Feature selection as Monte-Carlo Search in Growing Single Rooted Directed Acyclic Graph by Best Leaf Identification

A.Pelissier, A.Nakamura, K.Tabata - November 6th 2018

Laboratory for Pattern Recognition and Machine Learning, Hokkaido University, Japan

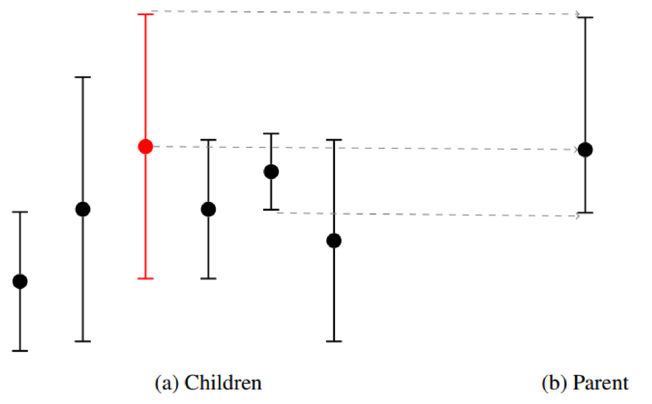
Abstract

We extend MCTS to single rooted directed acyclic graph (SR-DAG), and consider the Best Arm Identification (BAI) and the Best Leaf Identification (BLI) problem of an expanding SR-DAG of arbitrary depth. We propose algorithms that are (epsilon, delta)-correct in the fixed confidence setting, and prove an asymptotic upper bounds of sample complexity for our BAI algorithm. As a major application for our BLI algorithm, a novel approach for Feature Selection is proposed

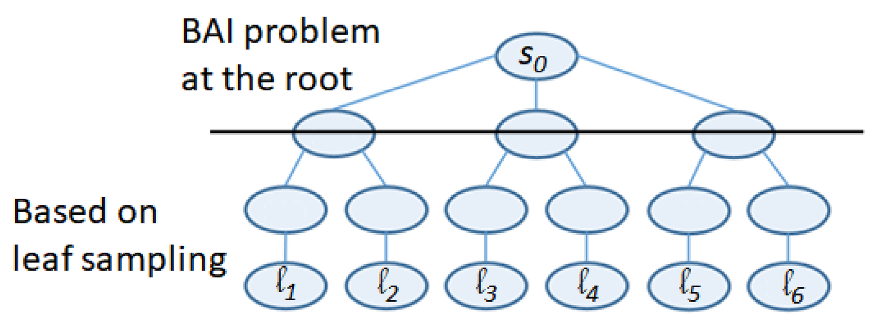
I. Monte Carlo DAG Search by Best Arm identification

BAI - MCDS

- In the context of MCDS, the BAI problem corresponds to identifying the next best action to take at the root.
- The best arm is identified by building confidence intervals using past observation from leaves.



$$\mathrm{L}_s(t) = \max_{c \in \mathcal{C}(s)} \mathrm{L}_c(t), \qquad \mathrm{U}_s(t) = \max_{c \in \mathcal{C}(s)} \mathrm{U}_c(t).$$



The confidence intervals are recursively defined from the children and updated after each sample leaf

For a risk level δ and some accuracy parameter ϵ, an algorithm is said (ϵ, δ)-PAC if

$$\mathbb{P}\left(\mu_{\hat{s}_\tau} \geq \mu^* - \epsilon\right) \geq 1 - \delta.$$

II. Monte Carlo search in Expanding DAGs

Algorithm BAI-exMCDS: Basic architecture of BAI-exMCDS

input : sr-dags $\mathcal{D}_0$, $\mathcal{D}$, risk level δ, accuracy ε, expansion parameter b
output: recommended arm $\hat{s}^*$

```
D_t ← D_0,  t ← 0
while not BAIStop(s_0, D_t) do
    if BAIExpand(D_t, t) then
        D_{t+1} ← BAIAdd(D_t, D)
    else
        D_{t+1} ← D_t
    R_{t+1} ← BAISelect(s_0, D_{t+1})
    X ← O_{ℓ_{R_{t+1}}}
    Update the confidence interval of ℓ_{R_{t+1}(t)}
    Update the confidence intervals for all the
      ancestors of ℓ_{R_{t+1}}(t).
    t ← t + 1
return BAIReco(s_0, D_t)
```

5 Rules of BAI - exMCDS

- *The sampling rule* **BAISelect**: that select a child from the root node.
- *The stopping rule* **BAIStop**: that return True if the algorithm decides to stop.
- *The recommendation rule* **BAIReco**: that select a candidate for the best arm.
- *The expansion rule* **BAIExpand**: that returns True if the algorithm decides to expand the DAG.
- *The addition rule* **BAIAdd**: that selects a new node to be added to the DAG

$\mathcal{O}_\ell$ is an oracle that evaluate the leaves, generated according to an unknown distribution over [0,1] with mean μ for each call.

III. LUCB - exMCDS

The five rules of the algorithm

$\mathtt{BAISelect}(s_0, \mathcal{D}_t)$: Return the node's child R_{t+1} selected as follows:

$$b_t \leftarrow \operatorname*{argmax}_{s \in \mathcal{C}(s_0)} \mathrm{L}_s(t), \qquad c_t \leftarrow \operatorname*{argmax}_{s \in \overline{\mathcal{C}}_{b_t}(s_0)} \mathrm{U}_s(t),$$

$$R_{t+1} \leftarrow \operatorname*{argmax}_{s \in \{b_t, c_t\}} [\mathrm{U}_s(t) - \mathrm{L}_s(t)].$$

$\mathtt{BAIStop}(s_0, \mathcal{D}_t)$: Return $\mathtt{True}$ if and only if $(\mathrm{U}_{c_t(t)} - \mathrm{L}_{b_t}(t) < \varepsilon)$.

