Big Question

What does society want to do with robots?

Informally, what is the “killer app” of robotics?

Note: not the “killer robot app”
Big Question

What does society want to do with robots?

Problems:

- Society has little idea what robots can do
- Programming robots requires significant technical expertise
- Chicken-egg problem -> scifi notions disparate from reality
One Possible Answer

- Program robots from human demonstration

Research Problems:

- **Algorithms**: learn policy from data (exper., exprl., guidance, etc)
- **Data collection**: “lifelong” human supervision and robot performance
- **Usability** by humans; interruptions
Course Structure

- Group project for entire class
- Cover research papers in robot learning and object manipulation
- Cover 2-3 papers per class
- Student paper presentations (20 mins max, minus questions)
- Everyone must summarize each paper
Group Project

- Massive-scale learning from demonstration
- Implement in ROS; do Create tutorial:
  - http://code.google.com/p/brown-ros-pkg/
- Learn three tasks from demonstration
  - Create robot soccer
  - Nao magneto assembly
  - PR2 intern challenge
- One learning alg, one infrastructure box
- Human subjects study
WHY ROBOT LEARNING?

“Any controller that has been learned could have been programmed in less time and performed better”

- anonymous big name in robotics
A GOAL FOR ROBOTICS

Collaborators for human endeavors

• Robot → tool for user productivity

• path of least resistance for doing physical tasks

• user-developed applications through learning

• critical path tasks?

• societal utility?
“technology exponentials”, e.g., Moore’s Law; mentioned by Brooks and others
Currently "Personal Robotics Revolution"
DISTINCT CHALLENGES

Other exponentials predicated on deterministic manipulation of state

Enables “write local, run global” development

Variance and uncertainty in tasks, users, and environments limits this model for robotics
WHY ROBOT LEARNING?

When does learning make sense compared to teleop or manual programming?

• Discovery of controllers difficult to phrase analytically

• Enabling non-technical users to express robot controllers
WHY ROBOT LEARNING?

Either way, expression of computing required: FSMs, MDPs, objective functions, likelihoods etc.

• Discovery of controllers difficult to phrase analytically
  
  Trained users fluent in expressing models of computing

• Enabling non-technical users to express robot controllers
  
  Non-technical users might not gain such programming fluency
BROADER VIEW

Casted in FSMs, learn as a whole:

1) Policies for states/primitives

2) Transitions between states

3) State pre/postconditions

INFLAMMATORY STATEMENT:
Computational models learned for robots are significantly more limited than hand-coded models b/c learning focuses on individual issues above
BROADER VIEW

Casted in FSMs, learn as a whole:

1) Policies for states/primitives
   - Our use of pairwise kernels to learning primitives from human demonstration

2) Transitions between states

3) State pre/postconditions

Jenkins - Learning Motion Primitives - 17
BEGINNINGS: ROBOT IMITATION

Estimate a robot policy that matches observed human behavior.

[Estimate a robot policy that matches observed human behavior]

[Fod, Mataric, Jenkins 2002]
[Jenkins, Mataric 2004]
Estimate a robot policy that matches observed human behavior.

Linear basis for human motion (Neuro-inspired)

\[ D(x,u) = \sum_i u_i B_i(x) \]
BEGINNINGS: ROBOT IMITATION

Estimate a robot policy that matches observed human behavior.

Linear basis for human motion (Neuro-inspired)

D(x,u) = Σ_i u_i B_i(x)

Perception → Motion Primitives → Motion Dynamics → Motion Control → Decision Making → Embodiment Sensing/Actuation

[Estro, Mataric, Jenkins 2002]
[Jenkins, Mataric 2004]

BEGINNINGS: ROBOT IMITATION

Estimate a robot policy that matches observed human behavior.

Linear basis for human motion (Neuro-inspired)

D(x,u) = Σ_i u_i B_i(x)

Perception → Motion Primitives → Motion Dynamics → Motion Control → Decision Making → Embodiment Sensing/Actuation

[Estro, Mataric, Jenkins 2002]
[Jenkins, Mataric 2004]

BEGINNINGS: ROBOT IMITATION

Estimate a robot policy that matches observed human behavior.

