UNIVERSITY OF AMSTERDAM

Computational Social Choice

On Voting Approaches for Robust Reinforcement Learning Ensembles

Jose Gallego, Alexandre Lasbleis, Isa Steinecker, Yujie Xing

June 9, 2017

Outline

Introduction

Pole Balancing

Reinforcement Learning

Set up

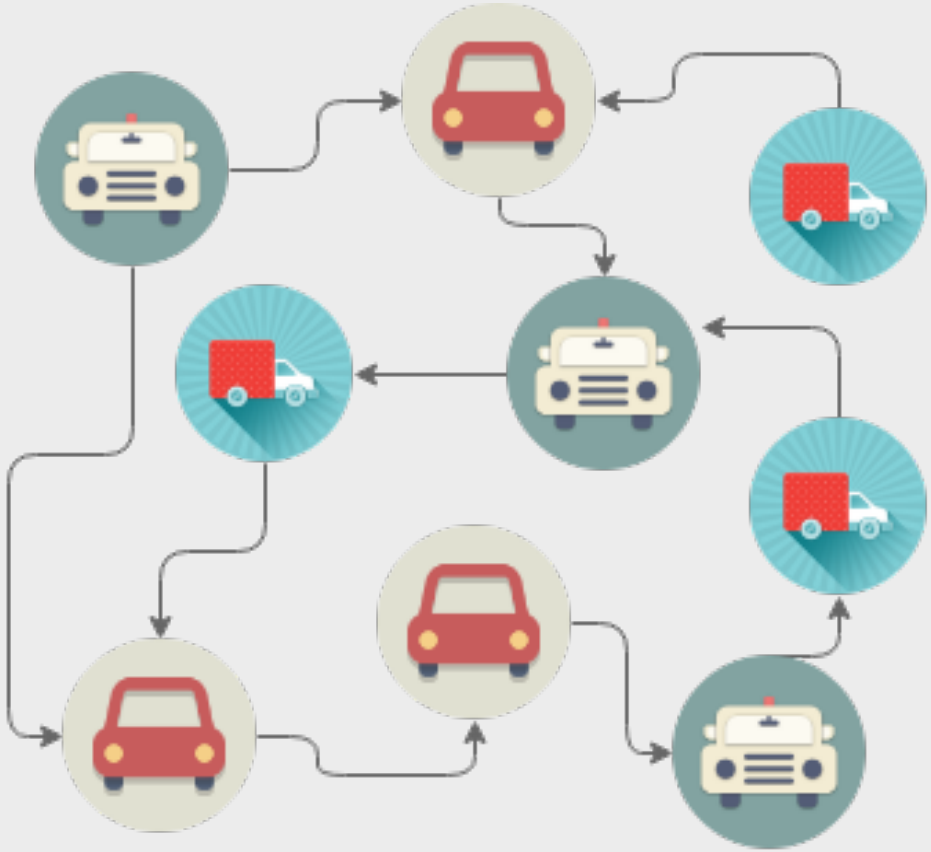
Results

Discussion

Future work

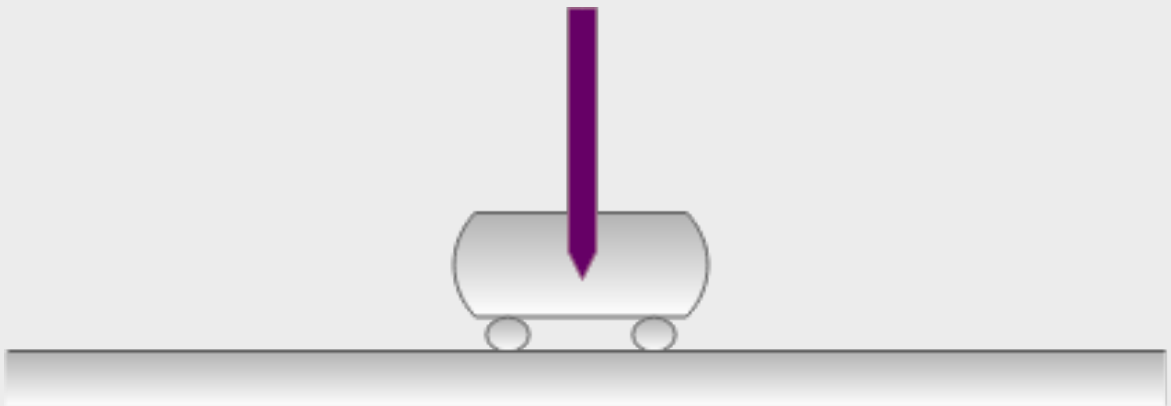
Conclusions

Introduction



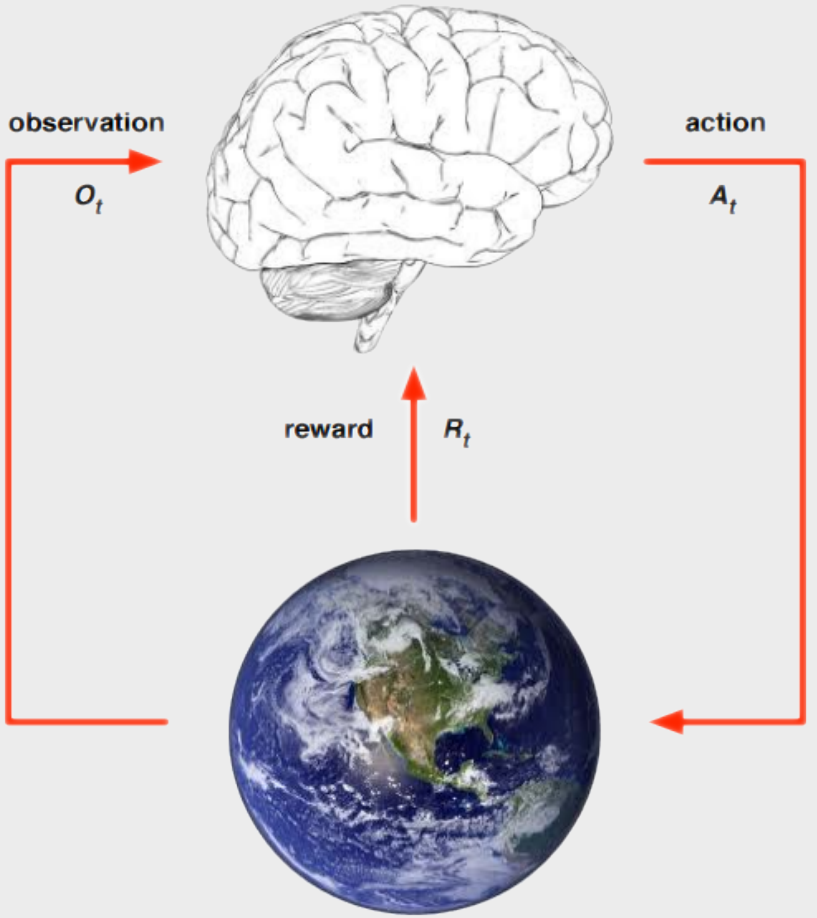
Pole Balancing

- **Goal**: Keep pole upright as much time as possible
- **How**: Apply force to the cart
- **Information**: $[x, \dot{x}, \theta, \dot{\theta}]$
- **End**: After 500 time steps or tilt angle $> \theta_{\text{max}}$, end of track



Reinforcement Learning

