Online Discovery of Feature Dependencies

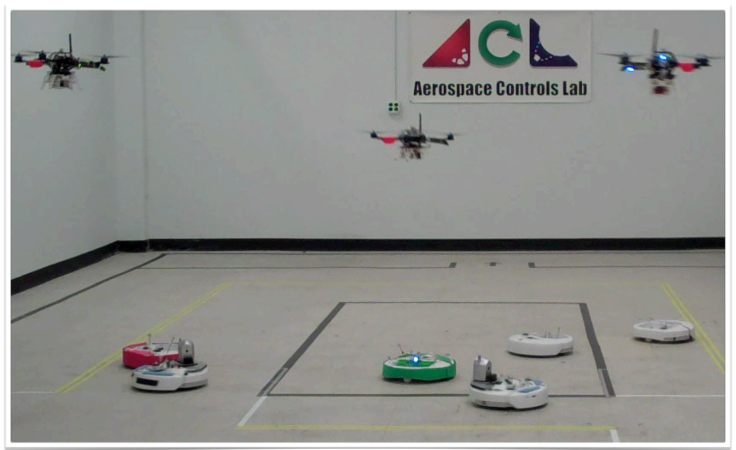
Alborz Geramifard, Finale Doshi-Velez, Joshua Redding, Nicholas Roy, Jonathan P. How

Abstract

Online **representational expansion** techniques have improved the learning speed of existing reinforcement learning (RL) algorithms in low dimensional domains, yet existing online expansion methods do not **scale** well to high dimensional problems. We conjecture that one of the main difficulties limiting this scaling is that features defined over the full-dimensional state space often generalize poorly. Hence, we introduce incremental Feature Dependency Discovery (**iFDD**) as a computationally-inexpensive method for representational expansion that can be combined with any online, value-based RL method that uses binary features. Unlike other online expansion techniques, iFDD creates new features in **low dimensional subspaces** of the full state space where feedback errors persist. We provide **convergence** and **computational complexity guarantees** for iFDD, as well as showing empirically that iFDD scales well to high dimensional multi-agent planning domains with **hundreds of millions** of state-action pairs.

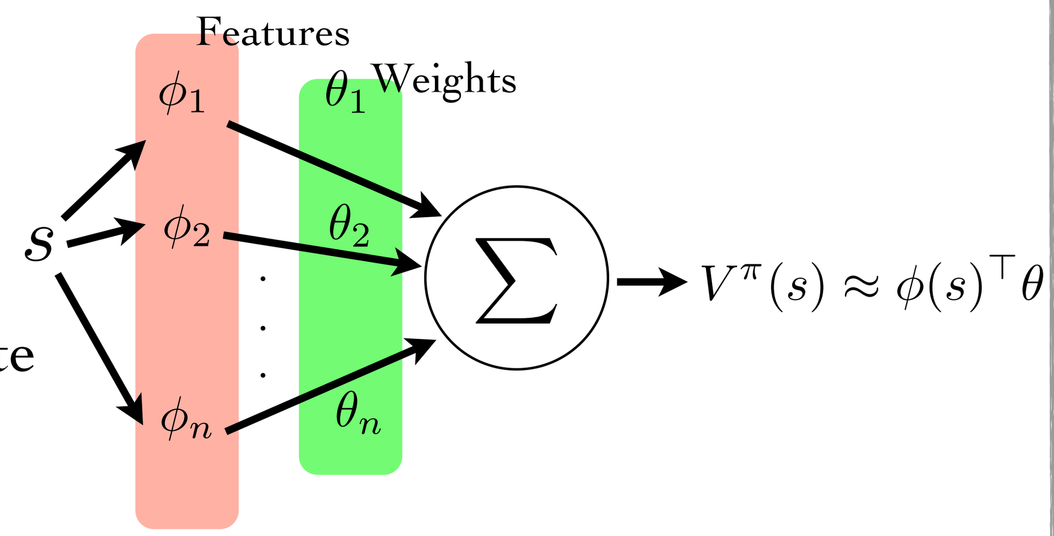
Problem

1. Real-world sequential decision making problems such as **multi-agent** domains have **large** state spaces, making it **impractical** to store state values using a lookup table.


Air and ground robots carrying out a mission, MIT, 2011

2. **Linear function approximation** has been a successful approach.

s = State
π = Policy
$V^\pi(s)$ = Value of state



3. Finding the **right** set of features used for approximation is hard. Our algorithm **autonomously** finds relevant **feature dependencies** as new features, given an initial set of **binary** features.

