Inverse Optimal Control II:
Battle of the Senior Graduate Students

Drew Bagnell
16-899 Adaptive Control and Reinforcement Learning

THE ROBOTICS INSTITUTE

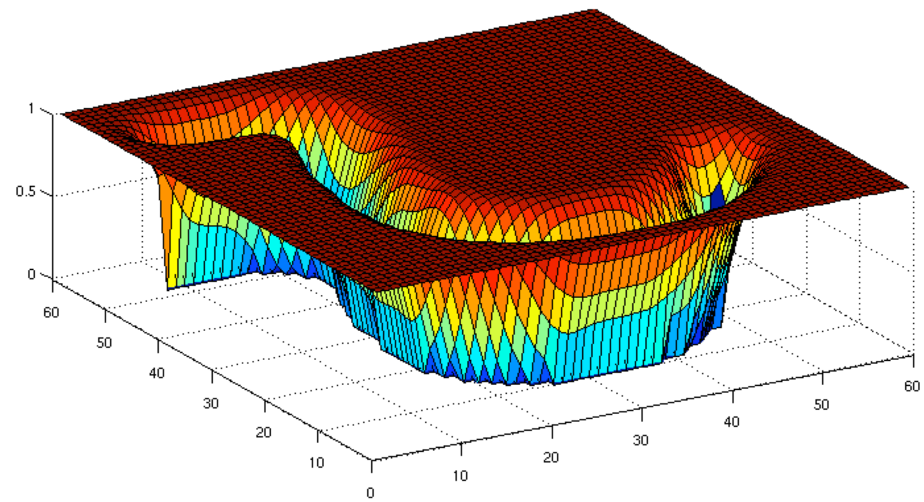
Recovering the True Reward Function is ill-posed…

Recovering the True Reward Function is ill-posed…

Instead we design a loss function to measure performance of our ability to imitate using a cost function

Measuring performance and Margin/Loss-augmentation

- Before planning through the learned cost maps we lower the cost of cells proportionally to their loss
- It becomes more likely that we plan through bad areas
- We train on slightly harder problems so that we do better in practice



- Forces the final solution to be correct by a margin.
- Formally, this step places the algorithm under the margin-maximization framework

Formulate as Convex Optimization Problem: MMP

min Complexity(cost)

$$\forall i \; \boxed{\text{cost of expert plan}}_i \leq \boxed{\text{cost of arbitrary plan}}_i - \boxed{\text{margin}}_i$$

Subgradient optimization

$$c(w) = \sum_{i=1}^{N} \left(w^T f_i(\xi_i) - \min_{\xi \in C_i} (w^T f_i(\xi) - l_i(\xi)) \right) + \frac{\lambda}{2} \|w\|^2$$

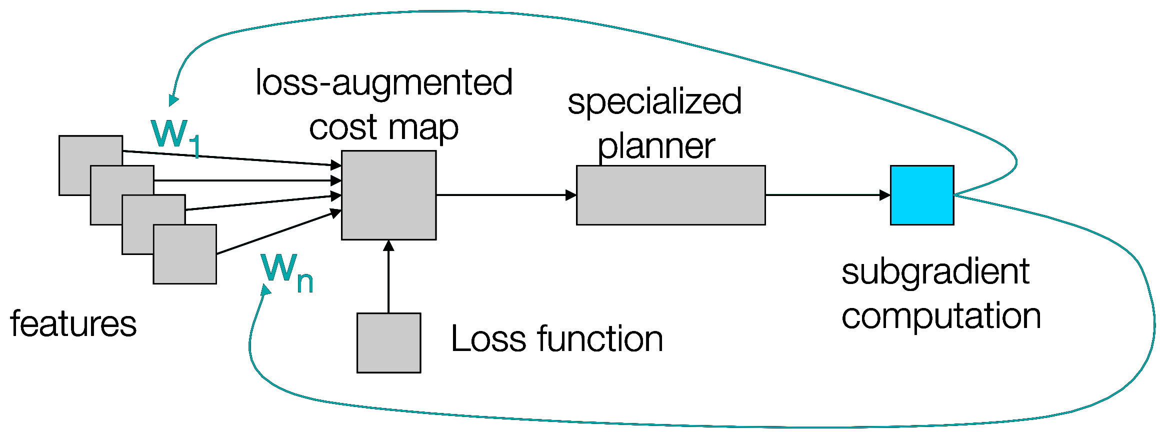
- All terms of our cost function are differentiable except that with the min
- Find a subgradient by evaluating the gradient at the minimizer

$$\frac{\partial}{\partial w_j} \left(w^T f_i(\xi^*) - l_i(\xi^*) \right)$$

where $\xi^* = \arg\min_{\xi \in C_i} (w^T f_i(\xi) - l_i(\xi))$

- Resulting sub-gradient: $\sum_{i=1}^{N} (f_j(\xi_j) - f_j(\xi^*)) + \lambda w_j$

Maximum margin planning algorithm



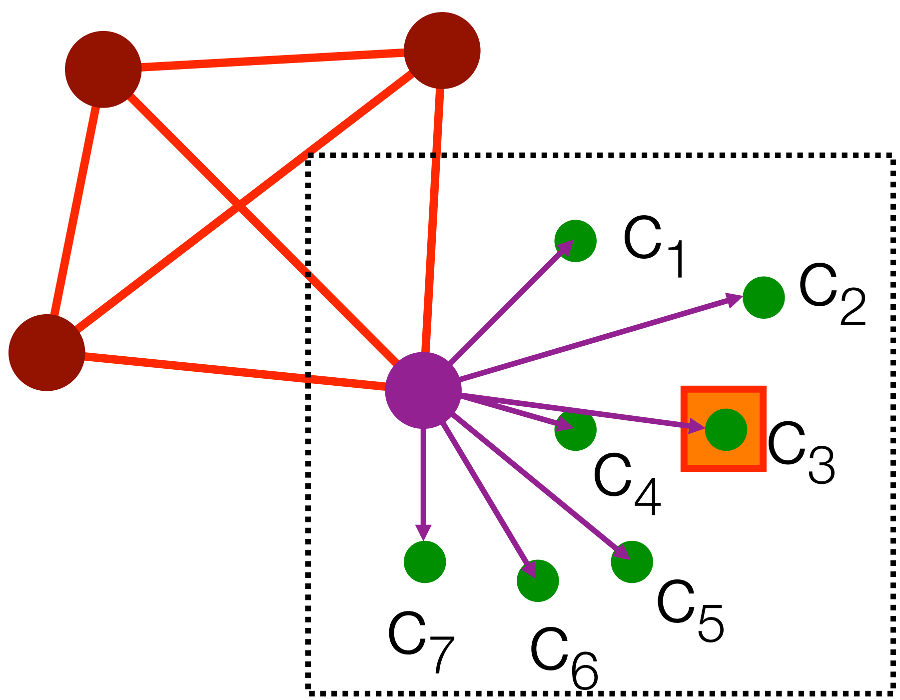
- Incremental subgradient method: $$R(w) = \frac{1}{N}\sum_{i=1}^{N}\left(w^T F_i \mu_i - \min_{\mu \in \mathcal{G}_i}(w^T F_i - l_i^T)\mu\right) + \frac{\lambda}{2}\|w\|^2$$
- Until convergence do

