Planning Surface Cleaning Tasks by Learning Uncertain Drag Actions Outcomes

David Martínez, Guillem Alenyà and Carme Torras

CSIC-UPC

23rd International Conference on Automated Planning and Scheduling Workshop in Planning and Robotics (PlanRob)

June 11, 2013
Index

1 Introduction
2 Problem
3 Learning
4 Experiments
5 Conclusions
Motivation

Robot to clean surfaces
- Moving lentils to a container
- Fast execution
  - Planning
  - Minimize cleaning actions
Plan examples
Plan examples
Plan examples
Plan examples
Plan examples
Plan examples
Plan examples
Robot to clean surfaces
- Fast execution
  - Minimize cleaning actions
- Adapts to changes
  - Cloth grasping
Motivation

Robot to clean surfaces
- Fast execution
  - Minimize cleaning actions
- Adapts to changes
  - Cloth grasping
Plan examples

Grasp 1
Plan examples

Grasp 1
Grasp 2
Plan examples

Grasp 2
Objectives

- Minimize execution time
  - Robot actions are expensive
  - Planning best sequences of actions
    - Rules to define actions
- Adapting to changes
  - Adapting rules to grasps
  - Learning
Objectives

- Minimize execution time
  - Robot actions are expensive
  - Planning best sequences of actions
    - Rules to define actions
- Adapting to changes
  - Adapting rules to grasps
  - Learning
Setup

Arm manipulator: WAM

Surface to clean

Camera: Kinect

Setup
Setup

Arm manipulator: WAM

Surface to clean

Camera: Kinect
Setup

Arm manipulator: WAM

Camera: Kinect

Surface to clean
Setup

Arm manipulator: WAM

Camera: Kinect

Surface to clean

Setup
Overview

- Observations $\rightarrow$ state
- Rules $\rightarrow$ actions
- Planning
- Learning
Observations

- **Kinect image**

- **Dirt segmentation**

- **Surface segmentation**

- **Extract information about dirty areas**
  - Position
  - Size
  - Shape
  - Scattered
Observations

Kinect image

Surface segmentation

Dirt segmentation

Extract information about dirty areas
- Position
- Size
- Shape
- Scattered
Observations

Kinect image

Dirt segmentation

Surface segmentation

Extract information about dirty areas

- Position
- Size
- Shape
- Scattered
Observations

Kinect image

Dirt segmentation

Surface segmentation

Extract information about dirty areas

- Position
- Size
- Shape
- Scattered
Actions

Cleaning actions

- Straight move
- Fast move

Grouping actions

- Group scattered lentils
- Join 2 or 3 groups
**Actions**

**Cleaning actions**
- Straight move
- Fast move

**Grouping actions**
- Group scattered lentils
- Join 2 or 3 groups
Actions

Cleaning actions

- Straight move
- Fast move

Grouping actions

- Group scattered lentils
- Join 2 or 3 groups
Probabilistic planner

- Actions are stochastic
- All outcomes are important

Example cleaning action

- Outcome 1: clean a group of lentils
- Outcome 2: clean a part of the group
- Outcome 3: scatters the group
Probabilistic planner

- Actions are stochastic
- All outcomes are important

Example cleaning action

- Outcome 1: clean a group of lentils
- Outcome 2: clean a part of the group
- Outcome 3: scatters the group
Cleaning lentils
Changing conditions

Problem
- Cloth grasps change rules

Solution
- Learn rules for new grasps
- Good performance
Changing conditions

Problem
- Cloth grasps change rules

Solution
- Learn rules for new grasps
- Good performance
Index

1. Introduction
2. Problem
3. Learning
4. Experiments
5. Conclusions
Learning requirements

- Robot actions are slow
  - Learn with few actions

- Observability
  1. Partial observability requires more experience
  2. Accurate observations
    - Problem with occlusions
Learning requirements

- Robot actions are slow
  - Learn with few actions
- Observability
  1. Partial observability requires more experience
  2. Accurate observations
    - Problem with occlusions
Learning requirements

- Robot actions are slow
  - Learn with few actions
- Observability
  1. Partial observability requires more experience
  2. Accurate observations
     - Problem with occlusions
Learning requirements

- Robot actions are slow
  - Learn with few actions
- Observability
  1. Partial observability requires more experience
  2. Accurate observations
     - Problem with occlusions
Learning requirements

- Robot actions are slow
  - Learn with few actions
- Observability
  1. Partial observability requires more experience
  2. Accurate observations
     - Problem with occlusions
Learning