Linear basis for human motion (Neuro-inspired)

D(x,u) = Σ_i u_i B_i(x)

Perception → Motion Primitives → Motion Dynamics → Motion Control → Decision Making → Embodiment Sensing/Actuation

[Estro, Mataric, Jenkins 2002]
[Jenkins, Mataric 2004]

BEGINNINGS: ROBOT IMITATION

Estimate a robot policy that matches observed human behavior.

Linear basis for human motion (Neuro-inspired)

D(x,u) = Σ_i u_i B_i(x)

Perception → Motion Primitives → Motion Dynamics → Motion Control → Decision Making → Embodiment Sensing/Actuation

[Estro, Mataric, Jenkins 2002]
[Jenkins, Mataric 2004]
Estimate a robot policy that matches observed human behavior.

BEGINNINGS: ROBOT IMITATION

D(x,u) = \sum_i u_i B_i(x)

Motion Primitives

Perception

Decision Making

Motion Control

Embodiment Sensing/Actuation

Perception

Decision Making

Motion Control

Embodiment Sensing/Actuation

Classify human motion

Mirror Neurons

[Weiss et al]

Spinal Fields

Predict human motion

[Jenkins, Mataric 2004]

Linear basis for human motion (Neuro-inspired)

Motion Dynamics

[Perkel et al]

[Oneill et al]
LEARNING FSMs FROM DEMONSTRATION?

Attacker FSM

Seek ➔ Ball - Acquire ➔ Trap ➔ Aim ➔ Kick

Basic robot soccer attack move

[Grollman, Jenkins 09]
PERCEPTUAL ALIASING

Standard attack is 2 overlapping policies
  • distinguished by latent context variable
Unimodal attacker is much less efficient

Standard offensive move:
acquire ball, find goal, shoot

Unimodal attacker:
line up ball and goal, then shoot

assume only camera in nose and proprioception
**SQUARE ROOT EXAMPLE**

- Consider $y = \sqrt{x}$
- Averaging outputs will be incorrect
- 2 regressors needed for pos. and neg.

Locally Weighted Projection Regression or Gaussian Process Regression

Multimap Regression
INFINITE MIXTURES OF EXPERTS

\[ \pi : X \rightarrow Y \]
\[ (x_i, y_i)_{i=1..t} \]

\[ p(X, Y, Z) \propto p(Z)p(X|Z)p(Y|X, Z) \]

- Infer
- Given
- \( p(Z) \): prior over mixture models
- \( p(X|Z) \): regressor for each model
- \( p(Y|X, Z) \): predict output given input
- \( Z \): space of mixture models

[Grollman, Jenkins 09]
INFINITE MIXTURES OF EXPERTS

\[ \pi : X \rightarrow Y \]

\[ p(X, Y, Z) \propto p(Z)p(X|Z)p(Y|X, Z) \]

\[ Z: \text{space of mixture models} \]

\[ \text{cluster inputs into models} \]

\[ \text{predict output given input} \]

\[ \text{prior over models} \]

\[ \text{mixture model} \]

\[ \text{regressor for each model} \]

User demonstration

Motivation

Through robot learning from demonstration (LfD), our long-term goal is to provide typical users of consumer technology a medium for realizing their intended behavior into autonomous robot control policies. Specifically, given the same situation awareness, a robot controller representing a decision-making policy should make decisions similar to those of its user, the creator of the policy. While several paradigms exist for users to express robot policies, we remain confronted by a human-robot divide. This divide refers to the disparity between the diverse needs and ideas of users across society and their ability to instantiate robot control to meet their desired ends. We claim this human-robot divide is related to the expected personal robotics revolution, where robots become ubiquitous tools of use for society. The impact of personal computing over the past decades centered on the ability of users to reliably manipulate virtual domains through the storage and processing of digital information, such as in spreadsheets, web authoring, and virtual worlds. Analogously, the potential impact for robotics rests upon enabling human users to manipulate physical domains by crafting and interacting with autonomous robot controllers. However, the evolution from personal computing to robotics faces several challenges imposed by the physical world. In particular, the "write local, run global" model may not be appropriate for robotics. The performance of autonomous robots in terms of functionality and reliability is often sensitive to variations in physical environments, capabilities of robot platforms, and the nature of user-desired tasks. Such sensitivity highlights the need for adaptation of robot behavior to their local environment and robot platform as well as unknown tasks, not originally programmed on a robot and known only to a user.