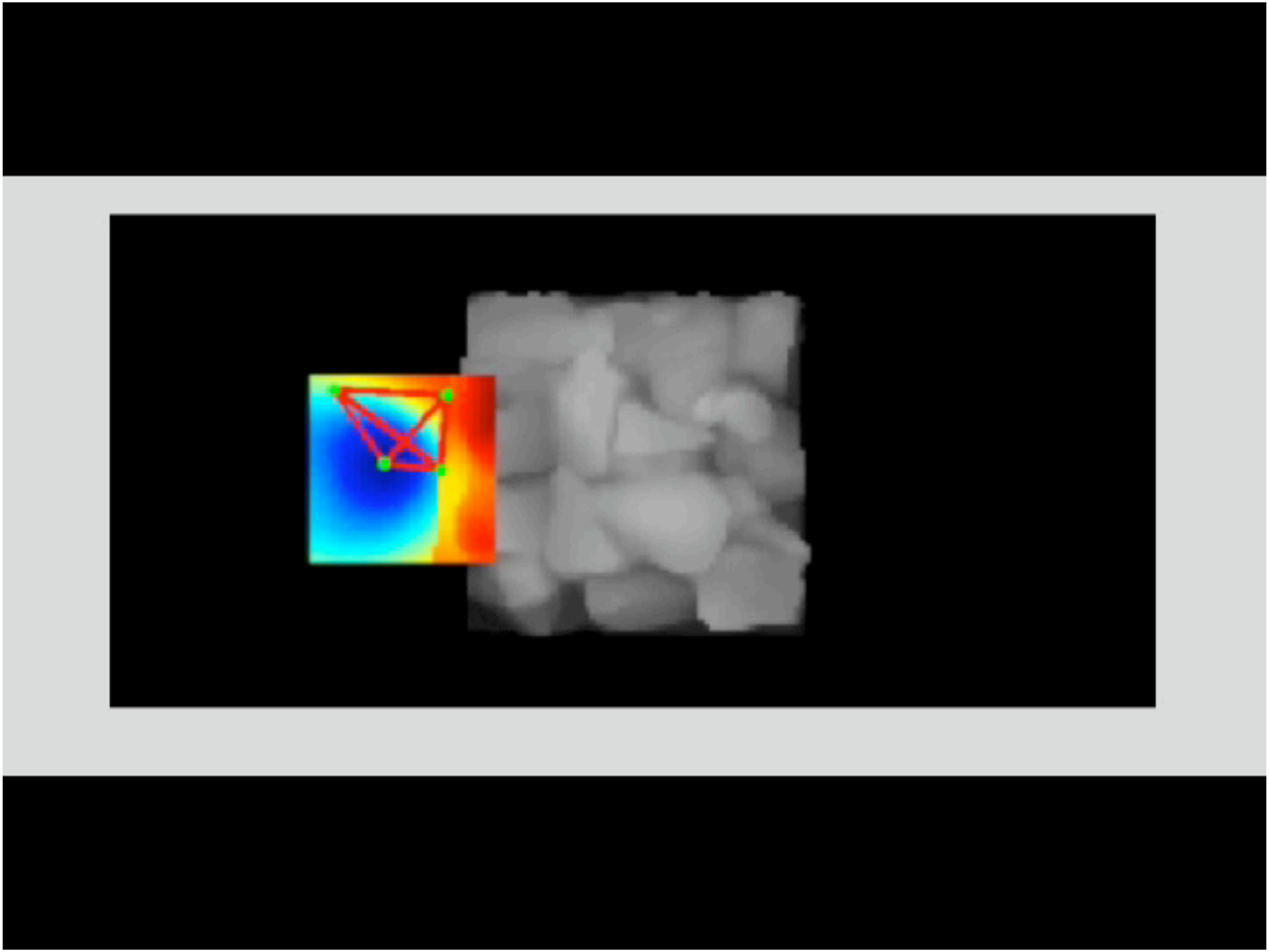
Subiteration cycle

- For each example
  - Create loss-augmented cost map
  - Run planner to find minimum cost path between endpoints of this example
  - Compare the resulting (loss-augmented) cost to that of the desired path to compute gradient
  - Update w (optional: project w onto constraints)

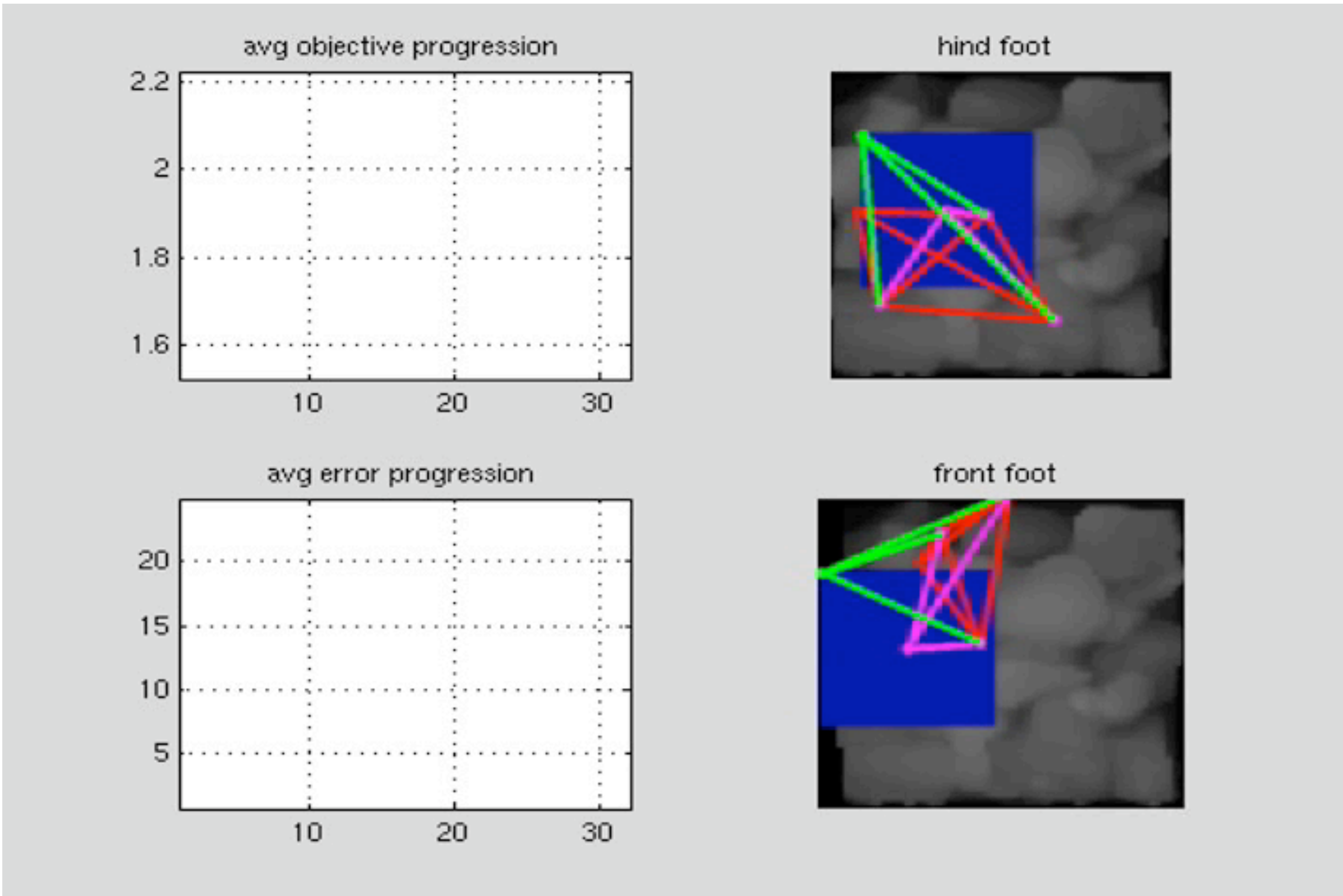
Mimicking footsteps

Where should we place the foot next?

