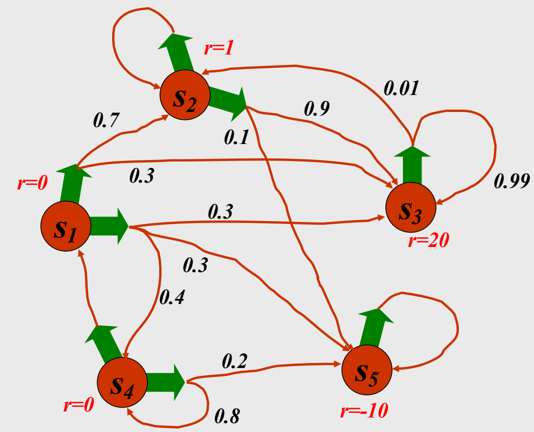
CSE-571
Robotics

Planning and Control:

Markov Decision Processes

1

Markov Decision Process (MDP)



2

Markov Decision Process (MDP)

- **Given:**
- States x
- Actions u
- Transition probabilities $p(x'|u,x)$
- Reward / payoff function $r(x,u)$

- **Wanted:**
- Policy $\pi(x)$ that maximizes the future expected reward

3

Rewards and Policies

- Policy (general case):

$$\pi: \quad z_{1:t-1}, u_{1:t-1} \rightarrow \; u_t$$

- Policy (fully observable case):

$$\pi: \quad x_t \rightarrow \; u_t$$

- Expected cumulative payoff:

$$R_T = E\left[\sum_{\tau=1}^{T} \gamma^{\tau} r_{t+\tau}\right]$$

  - T=1: greedy policy
  - T>1: finite horizon case, typically no discount
  - T=infty: infinite-horizon case, finite reward if discount < 1

4

Policies contd.

- Expected cumulative payoff of policy:

$$R_T^\pi(x_t) = E\left[\sum_{\tau=1}^{T} \gamma^\tau r_{t+\tau} \mid u_{t+\tau} = \pi(z_{1:t+\tau-1} u_{1:t+\tau-1})\right]$$

- Optimal policy:

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \quad R_T^\pi(x_t)$$

- 1-step optimal policy:

$$\pi_1(x) = \underset{u}{\operatorname{argmax}} \quad r(x,u)$$

- Value function of 1-step optimal policy:

$$V_1(x) = \gamma \max_u r(x,u)$$

5

2-step Policies

- Optimal policy:

$$\pi_2(x) = \underset{u}{\operatorname{argmax}} \quad \left[r(x,u) + \int V_1(x') p(x' \mid u,x) dx'\right]$$

- Value function:

$$V_2(x) = \gamma \max_u \quad \left[r(x,u) + \int V_1(x') p(x' \mid u,x) dx'\right]$$

6

T-step Policies

- Optimal policy:

$$\pi_T(x) = \underset{u}{\operatorname{argmax}} \quad \left[r(x,u) + \int V_{T-1}(x') p(x' \mid u,x) dx'\right]$$

- Value function:

$$V_T(x) = \gamma \max_u \quad \left[r(x,u) + \int V_{T-1}(x') p(x' \mid u,x) dx'\right]$$

7

Infinite Horizon

- Optimal policy:

$$V_\infty(x) = \gamma \max_u \quad \left[r(x,u) + \int V_\infty(x') p(x' \mid u,x) dx'\right]$$

- Bellman equation
- Fix point is optimal policy
- Necessary and sufficient condition

8

Value Iteration

- for all x do

  $$\hat{V}(x) \leftarrow r_{\min}$$

- endfor

- repeat until convergence
  - for all x do

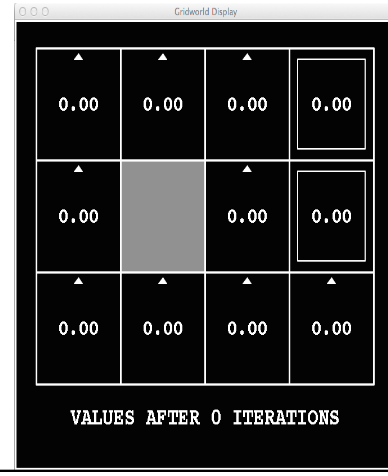
  $$\hat{V}(x) \leftarrow \gamma \max_u \left[r(x,u) + \int \hat{V}(x') p(x' \mid u,x) dx' \right]$$

