Imitation Learning for Autonomous Navigation in Complex Natural Terrain

David Silver
Carnegie Mellon University
Robotics Institute
Outline

• Introduction: Coupling Perception and Planning
• Static Imitation Learning from Prior Perceptual Data
• Dynamic Imitation Learning from Onboard Perceptual Data
• Engineering vs. Imitation Learning
• Future Work
Mobile Navigation Systems

Sense

Act

Plan
Mobile Navigation Systems

Sensor Data → Perception System → Planning System → Control System → Environment
Mobile Robot Perception

Raw Sensor Data → Perception System → Local World Model and Description
Mobile Robot Perception
Mobile Robot Planning

Preferences and Connectivity of States

Planning System

Sequence of Actions

low cost high cost
Mobile Robot Planning
Coupling Perception And Planning

Sensor Data → Perception System → Planning System → Control System → Environment

Local World Model and Description

Preferences and Connectivity of States
Coupling Perception and Planning

Local World Model and Description

Preferences and Connectivity of States

?
Metrics

• All motion and path planners make use of a metric
  – Defines which plan is optimal
  – For some planners, metric is only defined implicitly (e.g. shortest distance)

• Defining this metric helps to define a robot’s behavior
Metrics in Practice

• Defining a metric is necessary but not sufficient for planning
  – A planning system must be able to score candidate plans against the metric

• A scalar reward/utility/cost function is required that implies the metric
  – For some metrics, the cost function is well defined (shortest distance, minimum energy)
  – For other metrics (or combinations of metrics), the definition is less clear
Costing

• Costing is the process of mapping from an action through a state space to a scalar cost.

• Defining costs that imply the proper metric is difficult. Defining a function that maps from (features of) state/action inputs to the appropriate cost is much harder.

• As the terrain becomes less structured, the definition of cost becomes more important.
Outline

• Introduction: Coupling Perception and Planning
• Static Imitation Learning from Prior Perceptual Data
• Dynamic Imitation Learning from Onboard Perceptual Data
• Engineering vs. Imitation Learning
• Future Work
Long Range Navigation Using Prior Data

- Complex, cluttered environments
  - trees, bushes, hills, high slopes, ditches, washes, trails, etc.
- Minimally constrained areas of operation
- Large waypoint spacing (100m – 1km)
- Courses from 4 – 40 km
Overhead Data

• Ideal vantage point for providing prior environmental knowledge

• Numerous available data sources at varying cost
  – Aerial or Satellite Imagery
  – Digital Elevation Maps
  – Aerial LiDAR

Quickbird imagery courtesy of Digital Globe, Inc.
Prior Data

• In general, long range navigation does not require prior knowledge; *Efficient* long range navigation does

• [Silver et al., 2006] 38% increase in speed, 75% decrease in safety intervention incidents when using prior data
Path Planning from Overhead Data

• For use in both online and offline planning, available overhead data must be translated into a cost map.
Overhead Cost Maps

- Raw Data is converted into features
- Features are converted into scalar costs
Choosing Cost Functions

- Manually construct a cost function
  [Vandapel et al., 2003], [Silver et al., 2006]
  - Repeated manual parameter tuning in a potentially high dimensional space
  - Non-intuitive relationship between features and cost
  - Must be repeated for all available feature sets
Cost Features

- Convert raw features into an intermediate feature space, and then the new ‘cost’ features to scalar costs
  - Example: Convert imagery into a classification map

- Advantages
  - Potentially lower dimensional and more intuitive space
  - Only need to determine cost function once

- Disadvantages
  - Additional complexity
  - Conversion may involve loss of information
  - Conversion itself must be recomputed or retrained for each feature set
Expert Examples

• Cost tuning essentially tries to make planned paths look reasonable
• People are good at indentifying the correct behavior, but not necessarily the cost function that encodes that behavior
• So learn a cost function from examples of correct behavior
Imitation Learning

