Batch-iFDD for Representation Expansion in Large MDPs
Alborz Geramifard, Tom Walsh, Nicholas Roy, Jonathan P. How

Abstract

Matching pursuit (MP) techniques are a promising class of feature construction algorithms for value function approximation. Yet until now, applying MP methods required creating a pool of potential features, mandating background knowledge or enumeration of a large feature set, both of which hinder scalability. This paper introduces batch incremental feature dependency discovery (Batch-iFDD), an algorithm that is proven to be an MP technique, hence inheriting a provable convergence property. Additionally, Batch-iFDD does not require a large pool of potential features, leading to lower computational complexity. Empirical policy evaluation results across domains with states up to one million states highlight the scalability of Batch-iFDD over the previous state-of-the-art MP algorithm.

Problem

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

Good Policy $\pi^*(s) = \arg\max_a Q(s,a)$

Good Value Function $Q(s,a)$

Good Representation $Q(s,a) \approx \phi(s,a)^T \theta$

Finding the right set of features used for approximation is hard.

Existing Gap in the Literature

Hand Coding
- Domain Specific
- Time Consuming
- Online methods such as iFDD
- High Sample Complexity
- Batch Techniques such as BEBF and OMP-TD
- Scalability, Tuning

Contributions

- Introduced Batch-iFDD as a new technique that is both scalable and sample efficient.
- Provided theoretical results that Batch-iFDD is an OMP technique.
- Empirically tested the advantage of Batch-iFDD against OMP-TD in 3 domains with sizes over a million states.

Orthogonal Matching Pursuit (OMP)

Inputs: $X$
Labels: $Y$

$\hat{y} = \sum_{i=1}^n \phi_i(x) w_i$

Given a set of features at each step, add the feature which reduces $|Y - \hat{Y}|$ the most.

Theoretical Results

Theorem: iFDD in batch approximately finds the feature conjunction with the best guaranteed error reduction on each iteration. This means iFDD is an OMP technique.

Empirical Results

LSTD used 10K samples for each domain with a fixed policy.
Features were added after each LSTD run using:
- iFDD*, iFDD\(_p\), OMP-TD\(_p\(n\))

Scale of the potential feature set

Conclusion

iFDD* scaled better compared to OMP-TD and the previous version of iFDD ran in batch as iFDD* expanded more useful features to drive TD-Error to zero. The iFDD family also requires less computation and memory compared to OMP-TD as they grow the size of the potential features incrementally.

References


Sponsored by:

[Image of sponsors logos]