$\mathtt{BAIReco}(s_0, \mathcal{D}_t)$: return b_t.

$\mathtt{BAIExpand}(\mathcal{D}_t, t)$: Return $\mathtt{True}$ if and only if $\lfloor (t+1)^b \rfloor - \lfloor t^b \rfloor = 1$

$\mathtt{BAIAdd}(\mathcal{D}_t, \mathcal{D})$:

BAIAdd empirically select the new node to be added by computing an index for each candidate nodes. But does not currently provide theoretical guaranties on all the possible leaves

Confidence intervals

$$\mathrm{L}_\ell(t) = \hat{\mu_\ell}(t) - \sqrt{\frac{\beta(N_\ell(t), \delta)}{2N_\ell(t)}} \quad \text{and}$$

$$\mathrm{U}_\ell(t) = \hat{\mu_\ell}(t) + \sqrt{\frac{\beta(N_\ell(t), \delta)}{2N_\ell(t)}} \quad \text{with}$$

$$\beta(N, \delta) = \ln\left(\frac{|\mathcal{L}_t|}{\delta}\right) + 3 \ln \ln\left(\frac{|\mathcal{L}_t|}{\delta}\right) + \frac{3}{2} \ln\left(\ln(N) + 1\right)$$

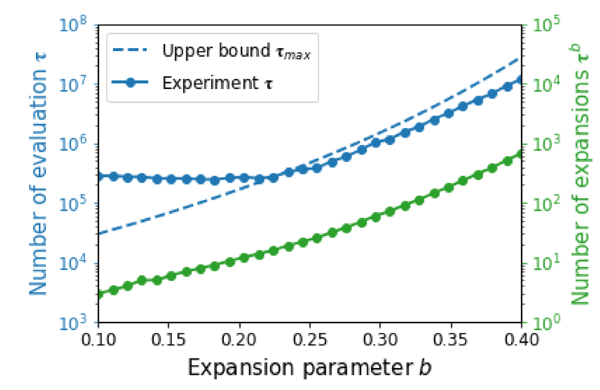
This exploration function guaranty (epsilon, delta) correctness of the recommended arm on the visited leaves.

Sample complexity guaranties

Theorem *For $\delta \leq \min(1, 0.1|\mathcal{L}_0|)$, under the condition that $\tau^b \gg |\mathcal{L}_0|$ and $\ln|\mathcal{L}_\tau| \gg \ln\ln\frac{1}{\Delta^2_{\varepsilon,*}}$, the number of leaf evaluation τ necessary to fulfill the stopping condition of UGapE-exMCDS is upper bounded by*

$$\tau \leq O\left(\left[\delta^{\frac{b-1}{b}} \exp\left(W_{-1}\left(\frac{\Delta^2_{\varepsilon,*}(b-1)}{8b\delta^{\frac{b-1}{b}}}\right)\right)\right]^{\frac{1}{b-1}}\right), \qquad \Delta_{\varepsilon,*} = \max(\Delta_*, \varepsilon)$$

where W_{-1} is the second real branch of the Lambert function [Corless et al. (1996)].



Number of iteration before the algorithm LUCB-exMCDS stops as a function of the expansion parameter *b.*

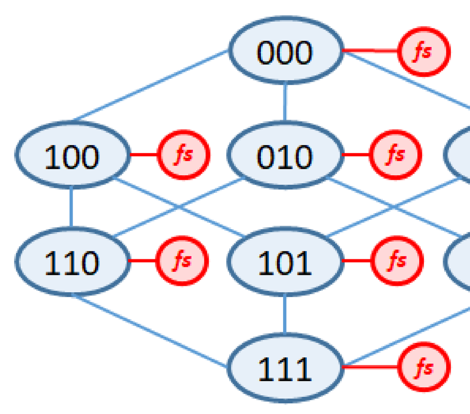
15 initial leafs
Δε = 0.05
δ = 0.1

IV. Best Leaf Identification and Feature selection

Feature DAG

- Each node in the feature lattice corresponds to a unique feature subset.
- An additional virtual stopping feature f_s is introduced to define the leaves
- The DAG corresponding to a feature set of size *f* contains 2^f nodes.

DAG architecture of a feature selection problem with 3 features.



Fast feature set evaluation

- Feature subsets are evaluated by a *k* Nearest Neighbor classifier trained with the selected features.
- To reduce the computational cost when dealing with large dataset, a small subsample $\mathcal{V}$ of the original set is computed.
- The score of the feature subset is taken as the Area under the ROC curve of the *k*NN predictions on $\mathcal{V}$.

RAVE score (Rapid Value Estimation)

- Each feature are associated to a global RAVE score, that is updated after each feature set evaluation

  $$\text{g-RAVE}_f = \text{average}\{V(F),\ f \in F\}$$

- RAVE score is used for the expansion policy **BAIAdd**

Best Leaf Identification

- We identify the best leaf by solving the BAI problem at each stage from the root until a terminal leaf is recommended
- The recommended leaf is (epsilon, delta) - correct within all the leaves that has been expanded at the end of the search.

Can compete with state of the art feature selection methods on various dataset