Learning requirements
- Learning a model
  - State has many objects
  - Symbolic domain
  - Robot actions are stochastic
  - Action uncertainty

Different approaches
- Model-based RL
- Object-oriented RL
  - Diuk et al, ICML 08
- RL in Relational world
  - Lang et al, JMLR 13
Learning

Learning requirements

- Learning a model
- State has many objects
  - Symbolic domain
  - Robot actions are stochastic
  - Action uncertainty

Different approaches

- Model-based RL
- Object-oriented RL
  - Diuk et al, ICML 08
- RL in Relational world
  - Lang et al, JMLR 13
Learning requirements

- Learning a model
- State has many objects
  - Symbolic domain
- Robot actions are stochastic
  - Action uncertainty

Different approaches

- Model-based RL
- Object-oriented RL
  - Diuk et al, ICML 08
- RL in Relational world
  - Lang et al, JMLR 13
Rule learning in robotics

- Exploration-exploitation
  - Initial steps are exploration
    - Random behaviour
  - Once some experience is obtained
    - Good results

- Problem in robotics
  - Actions are expensive
  - Poor performance during exploration
  - Guidance during initial steps
Rule learning in robotics

- Exploration-exploitation
  - Initial steps are exploration
    - Random behaviour
  - Once some experience is obtained
    - Good results

- Problem in robotics
  - Actions are expensive
  - Poor performance during exploration
  - Guidance during initial steps
Rule learning in robotics

- Exploration-exploitation
  - Initial steps are exploration
    - Random behaviour
  - Once some experience is obtained
    - Good results

- Problem in robotics
  - Actions are expensive
    - Poor performance during exploration
    - Guidance during initial steps
Rule learning in robotics

- Exploration-explotation
  - Initial steps are exploration
    - Random behaviour
  - Once some experience is obtained
    - Good results

- Problem in robotics
  - Actions are expensive
  - Poor performance during exploration
  - Guidance during initial steps
Rule learning in robotics

- Exploration-exploitation
  - Initial steps are exploration
    - Random behaviour
  - Once some experience is obtained
    - Good results
- Problem in robotics
  - Actions are expensive
  - Poor performance during exploration
  - Guidance during initial steps
Initial learning

- Improving initial learning steps
- We know some information about the model
  - Initial rules
- Start with optimistic initial rules
  - Get initial experiences
- Fast heuristic to refine the rules
- Until enough experience is obtained

Initial rule example:

Action: Fast clean
Preconditions:
  dirt(X)
Outcomes:
  1.0 -dirt(X)
  0.0 Nothing
Initial learning

- Improving initial learning steps
- We know some information about the model
  - Initial rules
- Start with optimistic initial rules
  - Get initial experiences
- Fast heuristic to refine the rules
- Until enough experience is obtained

Initial rule example:

| Action:   | Fast clean |
| Preconditions:  | dirt(X) |
| Outcomes:       | 1.0 -dirt(X) |
|                 | 0.0 Nothing |
Initial learning

- Improving initial learning steps
- We know some information about the model
  - Initial rules
- Start with optimistic initial rules
  - Get initial experiences
- Fast heuristic to refine the rules
- Until enough experience is obtained

Initial rule example:

**Action:** Fast_clean
**Preconditions:**
- dirt(X)
**Outcomes:**
- 1.0 -dirt(X)
- 0.0 Nothing
Initial learning

- Improving initial learning steps
- We know some information about the model
  - Initial rules
- Start with optimistic initial rules
  - Get initial experiences
- Fast heuristic to refine the rules
- Until enough experience is obtained

Initial rule example:

Action: Fast_clean
Preconditions:
dirt(X)
Outcomes:
  1.0 -dirt(X)
  0.0 Nothing
Initial rules refinement

- Requirements
  - Few experiences available
  - Rule refinement after every execution
  - Fast

Decreasing-m-estimate

- Learning heuristic to update probabilities
- Based on m-estimate
- Very fast
Initial rules refinement

**Requirements**
- Few experiences available
- Rule refinement after every execution
- Fast

**Decreasing-m-estimate**
- Learning heuristic to update probabilities
- Based on m-estimate
- Very fast
\[ P = \frac{p + mP_0}{p + n + m}. \] (1)

- **Parameters:**
  - \( m \) - Learning parameter
  - \( P \) - Probability
  - \( P_0 \) - Probability
  - \( p \) - Positive examples
  - \( n \) - Negative examples