  - endfor
- endrepeat

$$\pi(x) = \operatorname{argmax}_u \left[r(x,u) + \int \hat{V}(x') p(x' \mid u,x) dx' \right]$$

9

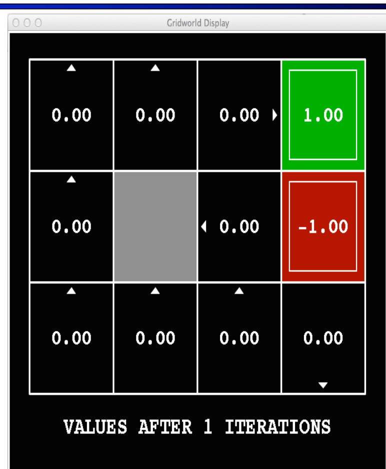
k=0



Noise = 0.2
Discount = 0.9
Living reward = 0

10

k=1



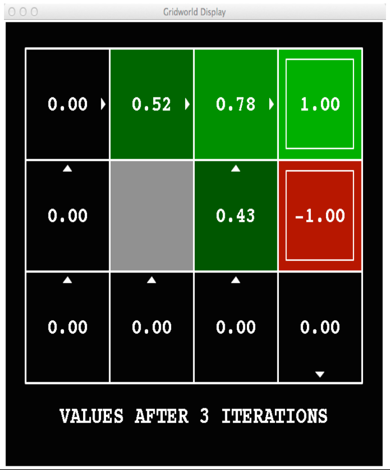
Noise = 0.2
Discount = 0.9
Living reward = 0