Robot LfD offers one compelling direction for implicitly affecting robot decision making without explicitly modifying its control executable. In Robot LfD, robots are programmed implicitly from user demonstration by estimating a policy from collected demonstration data. This approach to crafting controllers is data-driven and sits in the perception of its state in an environment through experiences such as teleoperation, text-based and visual programming, speech and gesture instructions, and optimization/search.

One of our goals is to learn basic robot soccer control from demonstration using video game style interfaces. Experiences from deployment of this system to users across our department has indicated there are multiple valid approaches to goal scoring. Many of which are multivalued mappings from perceived robot state to action outputs.
INFINITE MIXTURES OF EXPERTS

\[ \pi : X \rightarrow Y \]

\[ (x_i, y_i)_{i=1..t} \]

\[ p(X, Y, Z) \propto p(Z)p(X|Z)p(Y|X, Z) \]

\[ Z: \text{space of mixture models} \]

\[ \text{prior over models} \]

\[ \text{mixture model} \]

\[ \text{regressor for each model} \]

Sparse (Pairwise) Gaussian Process Regression

User demonstration

Project Description

1 Motivation

Through robot learning from demonstration (LfD), our long-term goal is to provide typical users of consumer technology a medium for realizing their intended behavior into autonomous robot control policies. Specifically, given the same situation awareness as a robot controller representing a decision making policy, we should make decisions similar to those of its user, the creator of the policy. While several paradigms exist for users to express robot policies, we remain confronted by a human-robot divide. This divide refers to the disparity between the diverse needs and ideas of users across society and their ability to instantiate robot control to meet their desired ends. We claim this human-robot divide is related to the expected personal robotics revolution, where robots become ubiquitous tools of use for society. The impact of personal computing over the past decades centered on the ability of users to reliably manipulate virtual domains through the storage and processing of digital information, such as in spreadsheets, web authoring, and virtual worlds. Analogously, the potential impact for robotics rests upon enabling human users to manipulate physical domains by crafting and interacting with autonomous robot controllers. However, the evolution from personal computing to robotics faces several challenges imposed by the physical world. In particular, the “write local, run global” model may not be appropriate for robotics. The performance of autonomous robots in terms of functionality and reliability is often sensitive to variations in physical environments, capabilities of robot platforms, and the nature of user-desired tasks. Such sensitivity highlights the need for adaptation of robot behavior to their local environment and robot platform as well as unknown tasks, not originally programmed on a robot and known only to a user.

Robot LfD offers one compelling direction for implicitly affecting robot decision making without explicitly modifying its control executable. In Robot LfD, robots are programmed implicitly from user demonstration by estimating a policy from collected demonstration data. This approach to crafting controllers is data-driven and sits in the perception of its state in an environment (\(i\), \(e\), \(e1\), \(g1\), \(t\), \(e\)) with teleoperation, text-based and visual programming, speech and gesture instructions, and optimization/search.
\[ p(X, Y, Z) \propto p(Z)p(X|Z)p(Y|X, Z) \]

\( \pi : X \rightarrow Y \)

\[ (x_i, y_i)_{i=1..t} \]

\( Z \): space of mixture models

predict output given input

cluster inputs into models

prior over models

mixture model

regressor for each model

User demonstration

Gaussian Mixture Model

\[
p(X|Z) = \prod_{k=1}^{K_z} p(x_k|z = k) = \prod_{k=1}^{K_z} p(x_k|\mu_k, \Sigma_k)
\]

