1709.04579 - Autonomous Extracting a Hierarchical Structure of Tasks in Reinforcement Learning and Multi-task Reinforcement Learning

- **Yunqiu Xu**

---

1. Introduction

- Challenges: Curse of dimensionality $\rightarrow$ Slow learning speed
- Recent related work includes FeUdal Net and Option Framework, similar to how they are mentioned in Meta Learning Shared Hierarchies $\rightarrow$ hard to handle MTRL
- Our goal is to speed up learning in both single task RL and multi-task RL
- Our work:
  - ARM-HSTRL $\rightarrow$ use association rule to extract sub-goals and their relationships, thus autonomously decompose tasks as hierarchical structure
  - Our method does not need the action model in advance or a separate phase of learning to obtain the required data for extracting hierarchy.
  - Our method does not need the action model in advance or a separate phase of learning to obtain the required data for extracting hierarchy.

2. ARM-HSTRL

- ARM: extracts association rules
  - Generate frequent itemsets using FP-growth
  - Generate association rules

Algorithm 1 ARM-HSTRL

1: **Input**: Transition that is a set of successful trajectories, *minsup*, *minconf*
2: **Output**: *HST*
3: Frequent Itemset = FP-growth (Transition, *minsup*)
4: Association Rules = Rule Generation (Frequent Itemset, *minconf*)
5: HST-construction (Association Rules) //See Algorithm 2

- HST-construction: converts association rules to a hierarchical structure tree

Algorithm 2 HST-construction

Input: *AR-set* is the set of association rules. *AR-set* = $\{AR_1, \ldots, AR_{NumRules}\}$
Output: *HST*

Construct a tree, T, with one node that is the root node, R.
for $i = 1 : NumRules$ **do**
  Parent-Node=R
  for $j = 1 : Len_i$ **do**
    $t = 1$
    $FlagM = 0$
    repeat
      num shows the number of children of the Parent-Node
      PN_t shows the t_{th} child of the Parent-Node
      if $AR_{ij} == PN_t$ **then**
        Parent-Node=PN_t
        $FlagM = 1$
      end if
      $t++$
    until $t <= num$ and $FlagM == 0$
    if $FlagM == 0$ **then**
      create a new child Node in the Parent-Node: $PN_{num+1} = AR_{ij}$
      Parent-Node=PN_{num+1}
    end if
  end for
end for

- ARM-HSTRL in MTRL

  - Some details can be found at Taylor and Stone (2009) Transfer learning for reinforcement learning domains: A survey

3. Experiment

- The experiments show the performance of Q-learning and ARM-HSTRL, from the figures we can see the significant difference, in both single task and MTRL

4. Conclusion

- Use association rule mining to extract sub-goals and build hierarchical structure
- Do not need the action model
- Do not limited to factored MDP
- Can learn from different and several trajectories
- Do not need to clean and to process the paths
- Efficient, practical, and leads to hierarchical optimal policies
- ARM-HSTRL can handle MTRL and transfer its learning among tasks with different transition functions, while a lack of structural knowledge makes Q-learning impractical.