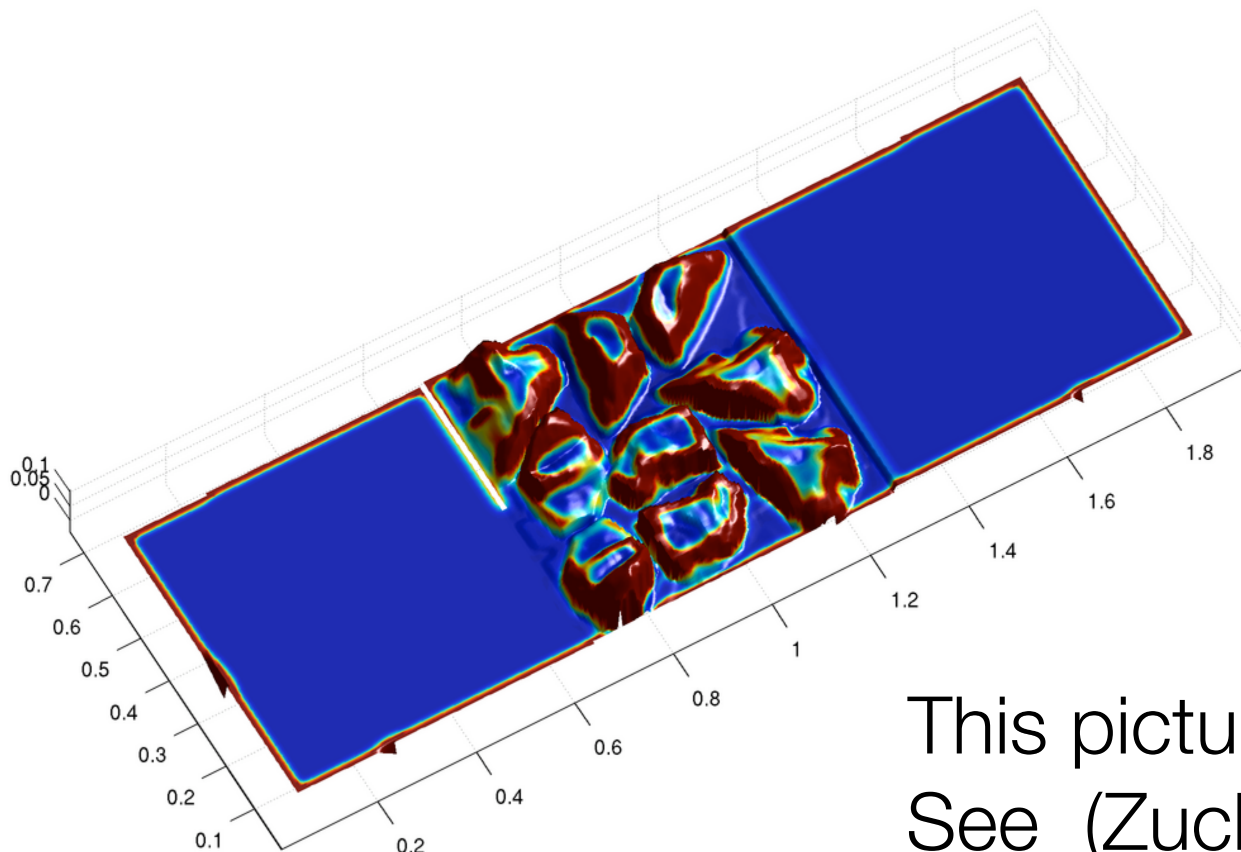




LittleDog Training Time : Enforcing Optimality

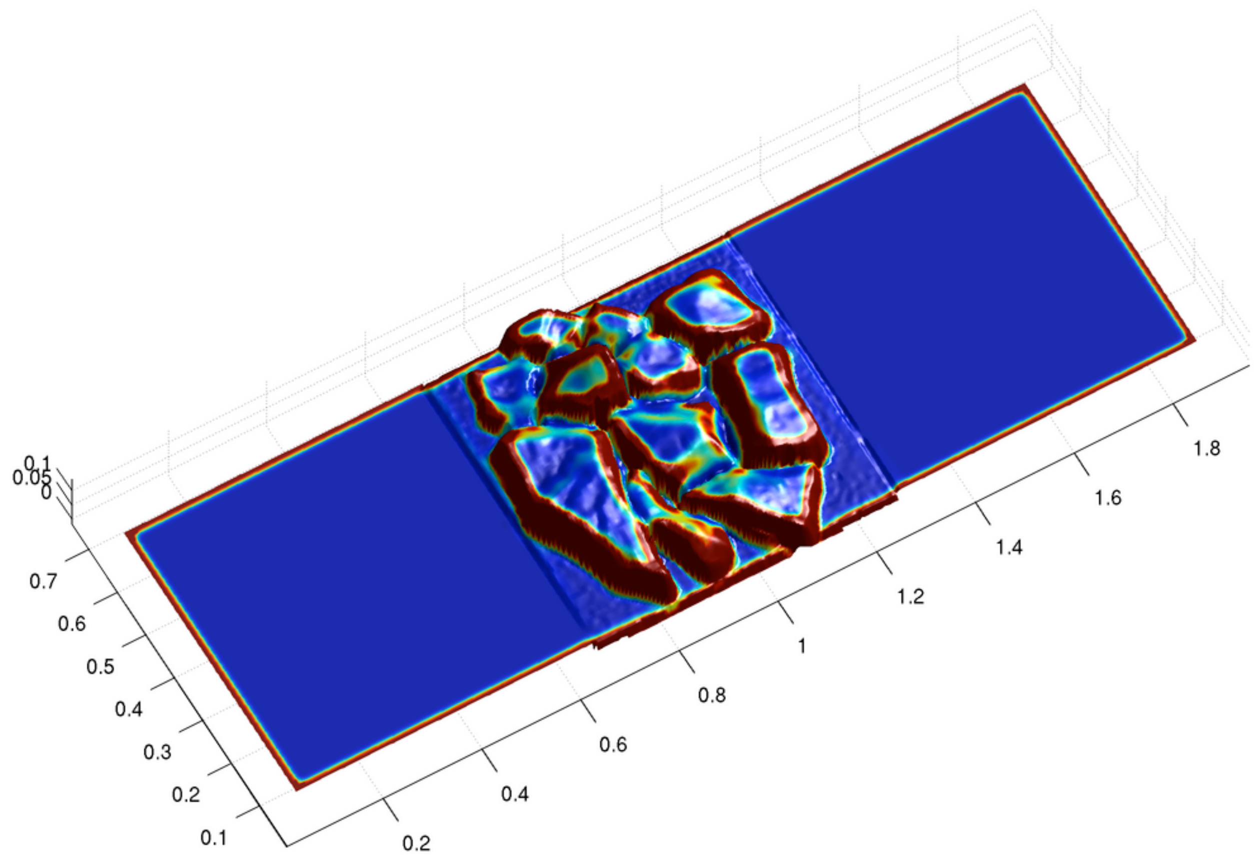


Learned Cost Function Examples

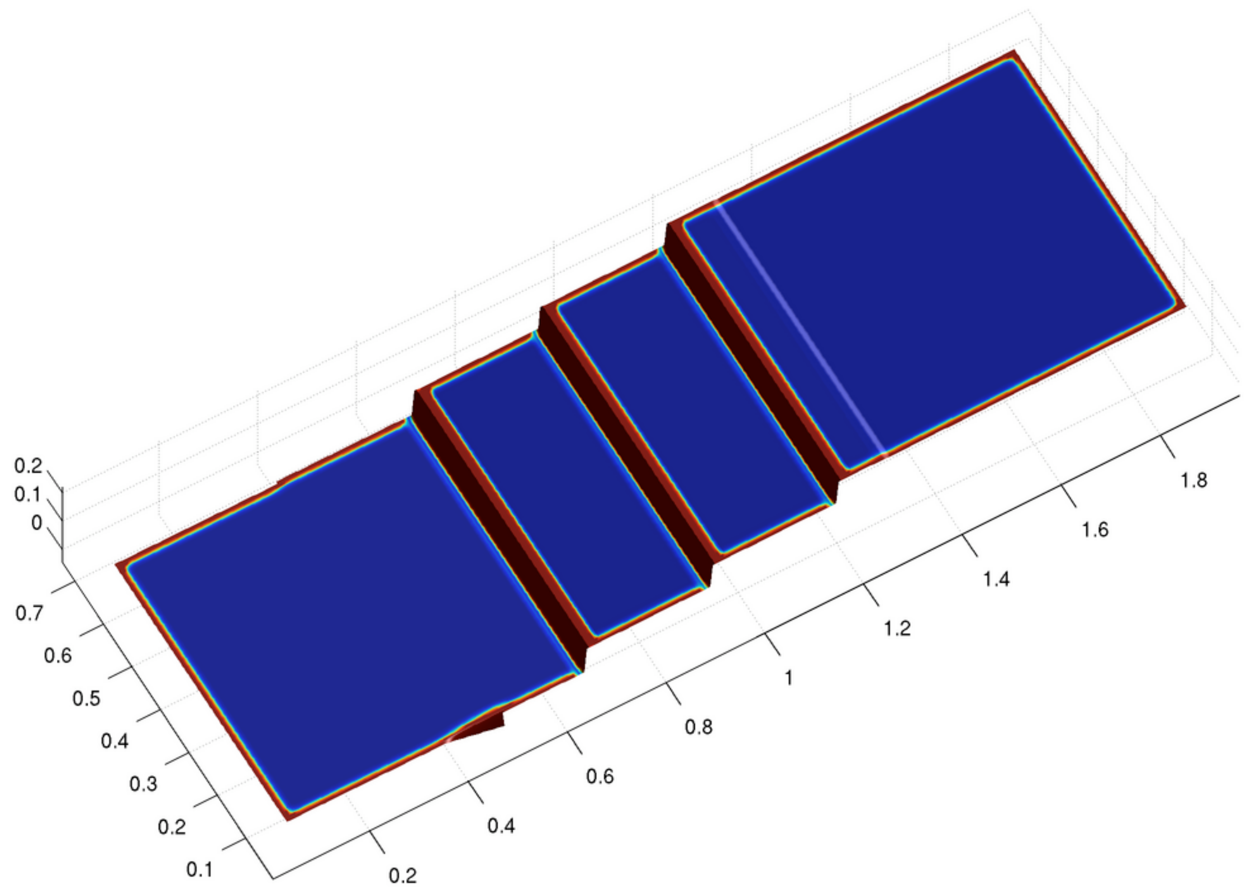


This picture is a lie. See (Zucker09) for what made **these** pretty pictures

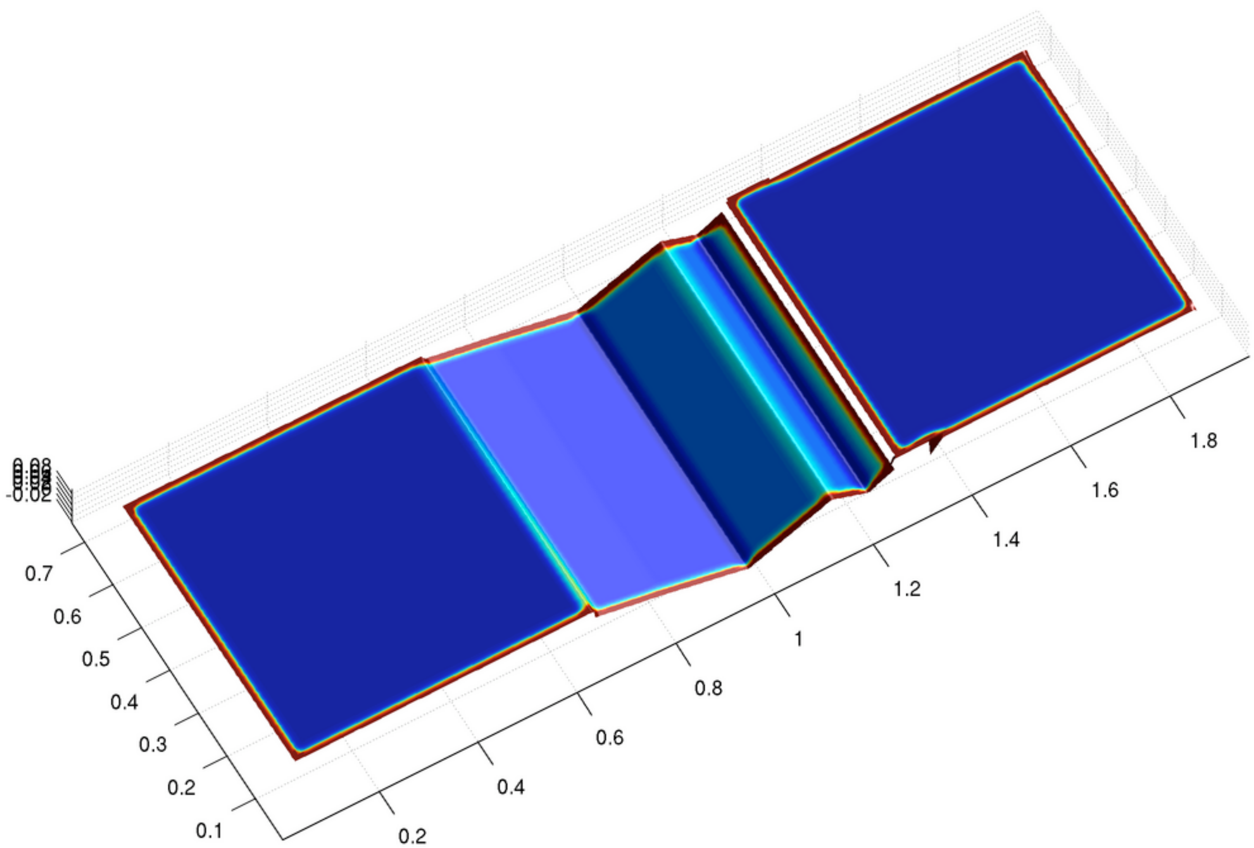
Learned Cost Function Examples



Learned Cost Function Examples



Learned Cost Function Examples



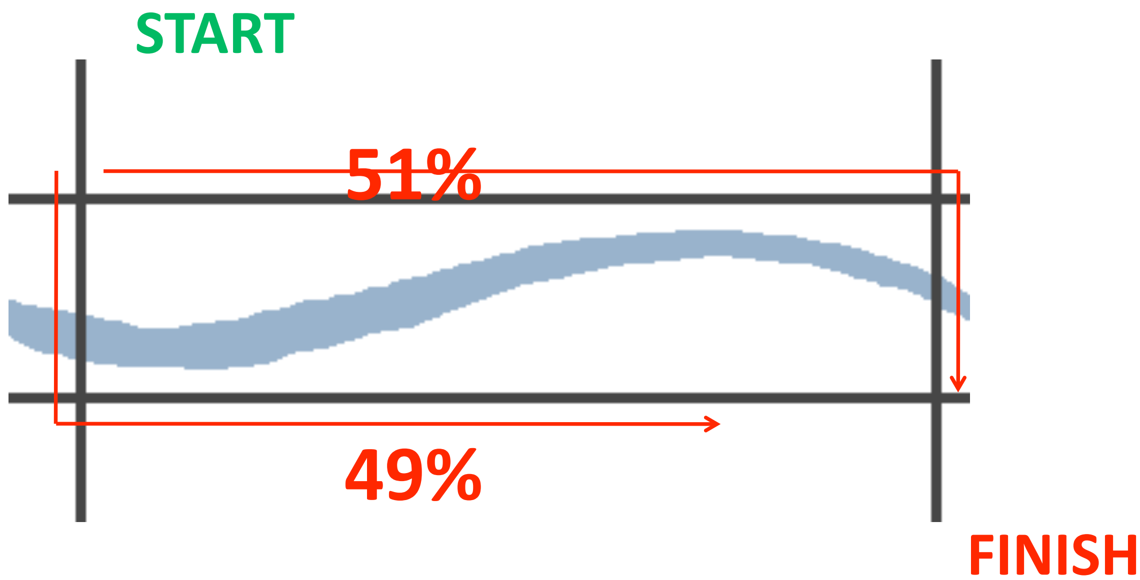
Update the Good!

How/Why can people behave sub-optimally?

- What’s the consequence?

Maximum Margin Planning

- What if two paths are "about equal?"



- Unique optimality assumption **violated!**

Update the Bad. ☹

An Alternate approach: Feature Matching

(Abbeel and Ng 2004)

Demonstrated Behavior

Bridges crossed: **3**

Miles of interstate: **20.7**

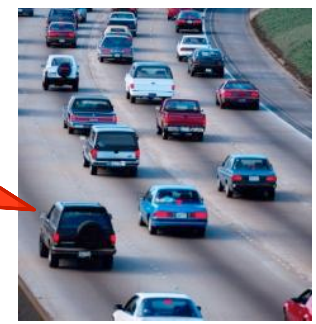