• Large body of previous work for vehicle control [Pomerleau, 1989]
  – Learns a mapping from features of a state to actions
  – Does not generalize to long range planning

• For long range planning, need to learn a mapping from features of a state to costs, and then plan on these costs
  – Essentially, learn to interpret perception data for a specific planning system
Maximum Margin Planning (MMP)

- [Ratliff et al., 2006]
- Enforce a constraint on cost functions: the cost of an example path must be less than the cost of all other paths with the same endpoints
- Convert to an optimization problem, and add a margin

\[
C(\text{ExamplePath}) \leq \min_{\text{PlannedPath}} C(\text{PlannedPath}) + L(\text{ExamplePath}, \text{PlannedPath})
\]
Functional Gradient Descent

\[ \nabla O_s[C] = \delta_s(\text{ExamplePath}) - \delta_s(\text{PlannedPath}) \]

- Approximate the gradient via regression, and take a step in the negative direction.
- Sum gradient approximations, and loop until convergence.
$C = K + R(\mathcal{U}^{-1}) + R(\mathcal{U}^{-1})$
Example
Different Metrics
Suboptimal Examples

- Expert examples are rarely *exactly* optimal
  - Noise in the path can negatively affect generalization
- Solution: Enforce constraint within a small corridor around original example
Unachievable Examples

• Some example path constraints can never be met by any consistent cost function
  – No solution exists in the hypothesis space for a given planner
  – These examples can also effect generalization

• Solution: perform a balanced regression
  – Drives the gradient of unachievable examples towards zero
Outline

• Introduction: Coupling Perception and Planning
• Static Imitation Learning from Prior Perceptual Data
• Dynamic Imitation Learning from Onboard Perceptual Data
• Engineering vs. Imitation Learning
• Future Work
Onboard Perceptual Data
Onboard Perceptual Data
Onboard Perceptual Data
Onboard Perceptual Data
Onboard Costing
Previous Work in Online Costing

- Feature space compression
  - Transform features to a lower dimension, more intuitive space
- Online learning via proprioception
  - Learn to predict robot’s interaction with terrain
- Physical simulation
  - Build a model for robot’s interaction with terrain
- Manual engineering
  - Parameter tuning
Static vs. Dynamic Imitation Learning

- Training data collection
- Dynamics
- Unobserved regions of terrain
- More complex planning systems
Collecting Training Data
Training Data

• Expert must actually tele-operate through the example

• The training set consists not only of the path driven, but all raw sensor data collected during the traverse
  – Allows for re-running perception offline at any time
Discretize by Time
Discretize by Time
Unobserved Terrain
Unobserved Terrain
Replanning

• Replanning the example path within a small corridor has previously demonstrated benefits (smoothing)

• If the corridor is ‘opened up’ in unknown terrain, the example will adhere to the correct behavior in said terrain
Unobserved Terrain
Multiple Planning Systems

• MMP learns the correct cost function for a **specific** planning system
  – Iterates until the chosen planner sufficiently reproduces examples

• Therefore, costs must be learned with respect to the planning system that actually determines the commanded action
  – Otherwise, no guarantee that correct behavior will be reproduced in practice
Local/Global Planning
Local vs. Global Planner

• The local planner makes the final planning decision

• Learning a cost function with respect to the global planner is not sufficient
  – During training, the planner may appear to avoid certain obstacles, but in practice it will drive right over them
Local vs. Global Planner

$|GP1| + \text{Cost(Rock)} + \epsilon = |GP2|$

$|LP1| + \text{Cost(Rock)} + \epsilon = |GP2|$

$|LP2| > |GP2|$

$|LP2| > |LP1| + \text{Cost(Rock)} + \epsilon$
Local Planner Resolution

- The global planner does not consider kinematic or dynamic constraints
  - Can reproduce any path through the state space
- The Local planner does consider the kinematics of the robot
  - Depending on planner resolution, it may not be possible to sufficiently match an expert’s example behavior
Local Planner Resolution
Local Planner Resolution