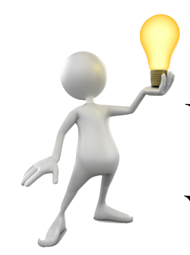
Existing Gap in the Literature

- Lack of **convergence** guarantees due to nonlinear approximation [1]
- Expensive **computational complexity** such as inverting large matrices[2]
- Excessive **sample complexity**[3]
- Requires tuning many **parameters** by hand[4]

Contributions

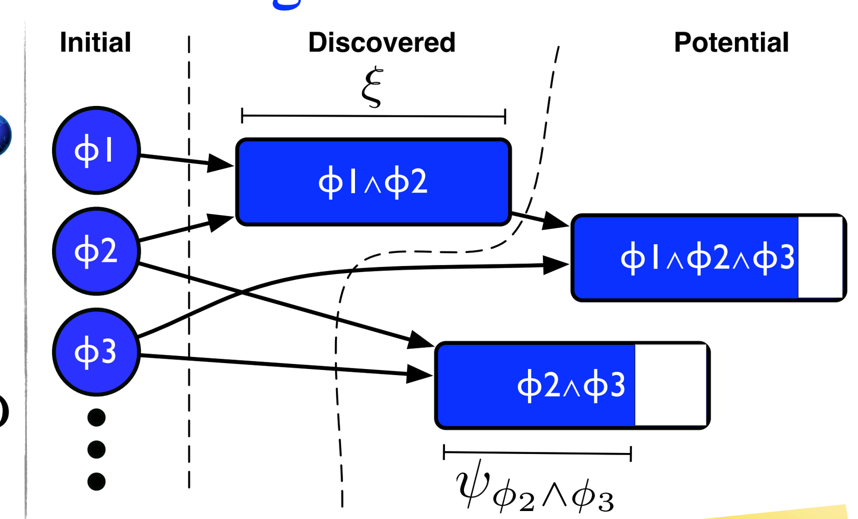
- Introduced incremental Feature Dependency Discovery (**iFDD**) as a **simple** and **computationally-inexpensive** feature expansion method
- Provided asymptotic **convergence** and per-time-step **computational complexity** analysis
- Empirically showed the **scalability** of the new approach in domains with large planning spaces ≈ $\mathbf{10^8}$

Approach



Where the most **accumulated error** is gathered, is where the representation should **grow**.

1. One Step Interaction
2. Update Weights — use Algorithm 2 for $\phi(.)$ — Sarsa $\downarrow \delta_t$
3. Discover Features — use Algorithm 1 — iFDD



Algorithm 1: Discover

Input: $\phi(s), \delta_t, \xi, \mathbf{F}, \psi$
Output: $\mathbf{F}, \psi$

foreach $(g, h) \in \{(i, j) | \phi_i(s)\phi_j(s) = 1\}$ **do**
    $f \leftarrow g \wedge h$
    if $f \notin \mathbf{F}$ **then**
        $\psi_f \leftarrow \psi_f + |\delta_t|$
        if $\psi_f > \xi$ **then**
            $\mathbf{F} \leftarrow \mathbf{F} \cup f$

Expand $\phi(s)$

ξ Discovery threshold
$\mathbf{F}$ Set of features
ψ Sum of errors
δ_t Error at time t
ϕ^0 Basis function for the initial feature set

Algorithm 2: Generate Feature Vector (ϕ)

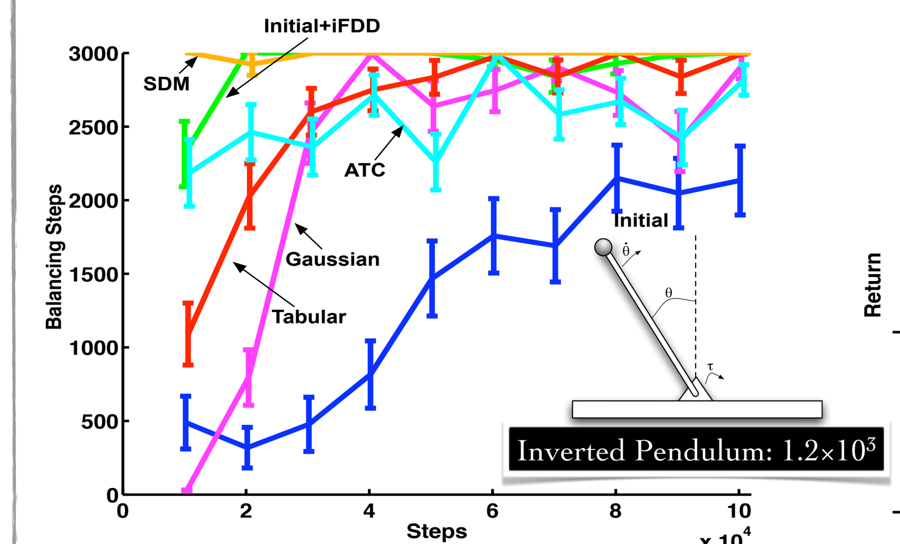
Input: $\phi^0(s), \mathbf{F}$
Output: $\phi(s)$