Sparse (Pairwise) Gaussian Process Regression

\[ \sigma \]

\[ \tau \]

\[ x' \]
The probability that the next integer takes on a new value of correct behavior is proportional to

\[ P(X, Y, Z) \propto p(Z)p(X|Z)p(Y|X, Z) \]

The Chinese Restaurant Process is given by:

\[
P(z_i = k|z_{-i}) = \begin{cases} \frac{m_k}{N + \alpha - 1}, & k \leq K_+ \\ \frac{\alpha}{N + \alpha - 1}, & k = K_+ + 1 \end{cases}
\]

\[
p(x_i, y_i)_{i=1..t}
\]

The Gaussian Mixture Model is:

\[
p(X|Z) = \prod_{k=1}^{K_+} p(x_k|z = k)
\]

\[
= \prod_{k=1}^{K_+} p(x_k|\mu_k, \Sigma_k)
\]

\[
P(X, Y, Z) \propto p(Z)p(X|Z)p(Y|X, Z)
\]

\[
\pi : X \rightarrow Y
\]

\[
(x_i, y_i)_{i=1..t}
\]

\[
P(z_i = k|z_{-i}) = \begin{cases} \frac{m_k}{N + \alpha - 1}, & k \leq K_+ \\ \frac{\alpha}{N + \alpha - 1}, & k = K_+ + 1 \end{cases}
\]

\[
p(x_i, y_i)_{i=1..t}
\]

\[
p(x_i, y_i)_{i=1..t}
\]

\[
P(z_i = k|z_{-i}) = \begin{cases} \frac{m_k}{N + \alpha - 1}, & k \leq K_+ \\ \frac{\alpha}{N + \alpha - 1}, & k = K_+ + 1 \end{cases}
\]

\[
p(x_i, y_i)_{i=1..t}
\]

\[
P(z_i = k|z_{-i}) = \begin{cases} \frac{m_k}{N + \alpha - 1}, & k \leq K_+ \\ \frac{\alpha}{N + \alpha - 1}, & k = K_+ + 1 \end{cases}
\]

\[
p(x_i, y_i)_{i=1..t}
\]
INFINITE MIXTURES OF EXPERTS

\[ \pi : X \rightarrow Y \]

\((x_i, y_i)_{i=1..t}\)

\[ p(X, Y, Z) \propto p(Z)p(X|Z)p(Y|X, Z) \]

Chinese Restaurant Process

\[
P(z_i = k|z_{-i}) = \begin{cases} 
\frac{m_k}{N+\alpha-1}, & k \leq K_+ \\
\frac{\alpha}{N+\alpha-1}, & k = K_+ + 1
\end{cases}
\]

Gaussian Mixture Model

\[
p(X|Z) = \prod_{k=1}^{K_+} p(x_k|z = k) = \prod_{k=1}^{K_+} p(x_k|\mu_k, \Sigma_k)
\]

User demonstration

Sparse incremental inference with particle filter

Sparse (Pairwise) Gaussian Process Regression

Project Description

Jenkins - Learning Motion Primitives - 30
LEARNED GOAL SCORER

Multimap Goal Scoring: Learned
Why robot learning?

Learning from demonstration

• Human motion primitives through dimension reduction
• Decision making primitives through infinite mixtures of experts

Learning → the path of least resistance?
Group Project

- Massive-scale learning from demonstration
- Implement in ROS; do Create tutorial:
  - http://code.google.com/p/brown-ros-pkg/
- Learn three tasks from demonstration
  - Create robot soccer
  - Nao magneto assembly
  - PR2 intern challenge
- One learning alg, one infrastructure box
- Human subjects study
Create CS148 Task

Robot Soccer
Create: Current Status

Goal scoring with regression

single objective possible, but multiple objectives remains problem
PR2 Intern Challenge

Serving Drinks
PR2: Current Status

PR2 simulator (Gazebo) running on maria/rlab

issues: experimental setup, code interface

PR2 Beta Program: Call for Proposals

Getting a PR2 at Brown! Applying for PR2 Beta Program
Nao: Current Status

issues: teleoperation interface, object recognition
Teleop interface with ARtags

Create: Current Status

issues: arintegration, localization