Planning and Optimization

B2. Regression: Introduction & STRIPS Case

Malte Helmert and Gabriele Röger

Universität Basel

October 13, 2016

Planning and Optimization

October 13, 2016 — B2. Regression: Introduction & STRIPS Case

B2.1 Regression

B2.2 Regression Example

B2.3 Regression for STRIPS Tasks

B2.4 Summary

B2.1 Regression

Forward Search vs. Backward Search

Searching planning tasks in forward vs. backward direction is not symmetric:

- forward search starts from a single initial state; backward search starts from a set of goal states
- when applying an operator o in a state s in forward direction, there is a unique successor state s'; if we just applied operator o and ended up in state s', there can be several possible predecessor states s

$\rightsquigarrow$ in most natural representation for backward search in planning, each search state corresponds to a set of world states

Planning by Backward Search: Regression

Regression: Computing the possible predecessor states $regr_o(S')$ of a set of states S' ("subgoal") given the last operator o that was applied.

$\rightsquigarrow$ formal definition in next chapter

Regression planners find solutions by backward search:

- start from set of goal states
- iteratively pick a previously generated subgoal (state set) and regress it through an operator, generating a new subgoal
- solution found when a generated subgoal includes initial state

pro: can handle many states simultaneously
con: basic operations complicated and expensive

Search Space Representation in Regression Planners

identify state sets with logical formulas (again):

- each search state corresponds to a set of world states ("subgoal")
- each search state is represented by a logical formula: φ represents $\{s \in S \mid s \models \varphi\}$
- many basic search operations like detecting duplicates are NP-complete or coNP-complete

Search Space for Regression

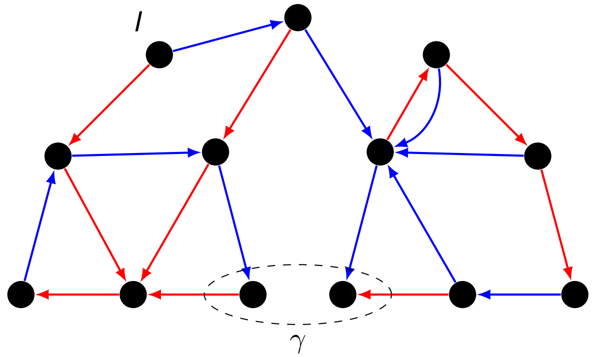
Search Space for Regression

search space for regression in a planning task $\Pi = \langle V, I, O, \gamma\rangle$ (search states are formulas φ describing sets of world states; actions of search space are operators $o \in O$)

- init() $\rightsquigarrow$ returns γ
- is_goal(φ) $\rightsquigarrow$ tests if $I \models \varphi$
- succ(φ) $\rightsquigarrow$ returns all pairs $\langle o, regr_o(\varphi)\rangle$ where $o \in O$ and $regr_o(\varphi)$ is defined
- cost(o) $\rightsquigarrow$ returns $cost(o)$ as defined in Π
- h(φ) $\rightsquigarrow$ estimates cost from I to φ ($\rightsquigarrow$ Parts C and D)

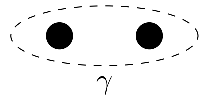
B2.2 Regression Example

Regression Planning Example (Depth-first Search)



Regression Planning Example (Depth-first Search)

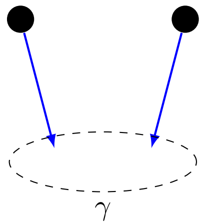
γ



Regression Planning Example (Depth-first Search)

$\varphi_1 = \mathit{regr}_{\longrightarrow}(\gamma)$

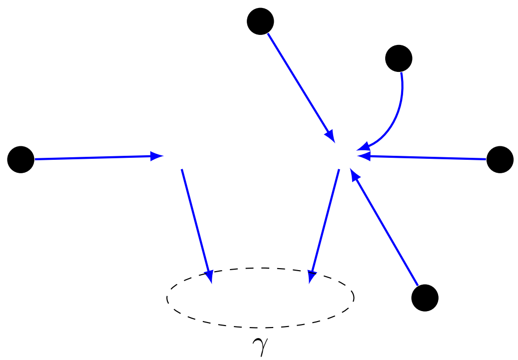
$\varphi_1 \longrightarrow \gamma$



Regression Planning Example (Depth-first Search)

$\varphi_1 = \mathit{regr}_{\longrightarrow}(\gamma)$
$\varphi_2 = \mathit{regr}_{\longrightarrow}(\varphi_1)$

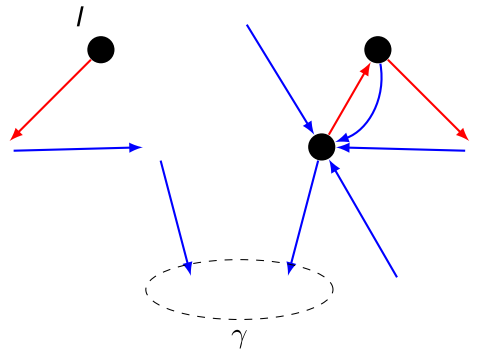
$\varphi_2 \longrightarrow \varphi_1 \longrightarrow \gamma$



Regression Planning Example (Depth-first Search)

$\varphi_1 = regr_{\longrightarrow}(\gamma)$
$\varphi_2 = regr_{\longrightarrow}(\varphi_1)$
$\varphi_3 = regr_{\longrightarrow}(\varphi_2), I \models \varphi_3$

$\varphi_3 \longrightarrow \varphi_2 \longrightarrow \varphi_1 \longrightarrow \gamma$



B2.3 Regression for STRIPS Tasks

Regression for STRIPS Planning Tasks

Regression for STRIPS planning tasks is much simpler than the general case:

- Consider subgoal φ that is conjunction of atoms $a_1 \land \dots \land a_n$ (e.g., the original goal γ of the planning task).
- First step: Choose an operator o that deletes no a_i.
- Second step: Remove any atoms added by o from φ.
- Third step: Conjoin $pre(o)$ to φ.