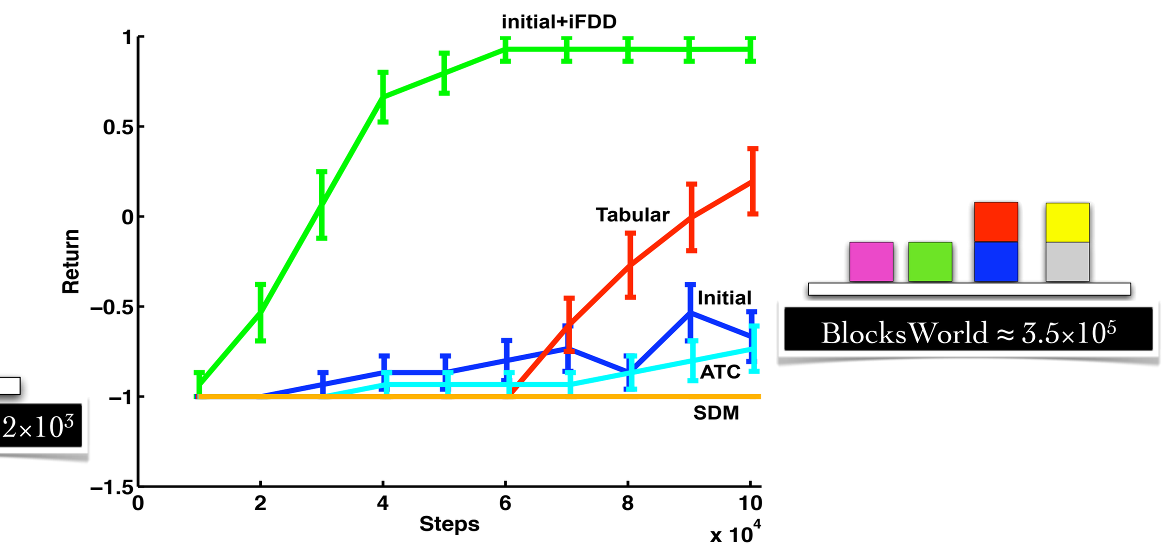
$\phi(s) \leftarrow \bar{0}$
activeInitialFeatures $\leftarrow \{i | \phi^0_i(s) = 1\}$
Candidates $\leftarrow$ SortedPowerSet(*activeInitialFeatures*)
while *activeInitialFeatures* $\neq \emptyset$ **do**
    $f \leftarrow$ *Candidates*.next()
    if $f \in \mathbf{F}$ **then**
        activeInitialFeatures $\leftarrow$ *activeInitialFeatures* $\setminus f$
        $\phi_f(s) \leftarrow 1$
return $\phi(s)$