- Initial probability has much influence
m-estimate

\[ P = \frac{p + mP_0}{p + n + m}. \]  \hspace{1cm} (1)

- Parameters:
  - \( m \) - Learning parameter
  - \( P \) - Probability
  - \( P_0 \) - Probability
  - \( p \) - Positive examples
  - \( n \) - Negative examples

- Initial probability has much influence
Decreasing m-estimate

\[ P = \frac{p + \left(\frac{m}{\sqrt{p + n}}\right)P_0}{p + n + \left(\frac{m}{\sqrt{p + n}}\right)}. \]  

- \( m \) decreases as experience is obtained
- **Parameters:**
  - \( m \) - Learning parameter
  - \( P \) - Probability
  - \( P_0 \) - Probability
  - \( p \) - Positive examples
  - \( n \) - Negative examples
Pre-trained initial rules

- Only grasps changes
- Pre-trained initial rules
  - Using
    - Optimistic initial rules
    - Good cloth grasp
- Obtain new rule set
  - Already learned some dynamics of the system
Pre-trained initial rules

- Only grasps changes
- Pre-trained initial rules
- Using
  - Optimistic initial rules
  - Good cloth grasp
- Obtain new rule set
  - Already learned some dynamics of the system
Pre-trained initial rules

- Only grasps changes
- Pre-trained initial rules
- Using
  - Optimistic initial rules
  - Good cloth grasp
- Obtain new rule set
  - Already learned some dynamics of the system
Learning overhead

- Learning requires
  - Accurate perception
  - No occlusions
  - Overhead
- Stop learning
  - Enough samples are obtained
  - Hoeffding bound
Learning overhead

- Learning requires
  - Accurate perception
  - No occlusions
  - Overhead
- Stop learning
  - Enough samples are obtained
  - Hoeffding bound

Occlusion

Moved arm
Learning overhead

- Learning requires
  - Accurate perception
  - No occlusions
  - Overhead

- Stop learning
  - Enough samples are obtained
  - Hoeffding bound
Learning overhead

- Learning requires
  - Accurate perception
  - No occlusions
  - Overhead

- Stop learning
  - Enough samples are obtained
  - Hoeffding bound
Pre-trained rules

Generating pre-trained rules

![Graph showing the good outcome probability over iterations for different scenarios. The graph includes lines for join 2 groups, join 3 groups, clean fast, and clean straight, with iterations ranging from 0 to 15.]
Rule adaptation

Refining rules with grasp 1

Refining rules with grasp 2

![Graph showing good outcome probability over iterations for different groups and grasps](image1.png)

![Graph showing good outcome probability over iterations for different groups and grasps](image2.png)
Performance tests

Actions executed and learned

![Graph showing executed and learned actions over iterations.]

Execution time

![Bar chart showing execution time for different iterations.]

- Executed actions
- Learned actions
- Planning
- Actions
- Learning + extra perceptions
Performance tests (decreasing m-estimate)

Comparing m-estimate vs decreasing m-estimate
Index

1. Introduction
2. Problem
3. Learning
4. Experiments
5. Conclusions
Conclusions

- Improved performance for robotic applications
  - Stochastic actions
- Online learning
  - Improving initial learning steps with simple rules
  - Fast heuristic to refine them
    - Decreasing m-estimate
- Robotic surface cleaning
Conclusions

- Improved performance for robotic applications
  - Stochastic actions
- Online learning
  - Improving initial learning steps with simple rules
  - Fast heuristic to refine them
    - Decreasing m-estimate
- Robotic surface cleaning
Conclusions

- Improved performance for robotic applications
  - Stochastic actions
- Online learning
  - Improving initial learning steps with simple rules
  - Fast heuristic to refine them
    - Decreasing m-estimate
- Robotic surface cleaning
Future work

- Have a prelearned set of grasps
  - Find a similar grasp
- Better integration with other learning methods
  - Incrementally update preconditions and outcomes
- Partial observability
Future work

- Have a prelearned set of grasps
  - Find a similar grasp
- Better integration with other learning methods
  - Incrementally update preconditions and outcomes
- Partial observability
Future work

- Have a prelearned set of grasps
  - Find a similar grasp
- Better integration with other learning methods
  - Incrementally update preconditions and outcomes
- Partial observability
Thanks!

Questions?
Planning Surface Cleaning Tasks by Learning Uncertain Drag Actions Outcomes

David Martínez, Guillem Alenyà and Carme Torras

23rd International Conference on Automated Planning and Scheduling Workshop in Planning and Robotics (PlanRob)

June 11, 2013