11

k=2

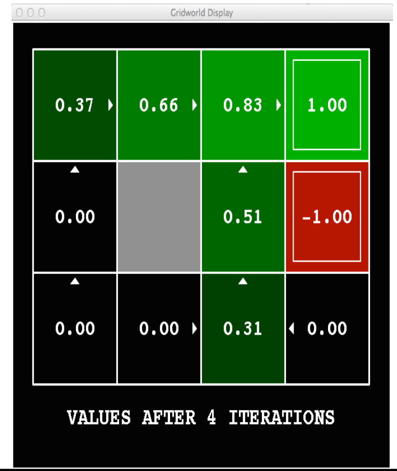
12

k=3



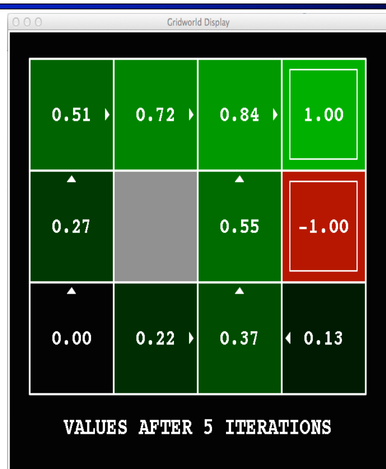
13

k=4



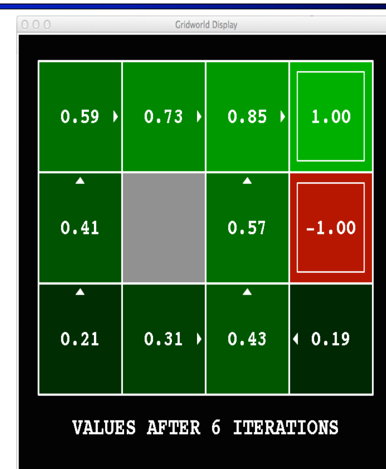
14

k=5



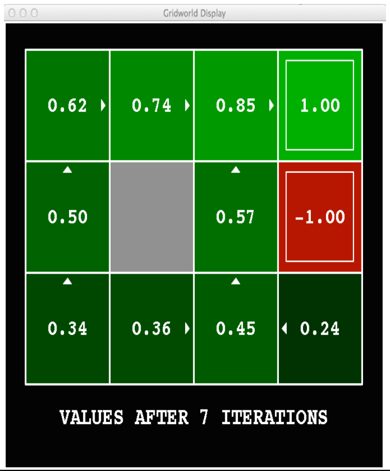
15

k=6



16

k=7

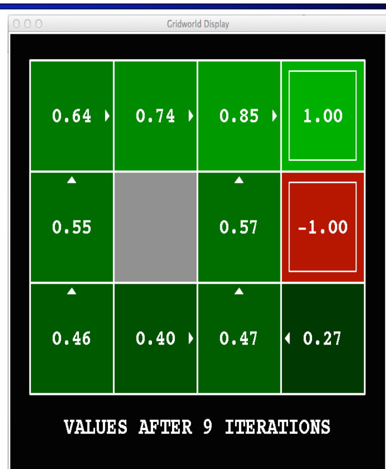


17

k=8

18

k=9



19

k=10

20

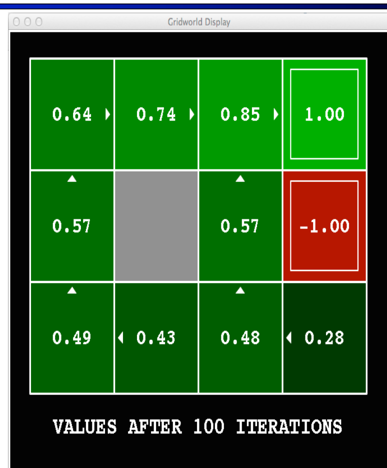
k=11

21

k=12

22

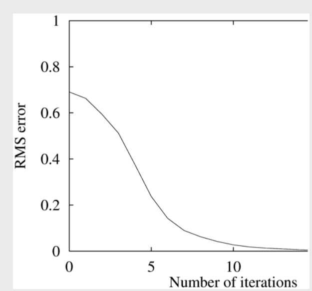
k=100

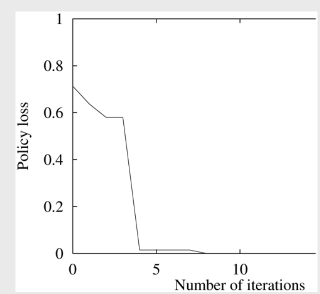


23

Value Function and Policy

- Each step takes O(|A| |S| |S|) time.
- Number of iterations required is polynomial in |S|, |A|, 1/(1-gamma)

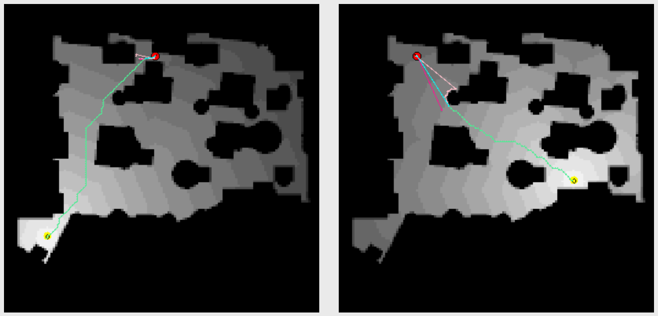




24

Value Iteration for Motion Planning

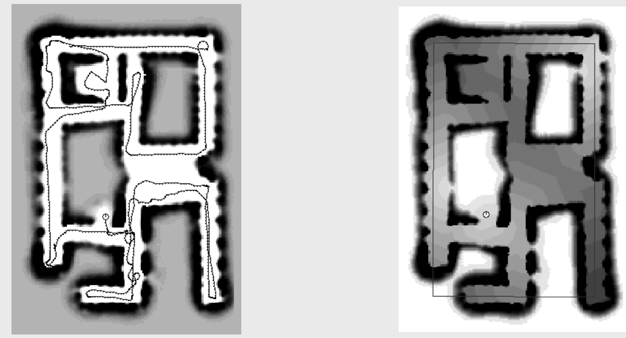
(assumes knowledge of robot's location)



25

Frontier-based Exploration

- Every unknown location is a target point.



26

POMDPs

- In POMDPs we apply the very same idea as in MDPs.
- Since the **state is not observable**, the agent has to **make its decisions based on the belief state** which is a posterior distribution over states.
- For finite horizon problems, the resulting value functions are piecewise linear and convex.
- In each iteration the **number of linear constraints grows exponentially**.
- Full fledged POMDPs have only been applied to very small state spaces with small numbers of possible observations and actions.
- **Approximate solutions are becoming more and more capable**.