Stoplights: **10**

Equal Performance in MDP

Model Behavior (Expectation)

Bridges crossed: **3**

Miles of interstate: **20.7**



Stoplights: **10**

An Alternate approach:
Feature Matching (Can we prove it?)

(Abbeel and Ng 2004)

Demonstrated Behavior

Bridges crossed: **3**

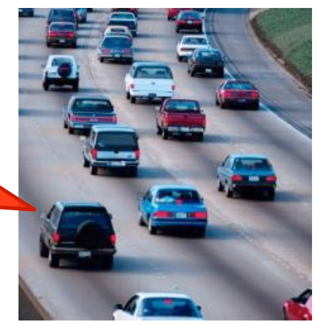
Miles of interstate: **20.7**

Stoplights: **10**

Model Behavior (Expectation)

Bridges crossed: **3**

Miles of interstate: **20.7**



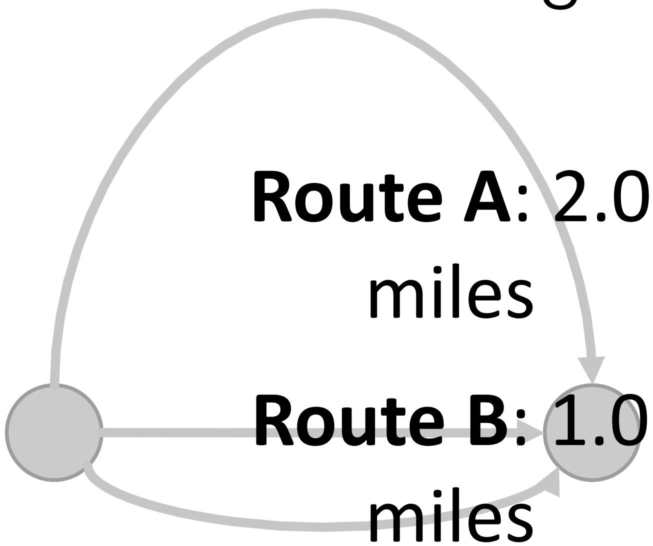
Stoplig hts: **10**

Equal Performan ce in MDP

Hmmmm….. Ambiguity again

- There is **no reward function and no optimal policy** that matches that matches almost all behavior
- There are **infinitely many** stochastic behaviors (policies or mixtures of policies) that can match feature counts….
- **How can we possibly pick a good one?**

Feature Matching



Route C: 1.1 miles

Route C demonstrated by driver.

- Abbeel and Ng match features using a mixture of reward functions/policies:

90% Route B ($\theta > 0$)

10% Route A ($\theta < 0$)

Zero probability for demonstrated route!?

Maximum Entropy Inverse Optimal Control

Maximizing the **entropy** over paths:

$$\max \mathbf{H}(\mathbf{P}_{\zeta})$$

While matching feature counts (and being a probability distribution):

$$\sum_{\zeta} P(\zeta)\, f_{\zeta} = f_{\mathbf{dem}}$$

$$\sum_{\zeta} P(\zeta) = 1$$

Maximum Entropy Inverse Optimal Control

Maximizing the **entropy** over paths:

$$\max \mathbf{H}(\mathbf{P}_{\zeta})$$

As uniform as possible

While matching feature counts (and being a probability distribution):

$$\sum_{\zeta} P(\zeta)\, f_{\zeta} = f_{\mathbf{dem}}$$

$$\sum_{\zeta} P(\zeta) = 1$$

Maximum Entropy Inverse Optimal Control

Maximizing the **entropy** over paths:

$$\max \mathbf{H(P_{\zeta})}$$

As uniform as possible

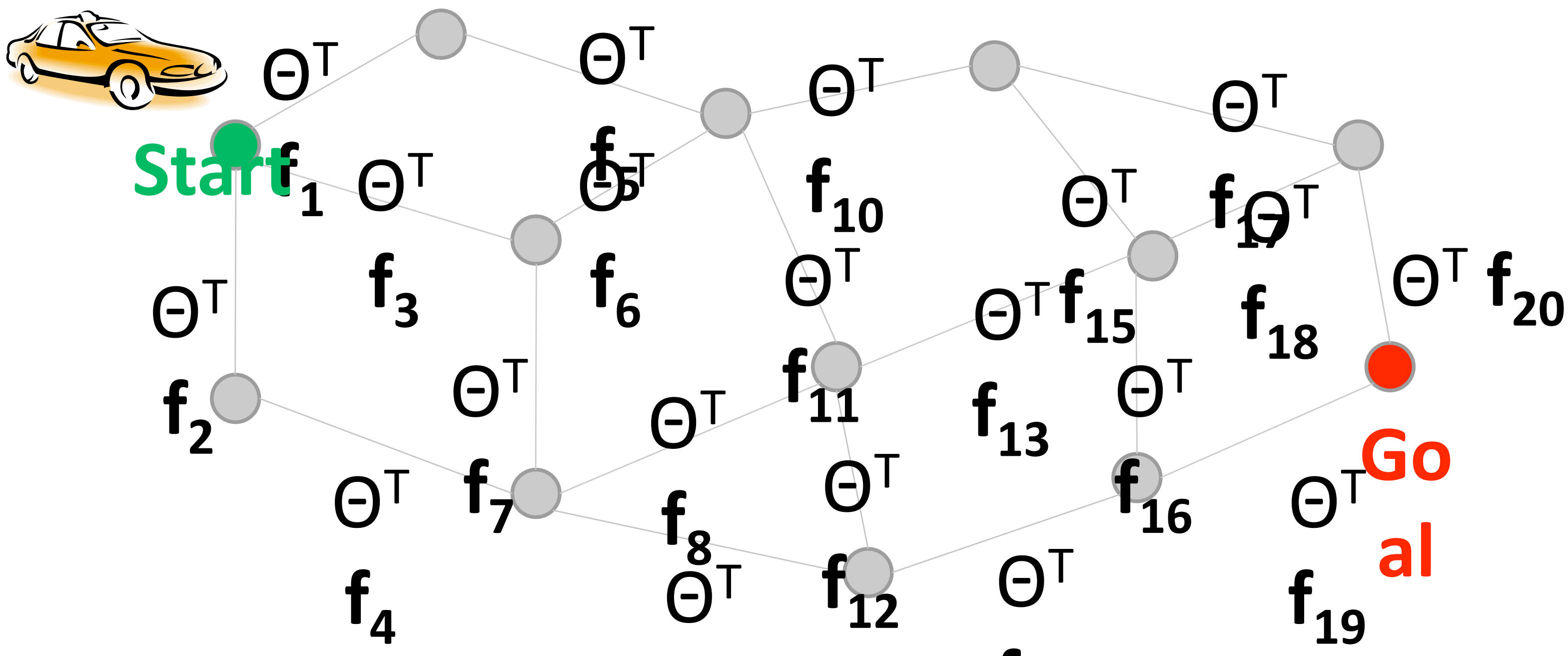
While matching feature counts (and being a probability distribution):

$$\sum_{\zeta} P(\zeta)\, f_{\zeta} = f_{\mathbf{dem}}$$

$$\sum_{\zeta} P(\zeta) = 1$$

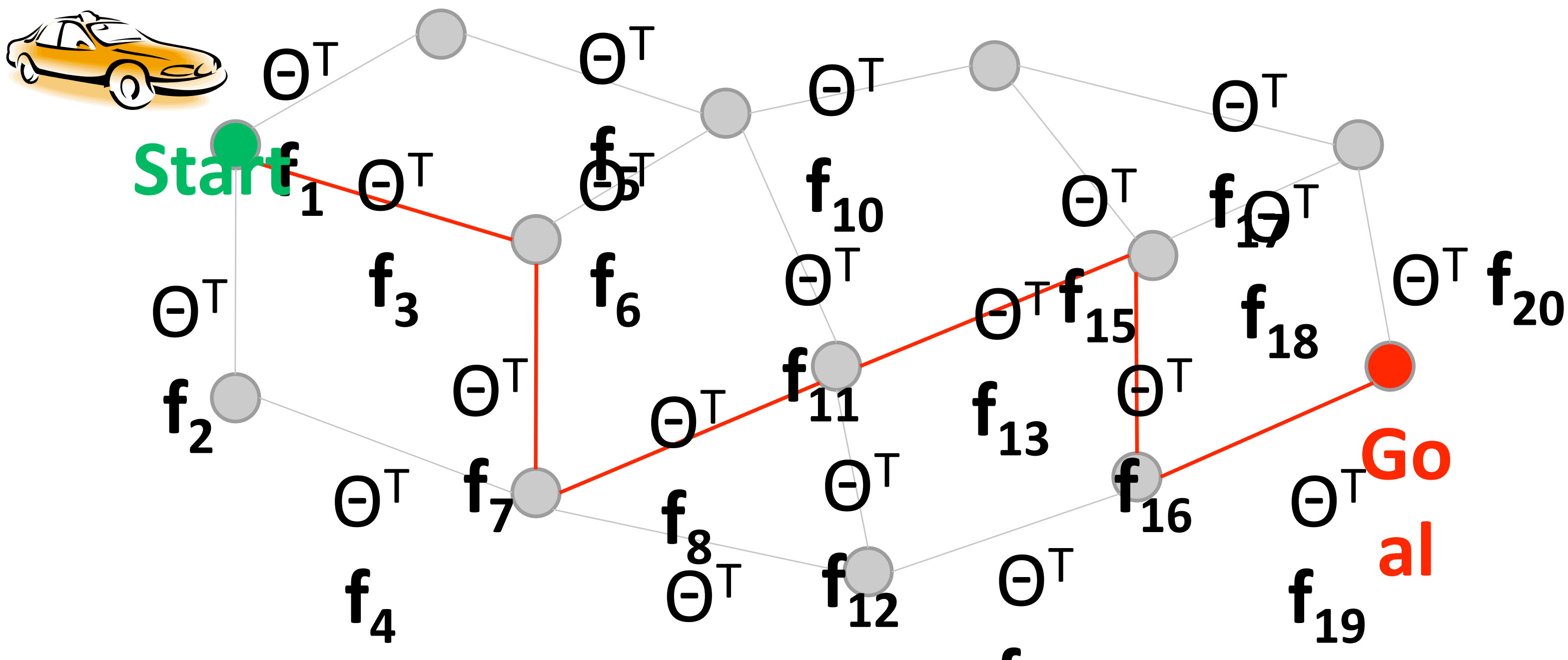
Performance guarantee

Maximum Entropy Inverse Optimal Control



Roads have **unknown costs** (linear in features)

Maximum Entropy Inverse Optimal Control



Roads have **unknown costs** (linear in features)

Paths have **unknown costs** (sum of road costs)

Path probability based on **unknown cost**

What Probability Distribution?

The Dual:

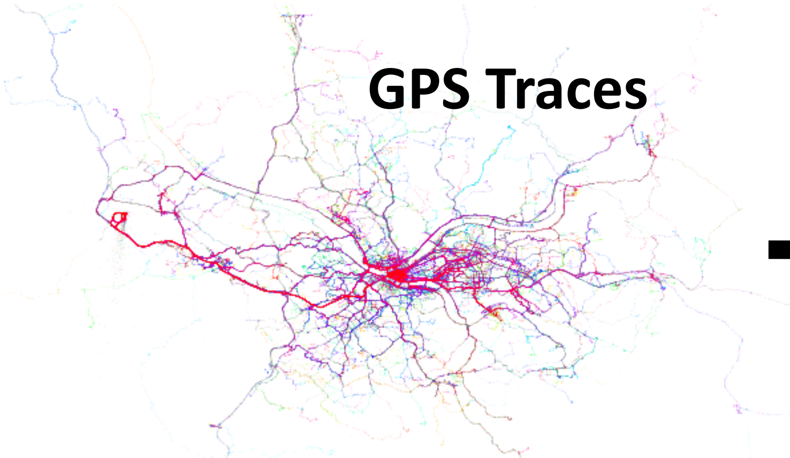
$$P(path|\theta) = \frac{e^{-cost(path|\theta)}}{\sum_{path'} e^{-cost(path'|\theta)}}$$

