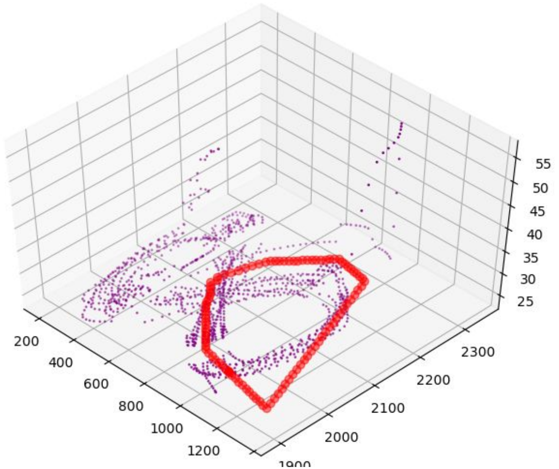
Bayesian Imitation Learning

Behaviour as a sequence of motor primitives.

Bayesian Imitation Learning



Map 1



Map 2

Bayesian Imitation Learning

Why trajectory differs for map 2?

Bayesian Imitation Learning

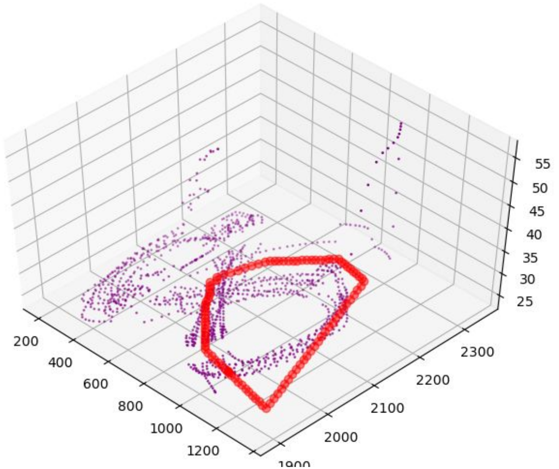
Why trajectory differs for map 2?

Solution:

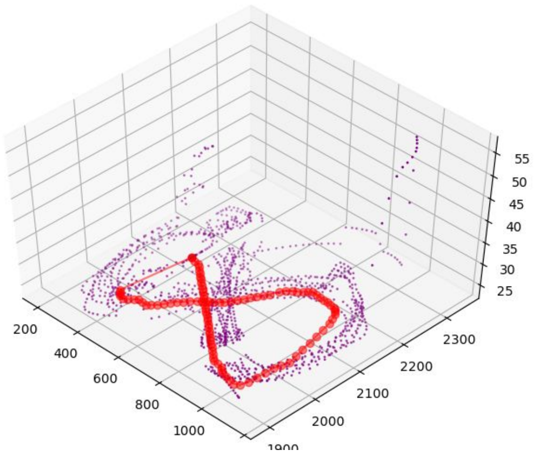
Probability of action dependent on previous actions. We used first order markov dependency following [*Thuran et. al.*]

$$a_t = \underset{r_j}{argmax} \frac{p(r_j|s_i)p(r_j|r_{j-1})}{\sum_{k=1}^{n} p(r_k|r_j)p(r_k|s_i)}$$

Bayesian Imitation Learning



Map 2



Map 2 with priors

Conclusion

- Connect 4
  - Exponential state space complexity. Use depth limited search.
- Breakout:
  - Smooth control with simple interpretable rules.
- SOM
  - Dependent on initial weights.
    - Using PCA can help.
  - Can capture topology of data.
  - Challenges: How to determine the topology?
    - Possible Solution: Looking at the Betti Numbers (Algebraic Topology).
- Bayesian Learning
  - Human like control for trajectory planning.
  - Challenges: Environmental difficulties.
  - Possible Solution: More conditional probabilities expressing greater variety of dependency.

Questions

Thank you!