Reinforcement Learning (RL) is a sub-field of Machine Learning in which an agent learns optimal behavior in a stochastic environment in order to maximize a reward signal.



Definition

A Markov Decision Process is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$ such that:

- $\mathcal{S}$ is a *finite* set of states
- $\mathcal{A}$ a *finite* set of actions
- $\mathcal{P}(s, a, s')$ is a state transition probability function, $\mathcal{P}^a_{ss'} = \mathbb{P}[S_{t+1} = s' | S_t = s, A_t = a]$
- $\mathcal{R}(s, a)$ describes the immediate reward, $\mathcal{R}^a_s = \mathbb{E}[R_{t+1} | S_t = s, A + t = a]$
- γ is a discount factor $\gamma \in [0, 1]$

Strategy

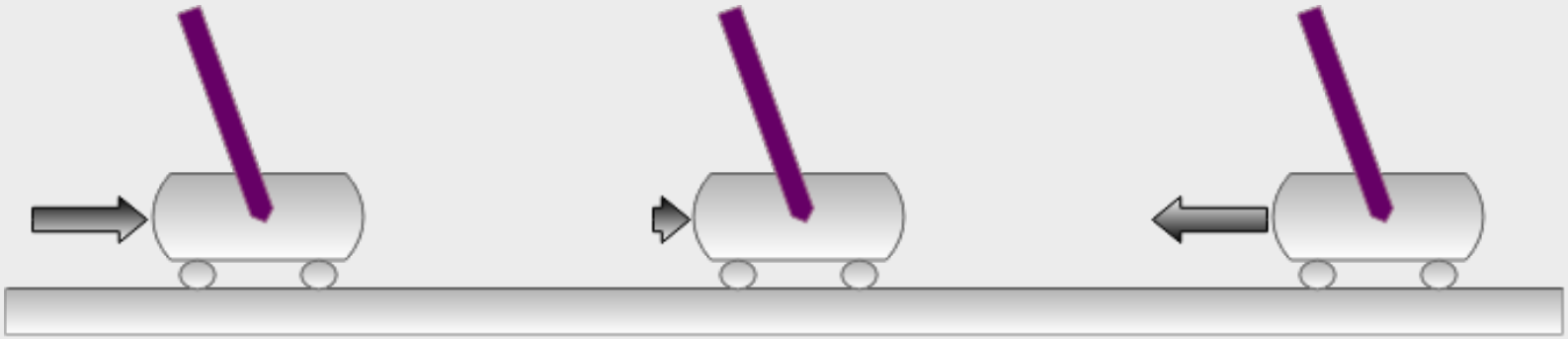
Learn a function $\mathcal{Q}(s, a)$ which describes what is the long term quality of choosing an action at a particular state.