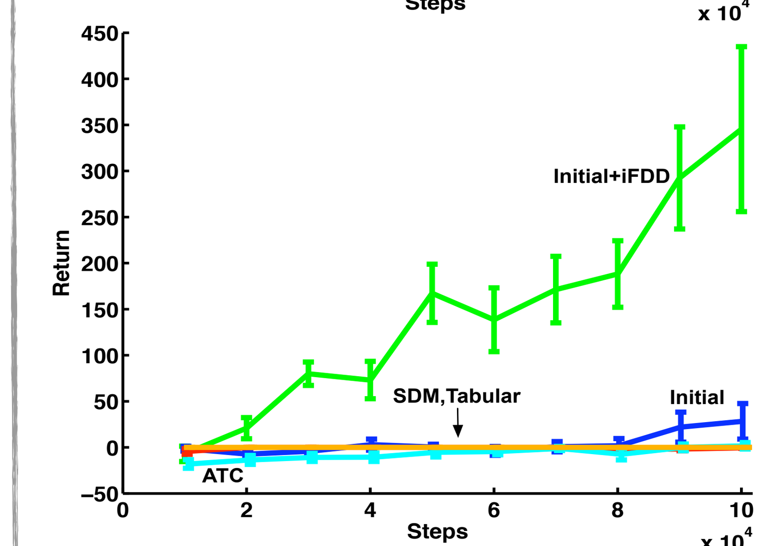
Make $\phi(s)$ sparse

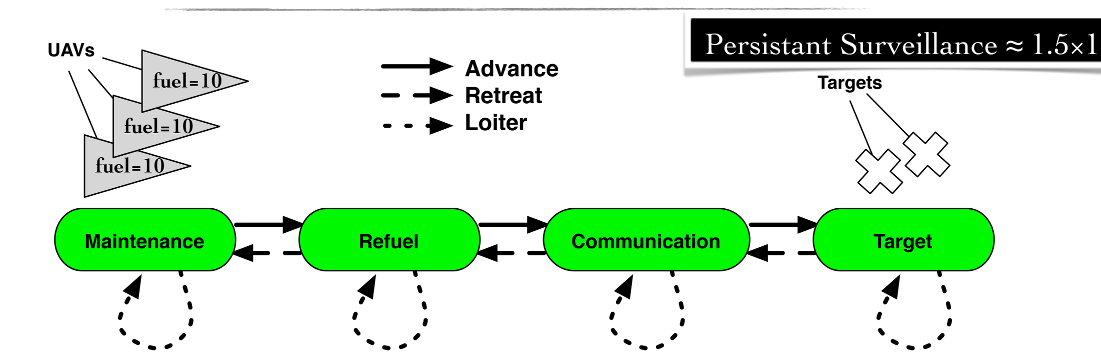
Empirical Results

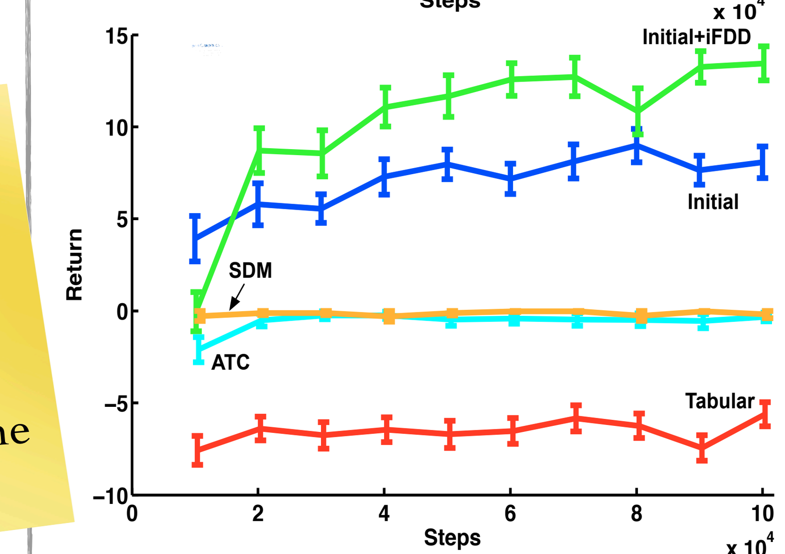
- Representations used with Sarsa:
  (1)initial (2)**initial+iFDD** (3)ATC[3](4)SDM[5] (5)Tabular