• Insufficient resolution can lead to modifying costs without proper cause
  – Can result in over-fitting
• Insufficient resolution can result in the opposite of an ‘unachievable’ case discussed earlier
• Full solution would require expert to choose only amongst planner actions
Heuristic Approximation
Expert Plan vs. Expert Behavior

• Technically, we should try and learn from an expert’s plan at each timestep
  – Far too tedious to ever collect in practice
• Instead, we use an expert’s final end behavior, and assume it is sufficiently close to the expert’s plan
• This creates the potential for poor or inconsistent examples
Expert Intent Change
Inconsistent Examples

• It is possible for an example behavior to be outside of system’s hypothesis space
  – No consistent cost function will cause the planner to match the example

• Can exist for numerous reasons
  – Plan/Behavior mismatch
  – Insufficient Planner resolution
  – Insufficient Perception resolution or descriptiveness
  – Expert Error
Filtering Inconsistent Examples

• An expert’s behavior should be consistent over a single tele-operation
• Therefore, individual timesteps that are inconsistent can be filtered
  – Learn a cost function for a single tele-operation
  – Use final difference in planned and example cost to determine outliers
  – Use a more powerful regressor than planned for actual system
Choice of Regressor

- In an online system, the computational complexity of a cost evaluation is a consideration
  - Learned cost function requires $O(N)$ regressor evaluations after $N$ iterations
- Using linear regressors, cost function can be condensed to single evaluation
- The limited expressiveness of linear regressors can be overcome by adding occasional ‘boosted’ features
Boosted Features
Outline

• Introduction: Coupling Perception and Planning

• Static Imitation Learning from Prior Perceptual Data

• Dynamic Imitation Learning from Onboard Perceptual Data

• Engineering vs. Imitation Learning

• Future Work
UPI Field Tests

• Large areas of operation
  – 10 - 200 km²
  – 2 weeks of sponsor monitored autonomous experiments and validation

• Variety of Overhead Data available
  – Satellite Imagery, Aerial Lidar
  – Multiple subsets are used to simulate different levels of available prior data

• Learned prior maps used during 5 field tests from May 2007 – June 2008
  – Engineered maps used at previous field tests
Engineered vs. Learned Maps
Overhead Results

- Learned prior maps were used during more than 600 km of sponsor monitored autonomous traverse
- Learned prior maps produced quicker, and without human parameter tuning
  - Human involvement reduced from days to hours
  - No expert knowledge of autonomy system required
- In 25 km of head to head experiments, autonomy with learned prior maps was safer and faster than with engineered maps [Silver et al., 2008]
Engineered Perception Costing

- Crusher’s engineered cost function was continually redesigned and retuned for over 3 years
  - ~450 committed changes in version control logs
  - Easily hundreds of hours of design time
Learned Perception Costing

• Training data consists of less than 1 hour (real time) of data
  – Collected piecemeal over several months at multiple test sites
  – Training data re-used whenever input perception system is modified
Perception Comparison

• Crusher was run through multiple courses varying only the cost function amongst 4 options
  – Engineered, learned for global planner, learned for local planner, learned for local planner with initial heuristic approximation

• Over 150km of total comparison runs
Perception Comparison Results

- Cost function learned for the global planner resulted in overly aggressive behavior compared to the engineered function.
- Cost function learned for the local planner resulted in similar behavior.
- Adding a heuristic approximation increased the aggressiveness without compromising safety [Silver et al., 2009]
Outline

• Introduction: Coupling Perception and Planning
• Static Imitation Learning from Prior Perceptual Data
• Dynamic Imitation Learning from Onboard Perceptual Data
• Engineering vs. Imitation Learning
• Future Work
Alternate Example Constraints
Active Learning
Costs over Actions
Costs over Actions
Acknowledgements

- DARPA
- UPI Team