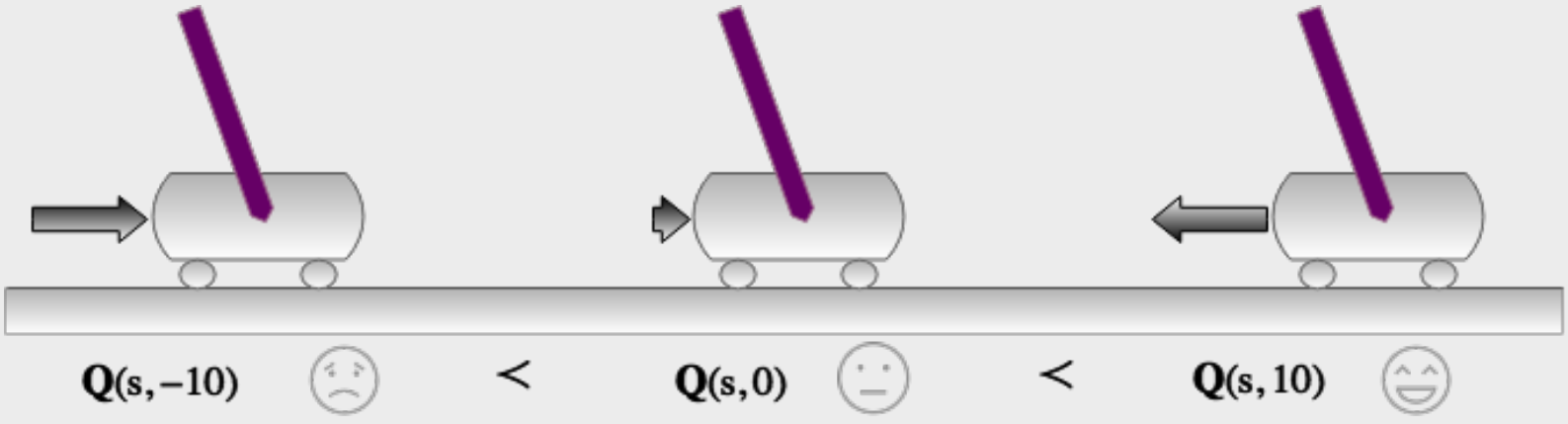
Then use this function to define the policy for how to react in each situation:

$$\pi(s) = \arg\max_{a \in \mathcal{A}} \mathcal{Q}(s, a)$$

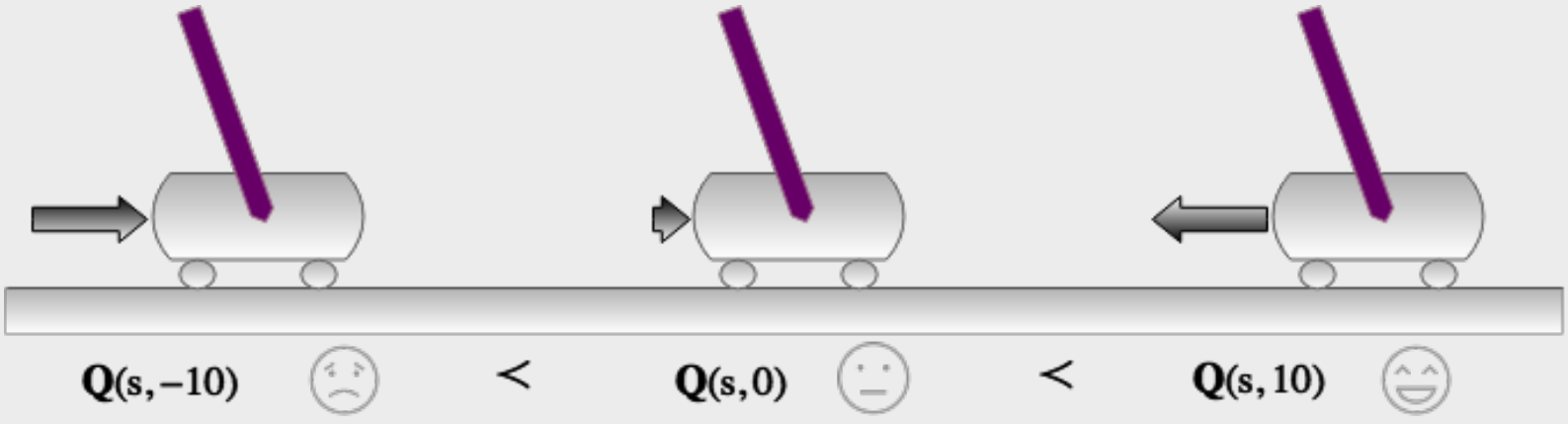
⚠ Approximate $\mathcal{Q}$ by a Neural Network (NN) and train it using gradient descent:

$$\Delta \mathcal{Q} \propto -\nabla \mathcal{Q}$$





We can obtain a ballot using Q!



We can obtain a ballot using $\mathcal{Q}$!

Potential Issues in RL

- Training agents takes (a lot of) time
- If using a gradient-based approach, beware of local optima
- How good is the expressive power of the proposed NN architecture?
- Are the values of Q reliable enough?
- How robust are the agents to observation noise?

Set up

Task modelling

- 11 possible actions (apply -50 to 50 Newton)
- 10 semi trained agents (different structure each)
- Voting rules: Borda, Plurality, Copeland, Range Voting
- Distinction between 2 cases
  - Control case 1: no noise is applied
  - Test case 2: noise is added to the observation of the agents(position, velocity,angle, angular velocity) and updated each time step

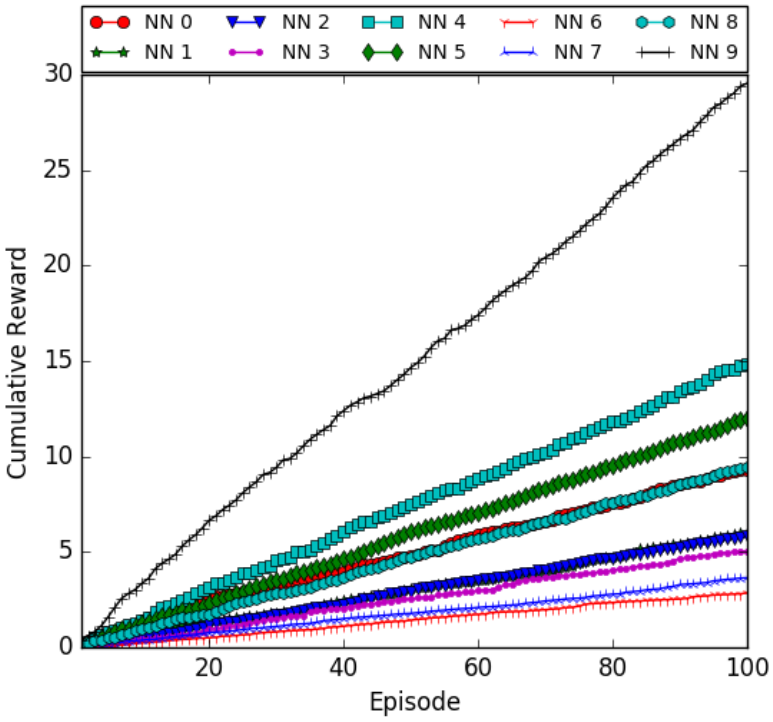
Why?

- 11 actions: we need enough for voting rules to show different behaviours
- 10 agents: trade-off between reasonable amount of voters and individual performance
- Voting rules: Most common ones and easy to implement
- Distinction between 2 cases
  - No noise: influence of voting under optimal conditions
  - Noise: simulate a more realistic environment