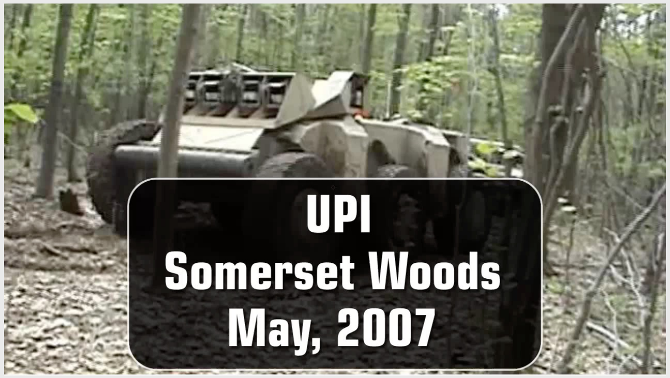
27

CSE 571
Inverse Optimal Control
(Inverse Reinforcement Learning)

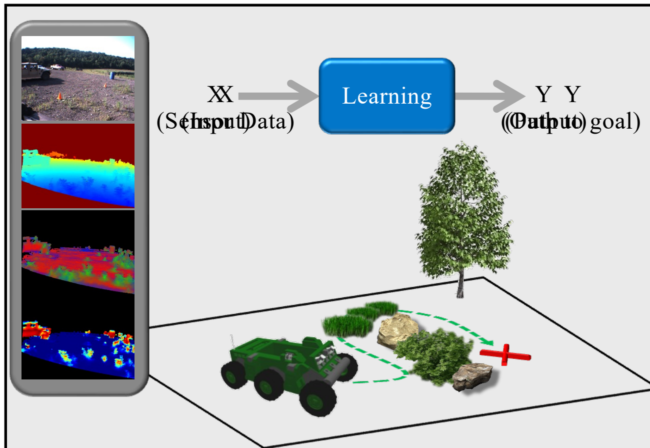
Many slides by Drew Bagnell
Carnegie Mellon University

