**Goal:** Enable autonomous agents to learn how to plan efficiently in massive stochastic state spaces using PFS.

**Want to Infer:**

\[
Pr(a_i \in A^* | s, G) 
\]

(Probability that action \( a \) is optimal for state \( s \) given goal \( G \) — i.e. a Bernoulli)

**Estimating Action Optimality:**

Tabularly solve a policy for easy worlds in domain, estimate action optimality

Use a set of binary features, \( (\phi) \), to featurize states where \( p \) is a propositional function and \( g \) is a goal

Train Naive Bayes classifiers on easy world policies

\[
Pr(a_i \in A^* | s, G) \approx \frac{C(\phi_j, a_i)}{C(a_i)} + C(a_i) 
\]

**Action Pruning:**

Prune actions unlikely to be involved in optimal plan, thereby pruning states, plan in resulting state space

---

**Leveraging Results:**

**Learning Results:**

- **Bellman Updates:**
  - No Pruning
  - Expert Pruning
  - Learned Pruning

- **Plan Cost:**
  - No Pruning
  - Expert Pruning
  - Learned Pruning

- **CPU Time:**
  - No Pruning
  - Expert Pruning
  - Learned Pruning

---

**Learning:**

**Goal:** Produce useful propositional functions given only an OO–MDP problem representation.

**Relationally Featurize States:**

First gather RL agent observations in domain

Featurize observed states based on the relative values of object attributes.

\[
\phi_i = \begin{cases} 
1 & p_i(s) \wedge g_i(G) \\
0 & \text{otherwise} 
\end{cases} 
\]

**Create Data Sets for Each DOORMAX Prediction:**

Algorithm 1: DOORMAX Learns Modified to Generate \( D, \psi \)

**Experimental Results**

- **Minecraft Results:**
  - Bellman Updates
  - Plan Cost
  - CPU Time

- **Robotic Cooking Assistant Results:**
  - Bellman Updates
  - Plan Cost
  - CPU Time

**Learning Results:**

- **Agreement with PFs:**
  - DOORMAX Without PFs

**Goals:**

- Produce useful propositional functions given only an OO–MDP problem representation.

- Produce useful propositional functions given only an OO–MDP problem representation.