$\leadsto$ Outcome of this is regression of φ w.r.t. o.
It is again a conjunction of atoms.

optimization: only consider operators adding at least one a_i

STRIPS Regression

Definition (STRIPS Regression)

Let $\varphi = \varphi_1 \land \dots \land \varphi_n$ be a conjunction of atoms, and let o be a STRIPS operator which adds the atoms $a_1, \dots, a_k$ and deletes the atoms $d_1, \dots, d_l$. (W.l.o.g., $a_i \neq d_j$ for all i, j.)

The STRIPS regression of φ with respect to o is

$$sregr_o(\varphi) := \begin{cases} \bot & \text{if } \varphi_i = d_j \text{ for some } i, j \\ pre(o) \land \bigwedge(\{\varphi_1, \dots, \varphi_n\} \setminus \{a_1, \dots, a_k\}) & \text{otherwise} \end{cases}$$

Note: $sregr_o(\varphi)$ is again a conjunction of atoms, or $\bot$.

Does this Capture the Idea of Regression?

For our definition to capture the concept of regression, it should satisfy the following property:

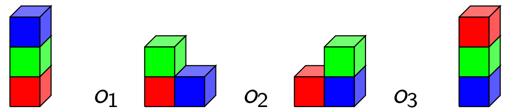
Regression Property

For all sets of states described by a conjunction of atoms φ, all states s and all STRIPS operators o,

$$s \models sregr_o(\varphi) \quad \text{iff} \quad s[\![o]\!] \models \varphi.$$

This is indeed true. We do not prove it now because we prove this property for general regression (not just STRIPS) later.

STRIPS Regression Example



Note: Predecessor states are in general not unique.
This picture is just for illustration purposes.

$$\begin{aligned}
o_1 &= \langle \textcolor{blue}{\blacksquare} on \textcolor{green}{\blacksquare} \land \textcolor{blue}{\blacksquare} clr, \quad \neg \textcolor{blue}{\blacksquare} on \textcolor{green}{\blacksquare} \land \textcolor{blue}{\blacksquare} onT \land \textcolor{green}{\blacksquare} clr \rangle \\
o_2 &= \langle \textcolor{green}{\blacksquare} on \textcolor{red}{\blacksquare} \land \textcolor{green}{\blacksquare} clr \land \textcolor{blue}{\blacksquare} clr, \quad \neg \textcolor{blue}{\blacksquare} clr \land \neg \textcolor{green}{\blacksquare} on \textcolor{red}{\blacksquare} \land \textcolor{green}{\blacksquare} on \textcolor{blue}{\blacksquare} \land \textcolor{red}{\blacksquare} clr \rangle \\
o_3 &= \langle \textcolor{red}{\blacksquare} onT \land \textcolor{red}{\blacksquare} clr \land \textcolor{green}{\blacksquare} clr, \quad \neg \textcolor{green}{\blacksquare} clr \land \neg \textcolor{red}{\blacksquare} onT \land \textcolor{red}{\blacksquare} on \textcolor{green}{\blacksquare} \rangle
\end{aligned}$$

$$\begin{aligned}
\gamma &= \textcolor{red}{\blacksquare} on \textcolor{green}{\blacksquare} \land \textcolor{green}{\blacksquare} on \textcolor{blue}{\blacksquare} \\
\varphi_1 &= sregr_{o_3}(\gamma) = \textcolor{red}{\blacksquare} onT \land \textcolor{red}{\blacksquare} clr \land \textcolor{green}{\blacksquare} clr \land \textcolor{green}{\blacksquare} on \textcolor{blue}{\blacksquare} \\
\varphi_2 &= sregr_{o_2}(\varphi_1) = \textcolor{green}{\blacksquare} on \textcolor{red}{\blacksquare} \land \textcolor{green}{\blacksquare} clr \land \textcolor{blue}{\blacksquare} clr \land \textcolor{red}{\blacksquare} onT \\
\varphi_3 &= sregr_{o_1}(\varphi_2) = \textcolor{blue}{\blacksquare} on \textcolor{green}{\blacksquare} \land \textcolor{blue}{\blacksquare} clr \land \textcolor{green}{\blacksquare} on \textcolor{red}{\blacksquare} \land \textcolor{red}{\blacksquare} onT
\end{aligned}$$

B2.4 Summary

Summary

- Regression search proceeds backwards from the goal.
- Each search state corresponds to a set of world states, for example represented by a formula.
- Regression is simple for STRIPS operators.
- The theory for general regression is more complex. This is the topic of the following chapters.