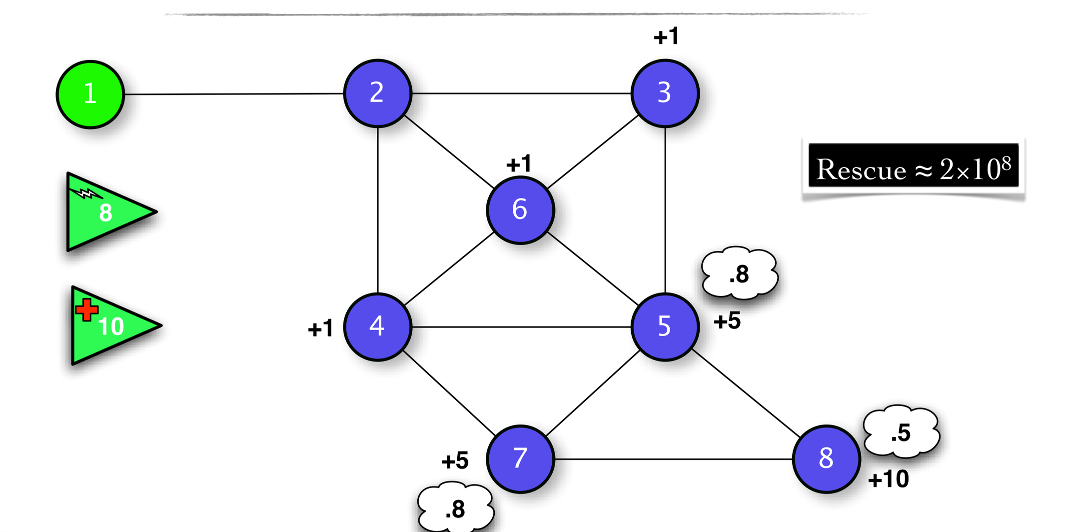






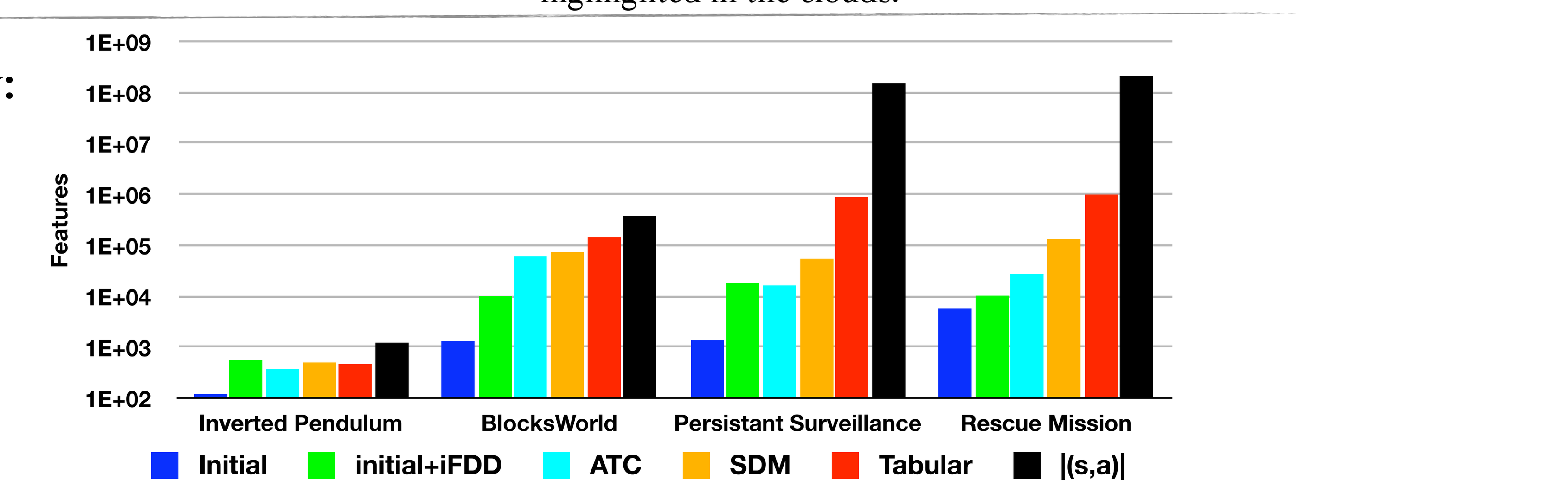






Two UAVs **collaborate** to rescue people at nodes. Number of people to rescue is shown as positive numbers. Probability of success is highlighted in the clouds.

- Memory:



Conclusion

- iFDD performed **really well** across all domains and its success was not due to the quantity of features as shown by the low memory usage but the **quality** of features. Given sparse features, iFDD has per-time-step complexity **independent** of the number of features. Combined with TD learning, iFDD is guaranteed to converge to the **best** possible approximation of the value function for a fixed policy.

[1] F. Rivest and D. Precup, "**Combining TD-learning with cascade-correlation networks**", In Proceedings of the Twentieth International Conference on Machine Learning (ICML), pages 632–639.AAAI Press, 2003.
[2] J. hong Wu and R. Givan, "**Feature-discovering approximate value iteration methods**", in Symposium on Abstraction, Reformulation, and Approximation (SARA), Vol. 3607, pp. 321-331, Springer, Lecture Notes in Computer Science, 2005.
[3] S. Whiteson, M. E. Taylor, and P. Stone, "**Adaptive tile coding for value function approximation**", Technical Report AI-TR-07-339, University of Texas at Austin, 2007.
[4] J. Zico Kolter and Andrew Y. Ng, "**Regularization and feature selection in least-squares temporal difference learning**", In Proceedings of the 26th Annual International Conference on Machine Learning (ICML), pages 521–528, New York, NY, USA, 2009.
[5] B. Ratitch and D. Precup, "**Sparse distributed memories for on-line value-based reinforcement learning**", In Jean-Francois Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, ECML, volume 3201 of Lecture Notes in Computer Science, pages 347–358. Springer, 2004.

ICML 2011 Bellevue, Washington
Sponsored by:
FA9550-09-1-0522
NSERC CRSNG 374172-2009