28

Autonomous Navigation

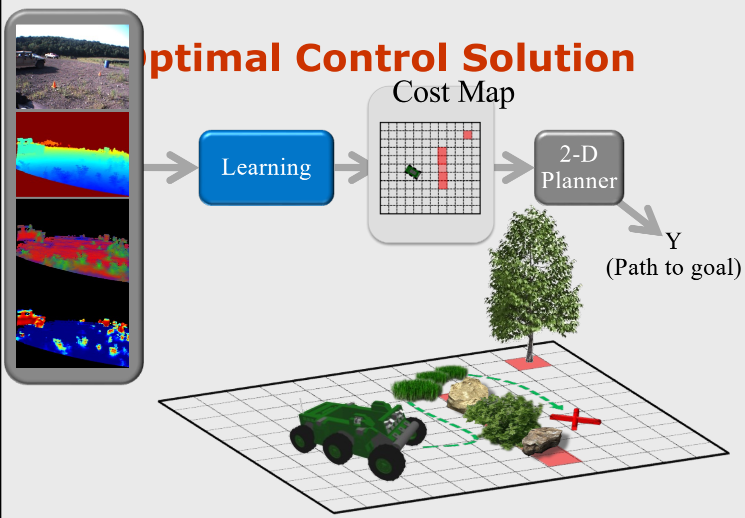


29



30

ptimal Control Solution



31

Mode 1: Training example

32

Mode 1: Training example

33

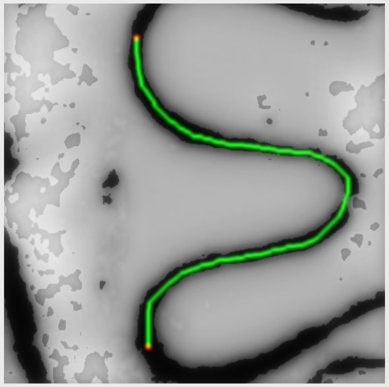
Mode 1: Learned behavior

34

Mode 1: Learned behavior

35

Mode 1: Learned cost map



36

Mode 2: Training example

37

Mode 2: Training example

38

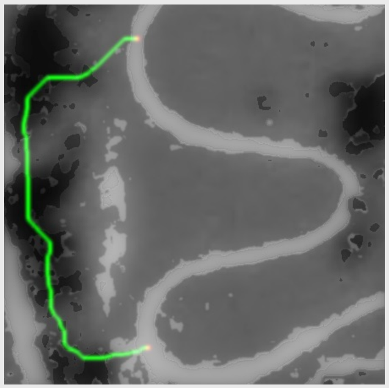
Mode 2: Learned behavior

39

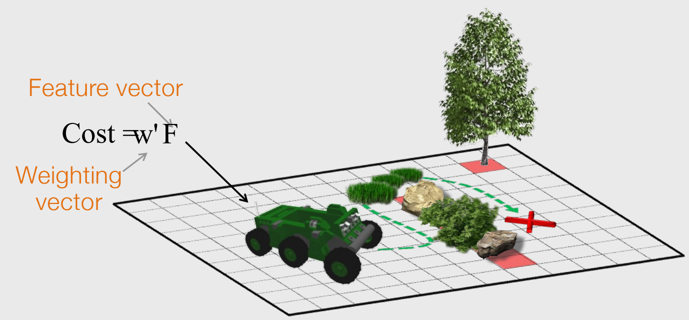
Mode 2: Learned behavior

40

Mode 2: Learned cost map

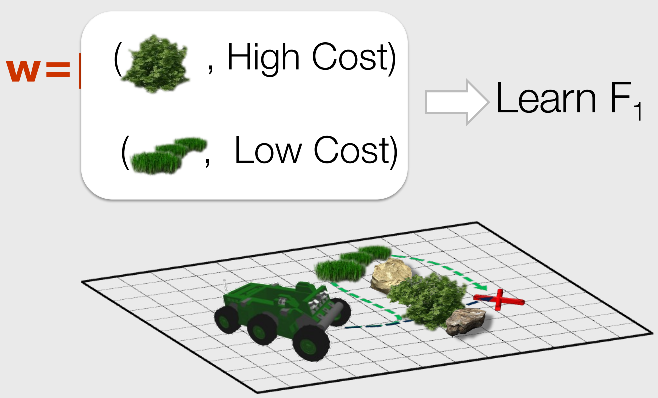


41

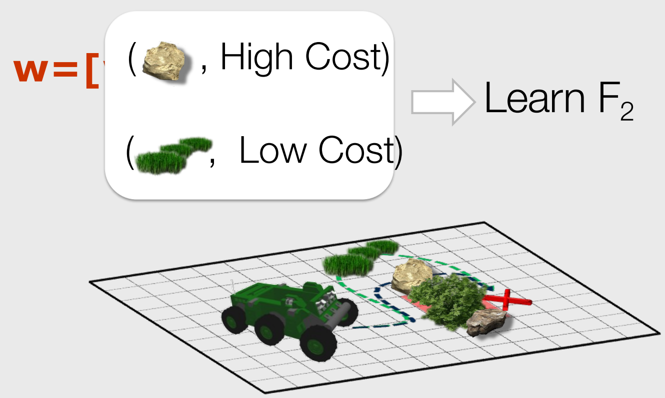


Ratliff, Bagnell, Zinkevich 2005
Ratliff, Bradley, Bagnell, Chestnutt, NIPS 2006
Silver, Bagnell, Stentz, RSS 2008

42



43



44

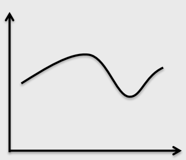
45

Learning Manipulation Preferences

- ***Input:*** Human demonstrations of preferred behavior (e.g., moving a cup of water upright without spilling)
- ***Output:*** Learned cost function that results in trajectories satisfying user preferences

46

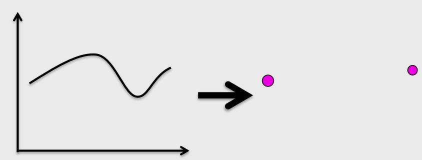
46



Demonstration(s)

47

47

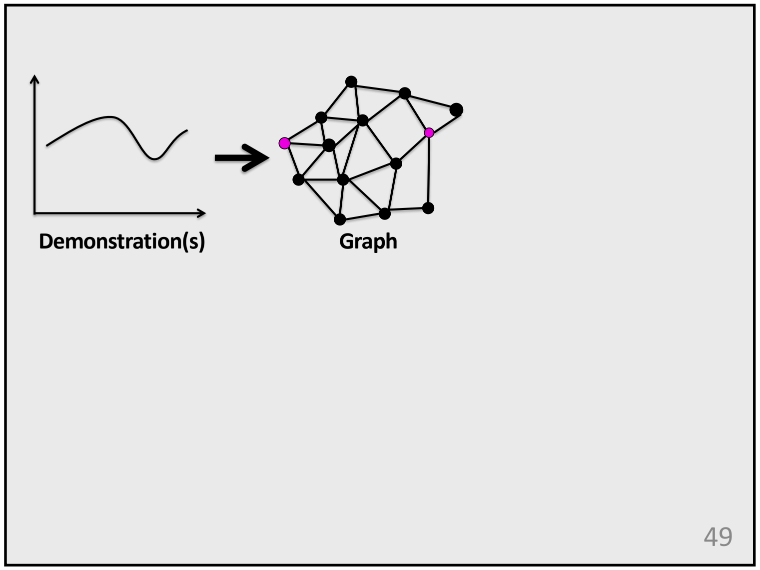


Demonstration(s)