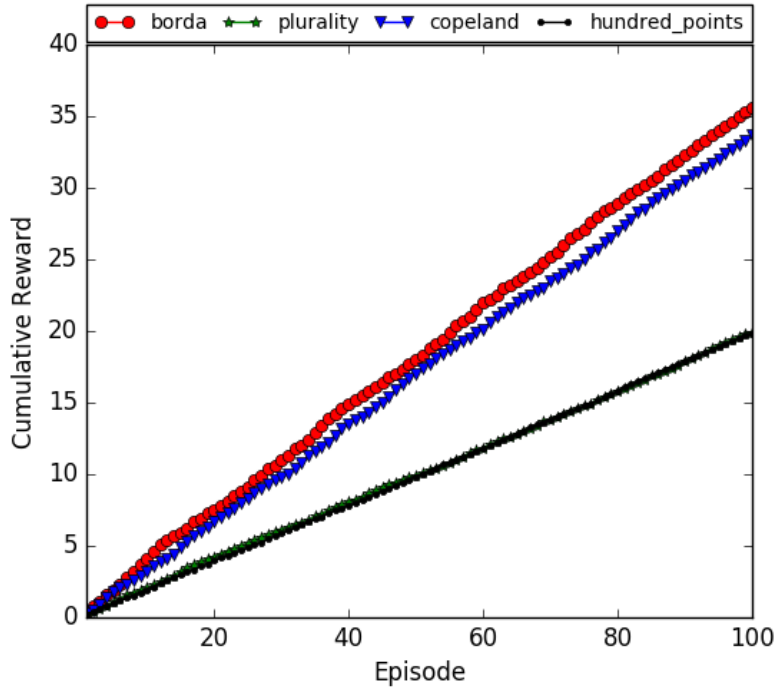
Statistical significance tests

- *Two-tailed t-Test*: test the hypothesis that two populations have equal means.
- *p-value*: how likely is that the data would come from a more extreme scenario than our observations.
- *Kendall's* τ: this is a measure of the similarity between two rankings (swap distance).
- *Spearman's* ρ: measures the similarity between rankings but takes the magnitude of the swaps into account.
- *Top 1 Match*: percentage of times that two rules coincide on the selected action.

Results



(a) Individual



(b) Ensemble

Figure: Performance under no noise

No noise - t-Test:

- **H0**: There is no difference between the performance of two rules
- Up to 99% confidence, Borda and Copeland perform better than Plurality and Range Voting.
- We can not infer a significant difference between Borda and Copeland; or between Plurality and Range Voting.

Top 1 Match	Borda	Plurality	Copeland	Range Voting
Borda	1	0.239	0.779	0.188
Plurality	0.357	1	0.398	0.600
Copeland	0.758	0.277	1	0.260
Range Voting	0.453	0.526	0.462	1

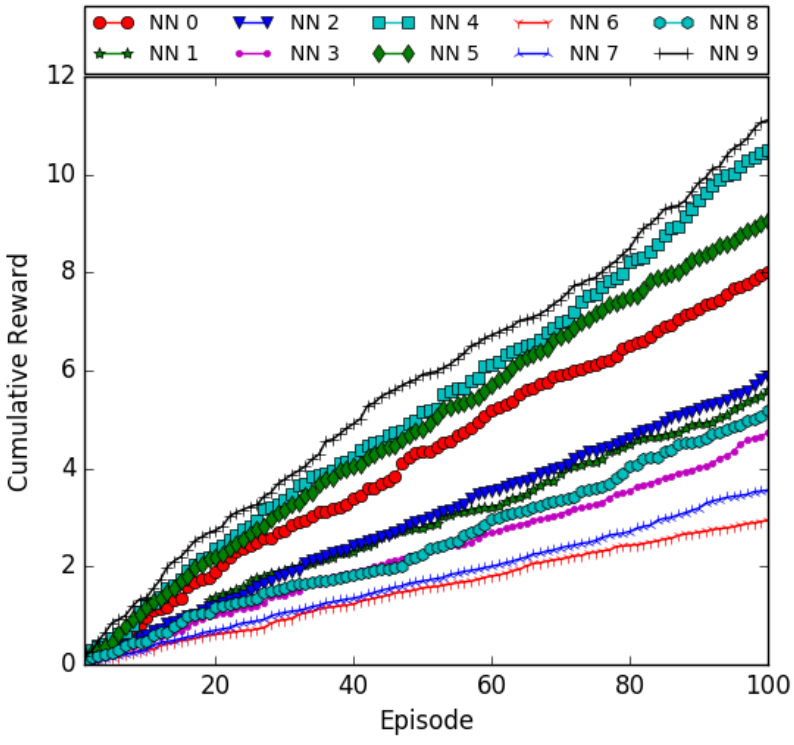
Table: Experiment 1: no noise

ρ-Value	Borda	Plurality	Copeland	Range Voting
Borda	1	0.133	0.470	0.297
Plurality	0.266	1	0.272	0.311
Copeland	0.504	0.156	1	0.338
Range Voting	0.522	0.296	0.500	1