Strong Preference for Low Cost Paths

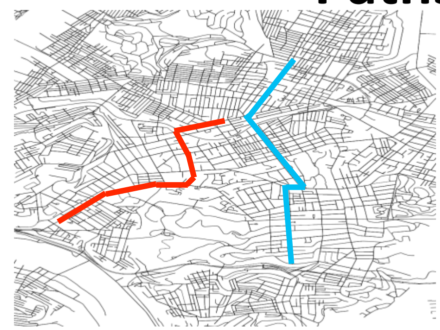
Equal Cost Paths Equally Probable

Decision Making Data:

GPS Traces



Paths



Model:

$$P(path|\theta) = \frac{e^{-cost(path|\theta)}}{\sum_{path'} e^{-cost(path'|\theta)}}$$

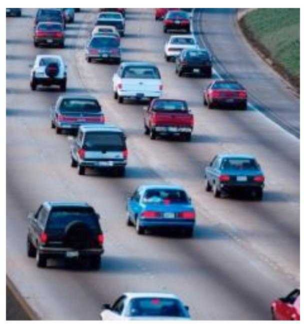
Choose cost parameters (θ) that best explain demonstrated paths

Learning from Demonstration

Demonstrated Behavior

Bridges crossed: **3**

Miles of interstate: **20.7**



Stoplights: **10**

Model Behavior (Expectation)

Bridges crossed: **?**

Miles of interstate: **?**

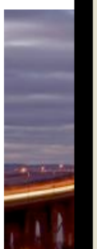
Cost Weight: 5.0

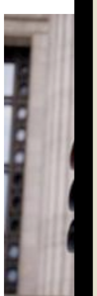
Cost Weight: 3.0

Stoplights: **?**

Learning from Demonstration

Demonstrated Behavior





Inference

$$P(\text{path } \zeta) = \frac{e^{-\text{cost}(\zeta|\theta)}}{\sum_{\text{path } \zeta} e^{-\text{cost}(\zeta|\theta)}}$$

$$\sum_{\text{path } \zeta} P(\text{path } \zeta) f_{\zeta}$$

Dynamic Programming

$O(e^{\text{length}}) \rightarrow O(\text{length})$

Model Behavior (Expectation)

Bridges crossed: **?**

Miles of interstate: **?**

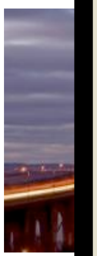
Cost Weight: 5.0

Stoplights: **?**

Cost Weight: 3.0

Learning from Demonstration

Demonstrated Behavior



Inference

$$P(\text{path } \zeta) = \frac{e^{-\text{cost}(\zeta|\theta)}}{\sum_{\text{path } \zeta} e^{-\text{cost}(\zeta|\theta)}}$$

$$\sum_{\text{path } \zeta} P(\text{path } \zeta)\, f_{\zeta}$$

Dynamic Programming

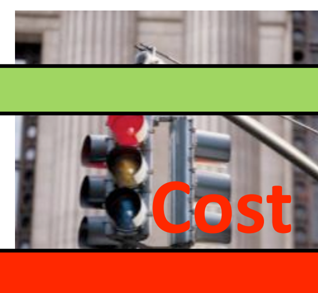
$O(e^{\text{length}}) \rightarrow O(\text{length})$

Model Behavior (Expectation)

Bridges crossed: **4.7**

Miles of interstate: **16.2**

Cost Weight: 5.0



Stoplights: **7.4**

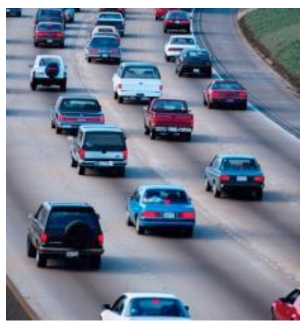
Cost Weight: 3.0

Learning from Demonstration

<u>Demonstrated Behavior</u>

Bridges crossed: **3**

Miles of interstate: **20.7**



Stoplights: **10**

<u>Model Behavior (Expectation)</u>

Bridges crossed: **4.7**

+1.7

Miles of interstate: **16.2**

-4.5

Cost Weight: 5.0

Cost Weight: 3.0

Stoplights: **7.4**

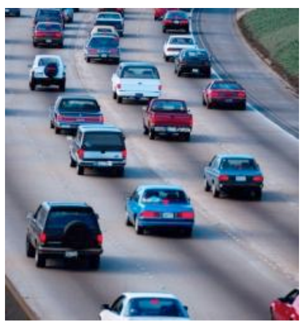
-2.6

Learning from Demonstration

Demonstrated Behavior

Bridges crossed: **3**

Miles of interstate: **20.7**



Stoplights: **10**

Model Behavior (Expectation)

Bridges crossed: **4.7**

Miles of interstate: **16.2**

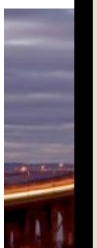
7.2 Cost Weight: 5.0

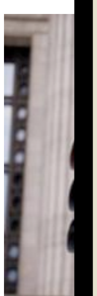
1.1 Cost Weight:

Stoplights: **7.4**

Learning from Demonstration

Demonstrated Behavior





Inference

$$P(\text{path } \zeta) = \frac{e^{-cost(\zeta|\theta)}}{\sum_{\text{path } \zeta} e^{-cost(\zeta|\theta)}}$$

$$\sum_{\text{path } \zeta} P(\text{path } \zeta)\, f_\zeta$$

i

Dynamic Programming

$O(e^{length}) \rightarrow O(length)$

Model Behavior (Expectation)

Bridges crossed: **4.7**

Miles of interstate: **16.2**

7.2

Cost Weight: 5.0

Stoplights: **7.4**

1.1

Cost Weight:

Learning from Demonstration

Demonstrated Behavior

Bridges crossed: 3

Miles of interstate: 20.7

Stoplights: **10**

Model Behavior (Expectation)

Bridges crossed: **4.7**

Miles of interstate: 16.2

7.2

Cost Weight: 5.0

1.1

Cost Weight:

Stoplights: **7.4**

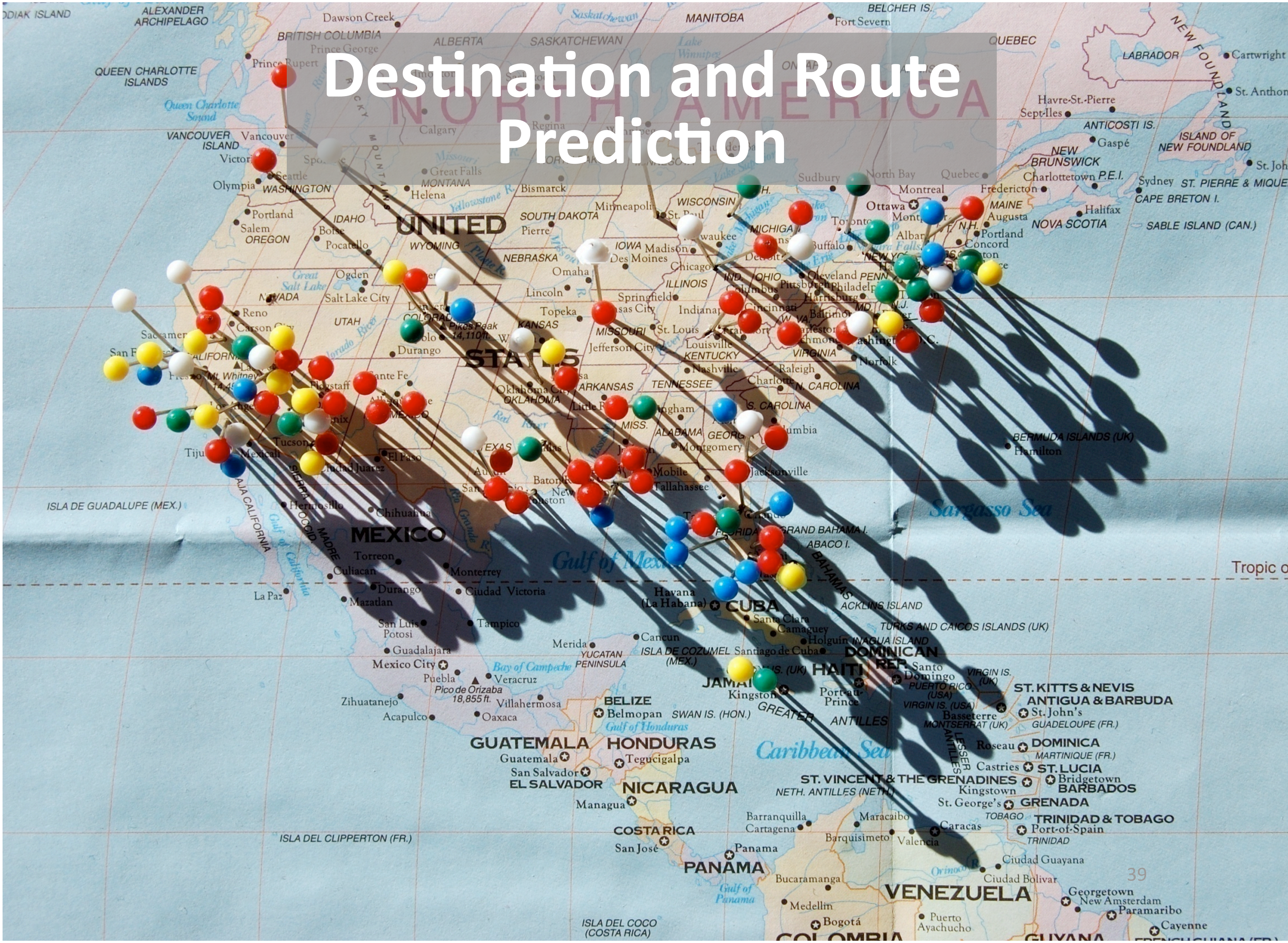
Optimal Solution!

(Convexity)

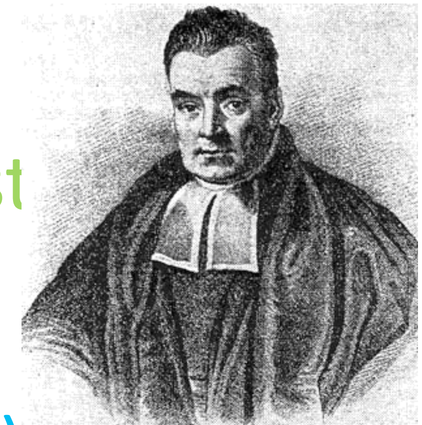
How does the dynamic program work?

- “Soft” value iteration

Destination and Route Prediction



Unknown Destination

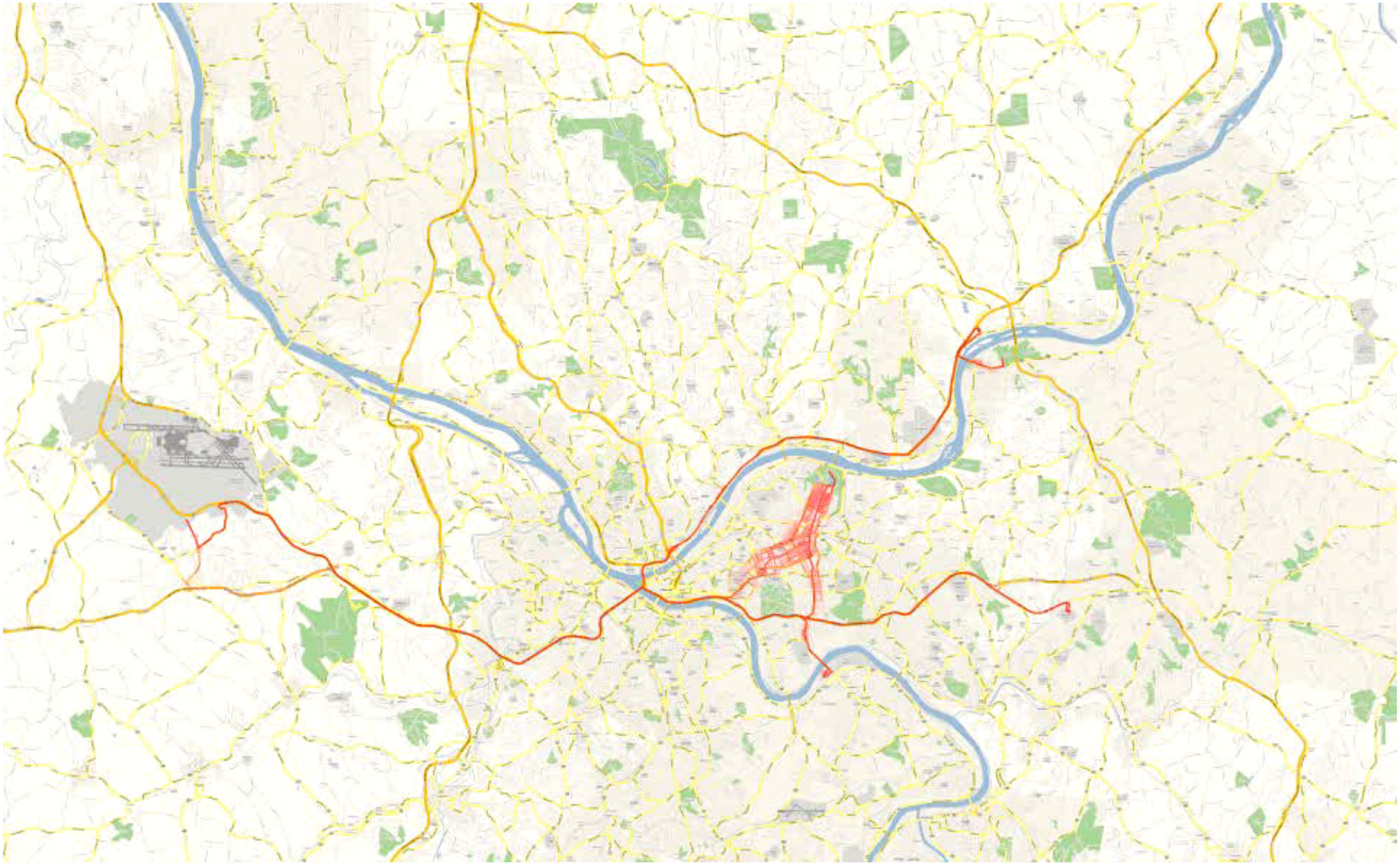


MaxEnt Model: P(path|dest

Bayes Rule:

$$P(\text{dest}|\text{path}) = \frac{P(\text{path}|\text{dest})\ P(\text{dest})}{\sum_{\text{dest}'} P(\text{path}|\text{dest}')\ P(\text{dest}')}$$

Prior Distribution: P(dest)



Pedestrian Modeling

(Brian Ziebart, Kevin Peterson, Martial Hebert)