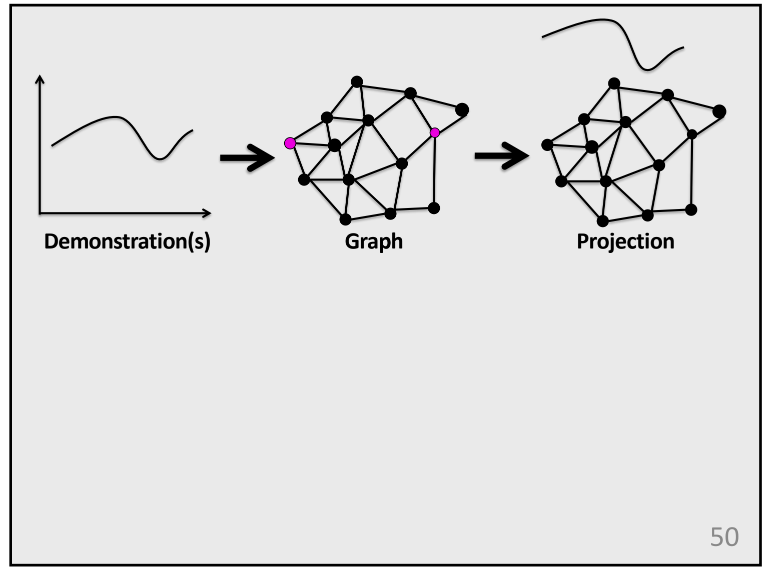
Graph

48

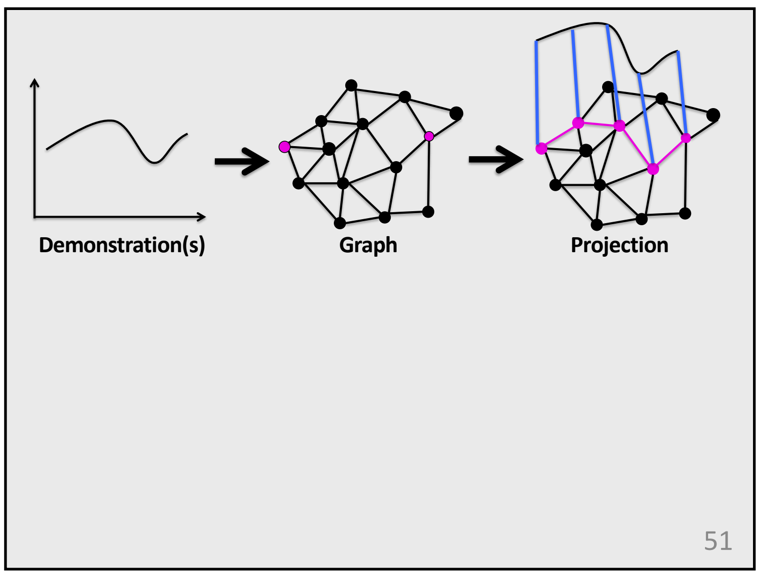
48


Demonstration(s)
Graph
49

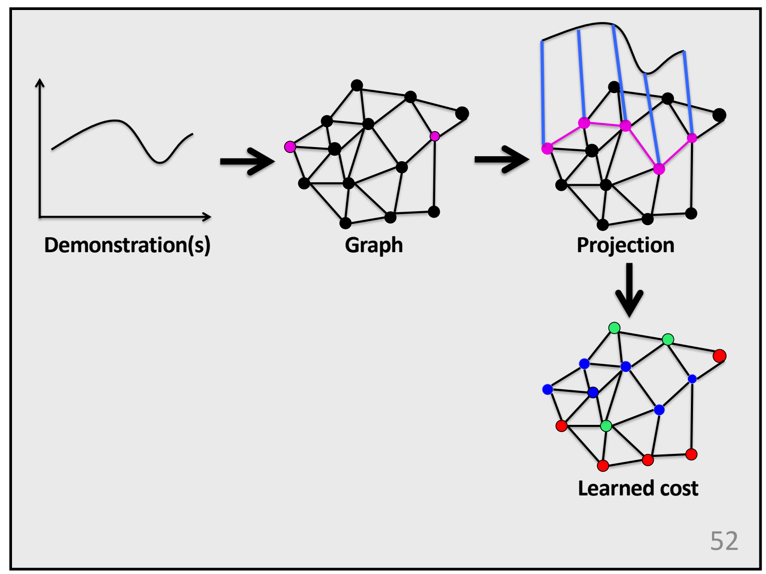
49


Demonstration(s)
Graph
Projection
50

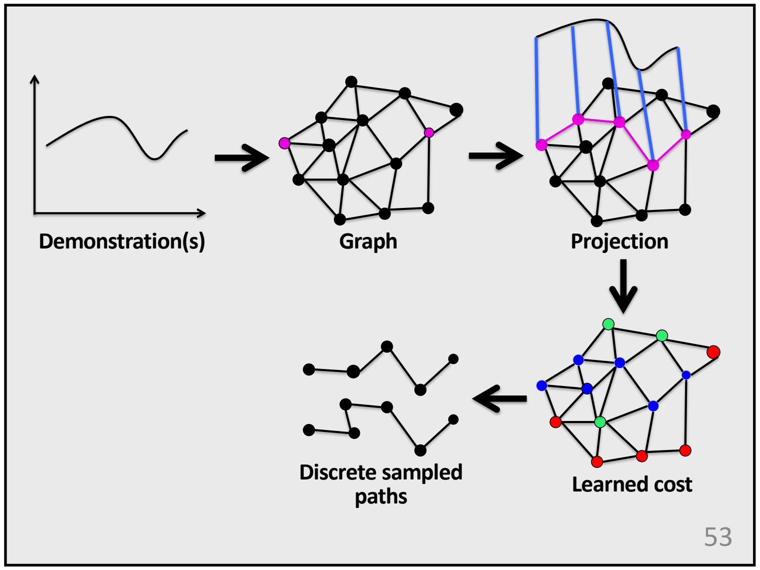
50


Demonstration(s)
Graph
Projection
51

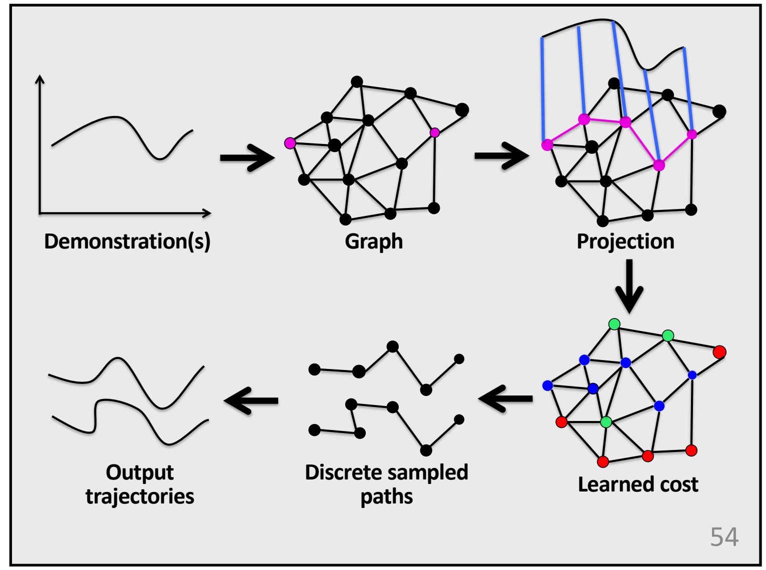
51


Demonstration(s)
Graph
Projection
Learned cost
52

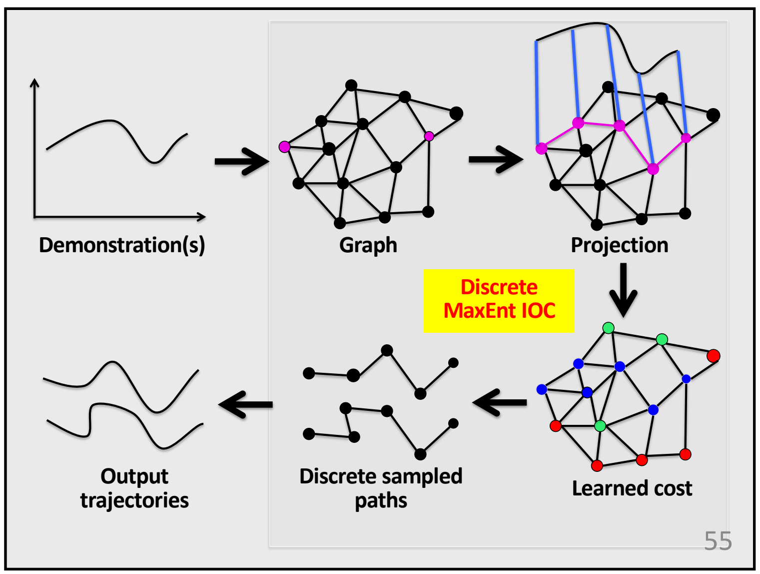
52


Demonstration(s)
Graph
Projection
Discrete sampled paths
Learned cost
53

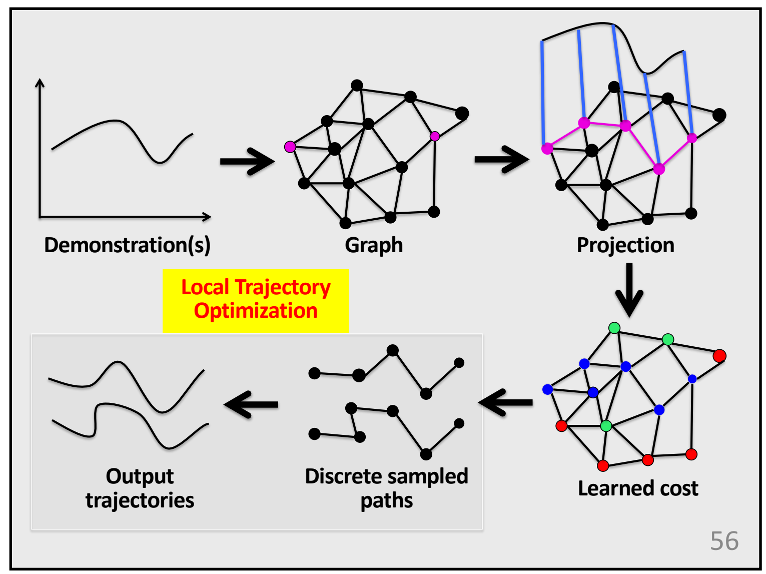
53


Demonstration(s)
Graph
Projection