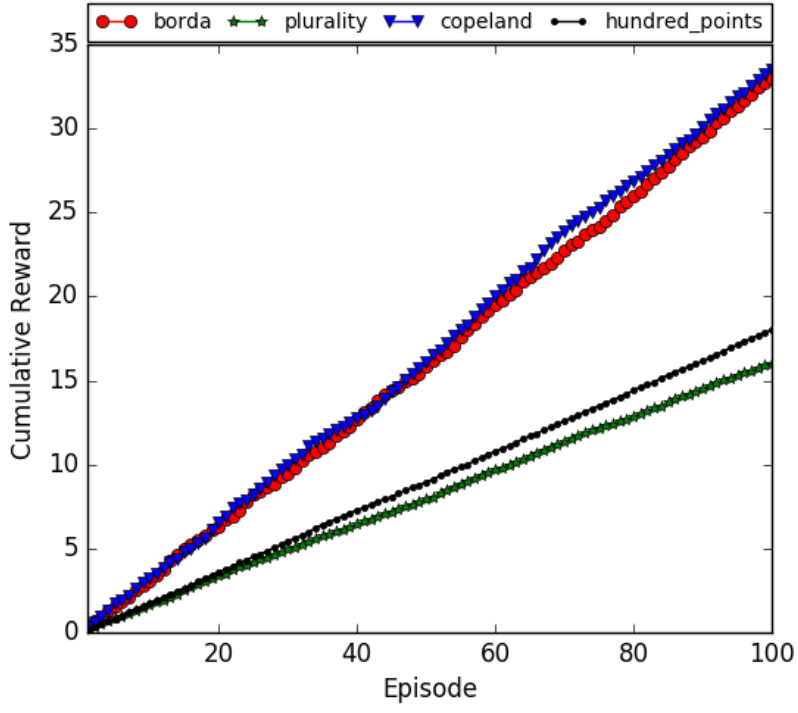
Table: Experiment 1: no noise

τ-Value	Borda	Plurality	Copeland	Range Voting
Borda	1	0.097	0.373	0.222
Plurality	0.195	1	0.202	0.233
Copeland	0.402	0.114	1	0.253
Range Voting	0.419	0.220	0.394	1

