The synchronization of the blockchain

Huang Yaxin

No. 515030910039

Abstract

Blockchain is essentially a special distributed database. Firstly, the main role of blockchain is to store information. Besides, anyone can set up a server, join a blockchain network and become a node. In the world of blockchain, there is no central node. Each node is equal and it holds the entire database. You can write/read data to any node, since all nodes will be synchronized at the end, ensuring consistent blockchains. However, since the synchronization between nodes must be guaranteed, the speed of adding new blocks cannot be too fast. Thus, we consider to change the method of synchronization. In the traditional blockchain, the synchronization is come true by decided multicast. We consider the feasibility of inconclusive synchronization by changing the probability of synchronization. During this propose, we use a learning algorithm BLAG (Bandit on Large Action set Graph), which is based on the algorithm $\epsilon - greed$.

I. Model

In the algorithm BLAG, the author consider the information connection between sensitive nodes, uninformed nodes and informed nodes. His target is to adaptively diffuse sensitive information towards low degree nodes, while avoiding high degree nodes.

To use the algorithm BLAG in the synchronization of blockchain, we change some definition of the algorithm. We model the blockchain system as an graph $G = (V, E)$ with neither self-loops nor multiple edges between any two nodes. Each nodes is classified as either synchronized node or unsynchronized node. Each edge $e \in E$ has a weight representing the transmission probability of information between the two nodes it connects. Let G' denote the subgraph induced by synchronized nodes and $G \setminus G'$ the subgraph induced by unsynchronized nodes. Let ξ denote the set of edges connecting nodes from G' and nodes from $G \setminus G'$. We denote edges in ξ with unsynchronized nodes being destination as $target\ edges$ and these connected unsynchronized nodes as $target\ nodes$. In a time slot, let the m-dimensional vector $\vec{D}$ denote the degree set of

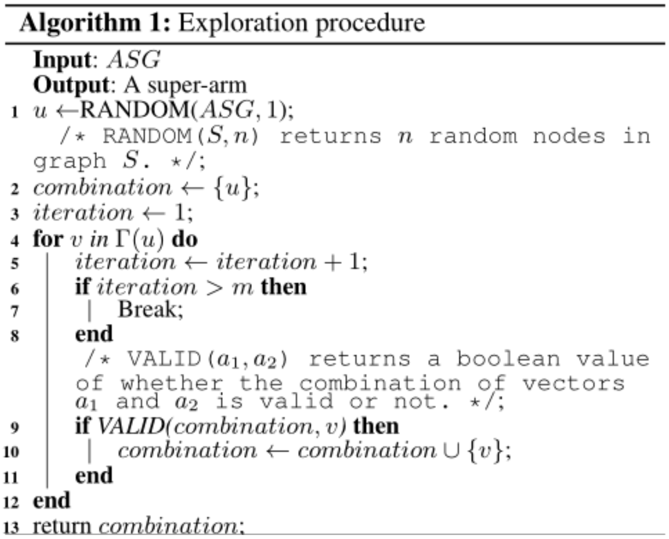
m $target\ nodes$, with $\vec{D}(i)$ representing the degree of $target\ node$ i. The m-dimensional vector $\vec{\beta}_0$ denotes original transmission probability on target edges, with $\vec{\beta}_0(i)$ representing original transmission probability on edge i. Let $\overrightarrow{\Delta\beta}$ denote the variation of probability on $target\ edges$, with $\overrightarrow{\Delta\beta}(i)$ represents variation of probability on edge i.
Thus, our target above is to learn an optimal $\overrightarrow{\Delta\beta^*}$ within the time slot that minimizes $\vec{D} \cdot (\vec{\beta}_0 + \overrightarrow{\Delta\beta^*})$.
The definition of trail, action, base-arm and reward is similar to which in the algorithm BLAG, so we only briefly introduce them and won't go into much detail here. A trial in round t refers to a transmission policy determined by $\overrightarrow{\Delta\beta^t}$. An action is a formed vector $\overrightarrow{\Delta\beta}$. Base-arms are vectors with pair-wise non-zero elements. Reward refers to feedback of an action, in the form of $\vec{D} \cdot \overrightarrow{\Delta\beta} + \sigma$, where σ is the noise in observation subjected to Gaussian distribution. Specially, because the sum of the probability will always smaller than 1, arbitrary combinations of several base-arms may not be valid. Thus, we give that combination of any two arms $\vec{\beta}_1$ and $\vec{\beta}_2$ is valid if and only if $\forall i, 1 \leq i \leq m, 0 \leq \vec{\beta}_0(i) + \vec{\beta}_1(i) + \vec{\beta}_2(i) \leq 1$. This combination of the valid base-arms is referred to super-arm.

II. Algorithm

Generally, BLAG, the global process of which follows an $\epsilon{-}greed$ process, can be decomposed into three parts: Exploration, Exploitation and Update parameters. We start with the ASG(action set graph), which represents whether combination of two base-arms is valid. The topology of ASG follows two rules: (1) Each node in ASG represents a base-arm $\vec{\beta}_i$, and (2) the weight of each nodes represents the current estimated reward of the corresponding base-arm. If the combination of the two base-arms is valid, their corresponding nodes in ASG are connected by an unweighted edge. Obviously, any valid combination is a clique in ASG, and nodes in a clique are pair-wise connected.

A. Exploration

The target in exploration procedure is to get a large size combination. Regarding ASG, this can be treated as a Maximum Clique like problem in the graph theory.