Output trajectories
Discrete sampled paths
Learned cost
54

54


Demonstration(s)
Graph
Projection
Discrete MaxEnt IOC
Output trajectories
Discrete sampled paths
Learned cost
55

55


Demonstration(s)
Graph
Projection
Local Trajectory Optimization
Output trajectories
Discrete sampled paths
Learned cost
56

56

Setup

- ***Binary*** state-dependent features (~95)
  - Histograms of distances to objects
  - Histograms of end-effector orientation
  - Object specific features (electronic vs non-electronic)
  - Approach direction w.r.t goal
- **Task**
  - Hold cup upright while not moving above electronics

57

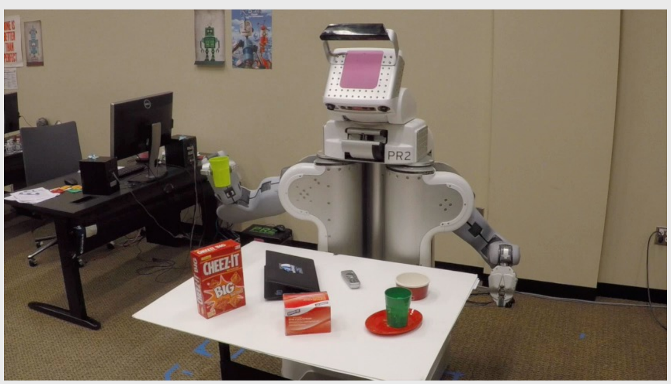
57

Laptop task: Demonstration
(Not part of training set)

58

58

Laptop task: LTO + Smooth random path



59

59

Readings

- Max-Ent IRL (Ziebart, Bagnell): http://www.cs.cmu.edu/~bziebart/
- CIOC (Levine) http://graphics.stanford.edu/projects/cioc/cioc.pdf
- Manipulation (Byravan/Fox): https://rse-lab.cs.washington.edu/papers/graph-based-IOC-ijcai-2015.pdf
- Imitation learning (Ermon): https://cs.stanford.edu/~ermon/
- Human/manipulation (Dragan): https://people.eecs.berkeley.edu/~anca/research.html

60

60