Table: Experiment 1: no noise

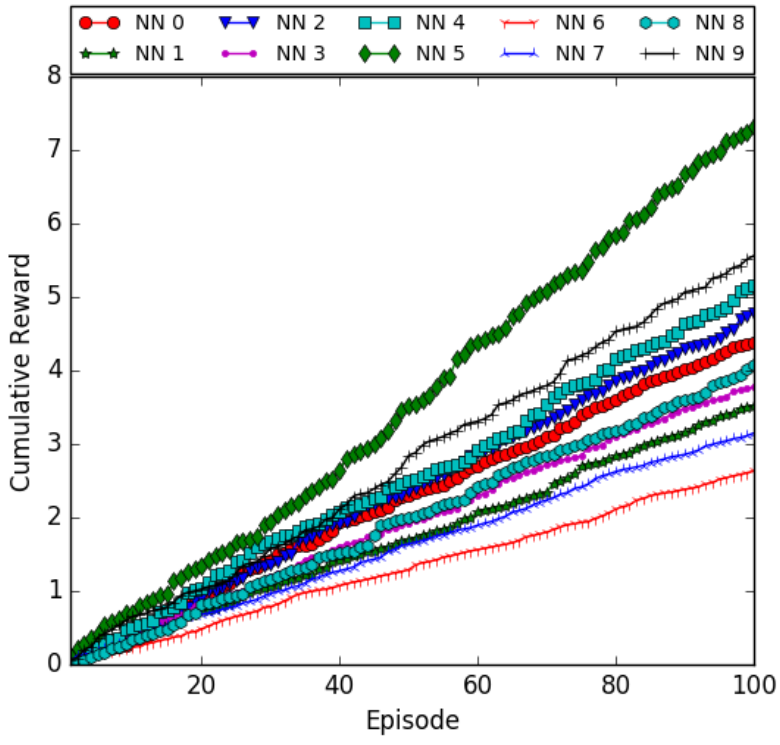


(a) Individual

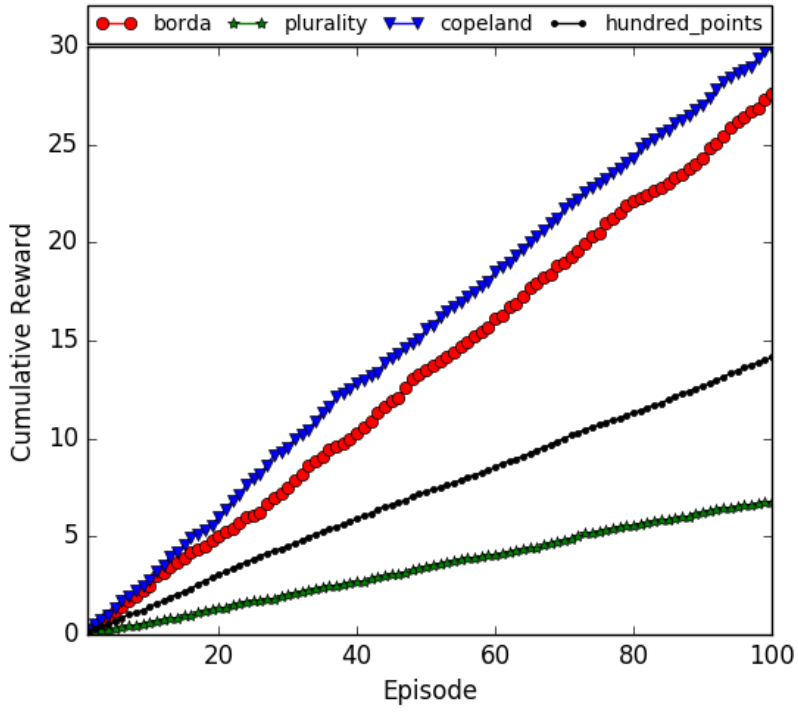


(b) Ensemble

Figure: Performances under noise 2

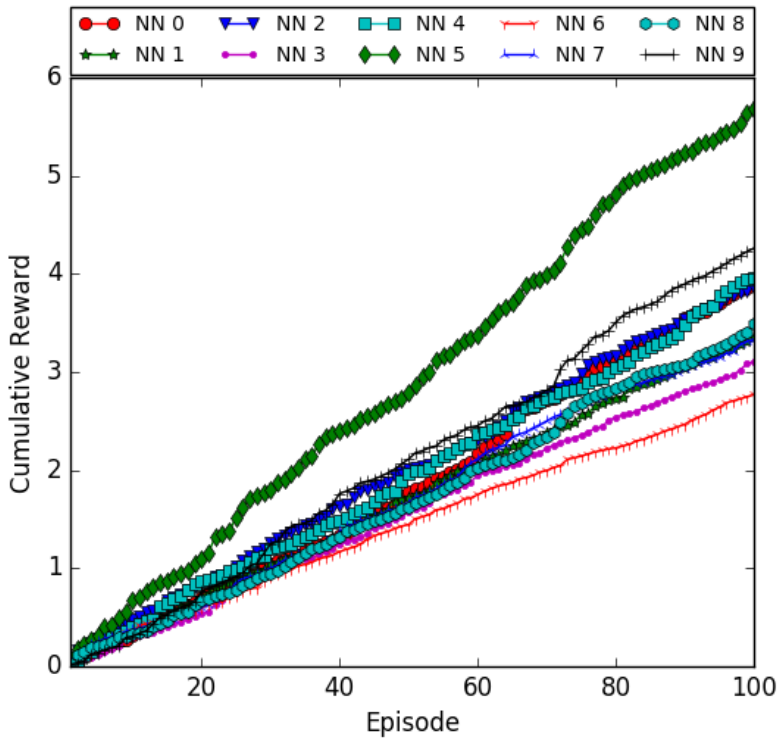


(a) Individual

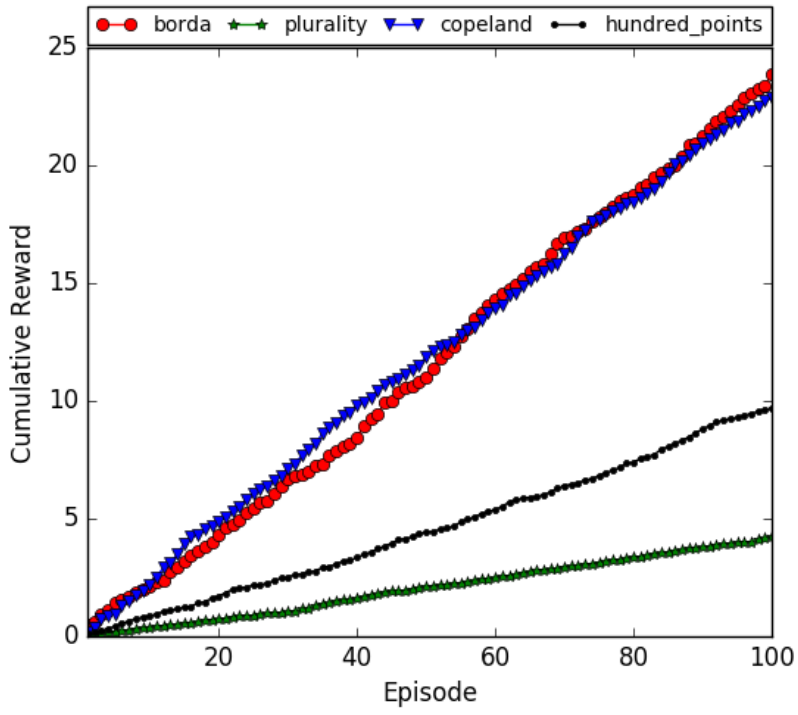


(b) Ensemble

Figure: Performances under noise 10



(a) Individual



(b) Ensemble

Figure: Performance under noise 20

Top 1 Match	Borda	Plurality	Copeland	Range Voting
Borda	1	0.239	0.779	0.188
Plurality	0.357	1	0.398	0.600
Copeland	0.758	0.277	1	0.260
Range Voting	0.453	0.526	0.462	1
Borda	1	0.296	0.743	0.315
Plurality	0.308	1	0.336	0.478
Copeland	0.742	0.326	1	0.343
Range Voting	0.347	0.490	0.377	1
Borda	1	0.293	0.742	0.313
Plurality	0.306	1	0.335	0.482
Copeland	0.738	0.323	1	0.341
Range Voting	0.341	0.481	0.364	1
Borda	1	0.295	0.737	0.312
Plurality	0.330	1	0.351	0.499
Copeland	0.742	0.314	1	0.331
Range Voting	0.332	0.492	0.353	1

Table: Top 1 Match for noise 0-2-10-20

τ-Value	Borda	Plurality	Copeland	Range Voting
Borda	1	0.097	0.373	0.222
Plurality	0.195	1	0.202	0.233
Copeland	0.402	0.114	1	0.253
Range Voting	0.419	0.220	0.394	1
Borda	1	0.163	0.448	0.246
Plurality	0.145	1	0.145	0.166
Copeland	0.444	0.120	1	0.244
Range Voting	0.282	0.153	0.278	1
Borda	1	0.131	0.458	0.282
Plurality	0.168	1	0.171	0.185
Copeland	0.457	0.133	1	0.274
Range Voting	0.327	0.177	0.317	1
Borda	1	0.119	0.444	0.237
Plurality	0.127	1	0.131	0.152
Copeland	0.444	0.116	1	0.229
Range Voting	0.260	0.147	0.261	1

Table: τ-Value for noise 0-2-10-20

ρ-Value	Borda	Plurality	Copeland	Range Voting
Borda	1	0.016	0.547	0.320
Plurality	0.198	1	0.199	0.224
Copeland	0.543	0.164	1	0.319
Range Voting	0.366	0.208	0.360	1
Borda	1	0.178	0.559	0.365
Plurality	0.228	1	0.233	0.250
Copeland	0.557	0.181	1	0.354
Range Voting	0.420	0.239	0.407	1
Borda	1	0.163	0.543	0.320
Plurality	0.198	1	0.199	0.224
Copeland	0.543	0.164	1	0.244
Range Voting	0.366	0.208	0.360	1
Borda	1	0.163	0.543	0.310
Plurality	0.174	1	0.180	0.206
Copeland	0.542	0.158	1	0.300
Range Voting	0.340	0.198	0.340	1

Table: ρ-Value for noise 0-2-10-20

p-Values		*No noise*	*Noise 2*	*Noise 10*	*Noise 20*
Borda	*Plurality*	0.00	0.00	0.00	0.00
Borda	*Copeland*	0.09	0.70	0.10	0.48
Borda	*Range Voting*	0.00	0.00	0.00	0.00
Plurality	*Copeland*	0.00	0.00	0.00	0.00
Plurality	*Range Voting*	0.78	0.00	0.00	0.00
Copeland	*Range Voting*	0.00	0.00	0.00	0.00

Table: t-Test for difference between reward means

Discussion

Rule	Condorcet	Pareto	Cardinal/Ordinal	Information
Borda	✗	✓	Ordinal	*nm* (ordering)
Plurality	✗	✓	Ordinal	*n*
Copeland	✓	✓	Ordinal	*nm* (ordering)
Range Voting	✗	✓	Cardinal	*nm* (number)

Table: Properties of the chosen voting rules

Future work

Future Work

- Investigate the influence of different types of noise. Which one seems most realistic/interesting?
- Try more challenging RL tasks
- Analyze the impact of the dimensionality of the spaces of voters and alternatives. Rule complexity?

Conclusions

Conclusions

- Voting in a reinforcement learning task like pole-balancing can boost performance (3x) with no training time trade-off.
- Problems due to *numerical* unreliability can be overcome with *ordinal* information.
- Also robustness under extreme noise conditions to the performance.
- In this task: Copeland $\succeq$ Borda $\succ$ Range Voting $\succ$ Plurality