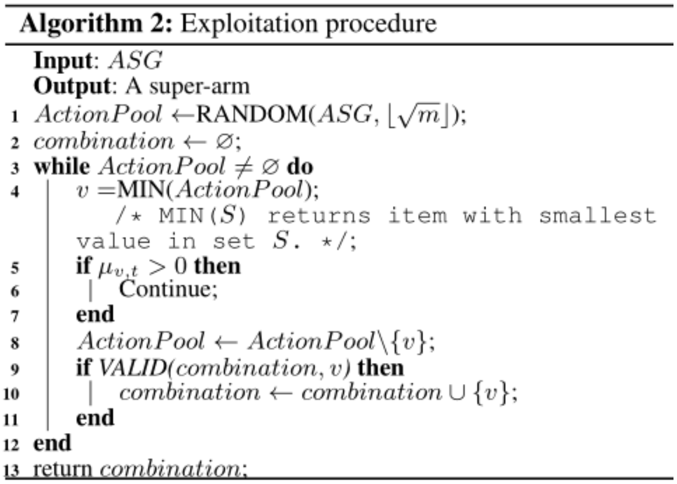


Algorithm 1: Exploration procedure

```
Input: $ASG$
Output: A super-arm
$u \leftarrow$RANDOM($ASG, 1$);
    /* RANDOM($S, n$) returns $n$ random nodes in graph $S$. */;
$combination \leftarrow \{u\}$;
$iteration \leftarrow 1$;
for $v$ in $\Gamma(u)$ do
    $iteration \leftarrow iteration + 1$;
    if $iteration > m$ then
        Break;
    end
    /* VALID($a_1, a_2$) returns a boolean value of whether the combination of vectors $a_1$ and $a_2$ is valid or not. */;
    if VALID($combination, v$) then
        $combination \leftarrow combination \cup \{v\}$;
    end
end
return $combination$;
```

B. Exploitation

The target in exploitation procedure is to select combination with minimum cumulative reward. Regarding ASG, this is a Maximum Weighted Clique like problem in the graph theory.

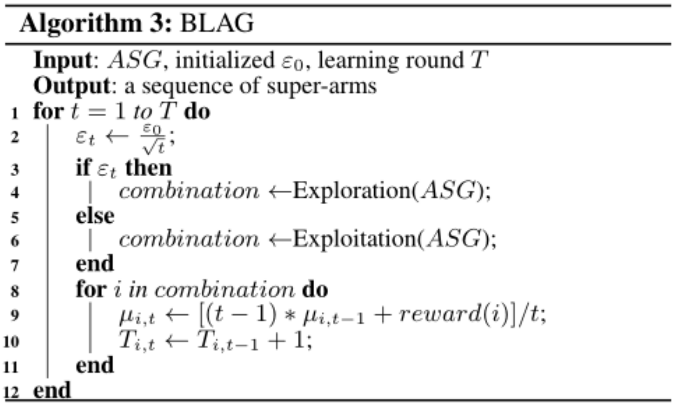


Algorithm 2: Exploitation procedure

```
Input: $ASG$
Output: A super-arm
$ActionPool \leftarrow$RANDOM($ASG, \lfloor\sqrt{m}\rfloor$);
$combination \leftarrow \varnothing$;
while $ActionPool \neq \varnothing$ do
    $v =$MIN($ActionPool$);
        /* MIN($S$) returns item with smallest value in set $S$. */;
    if $\mu_{v,t} > 0$ then
        Continue;
    end
    $ActionPool \leftarrow ActionPool \backslash \{v\}$;
    if VALID($combination, v$) then
        $combination \leftarrow combination \cup \{v\}$;
    end
end
return $combination$;
```

C. Update parameters

Combined with the updating procedure, the whole pseudo code of BLAG is in this part. After jumping out of procedure of forming combination, BLAG has a final procedure of updating estimation of reward and selected time of each nodes selected.



```
Algorithm 3: BLAG
  Input: $ASG$, initialized $\varepsilon_0$, learning round $T$
  Output: a sequence of super-arms
  for $t = 1$ to $T$ do
      $\varepsilon_t \leftarrow \frac{\varepsilon_0}{\sqrt{t}}$;
      if $\varepsilon_t$ then
          $combination$ ←Exploration($ASG$);
      else
          $combination$ ←Exploitation($ASG$);
      end
      for $i$ in $combination$ do
          $\mu_{i,t} \leftarrow [(t-1) * \mu_{i,t-1} + reward(i)]/t$;
          $T_{i,t} \leftarrow T_{i,t-1} + 1$;
      end
  end
```

III. Discussion

A. Time-Limited Regret Bound of BLAG

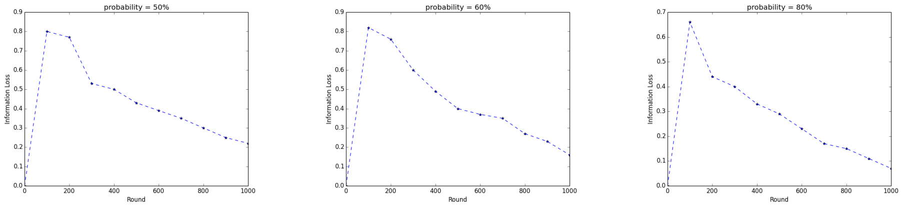
The regret bound of BLAG:

$$E(R_{BLAG}) =$$
$$E[\sum_{t=1} \cdot\vec{D} \cdot \overrightarrow{\Delta\beta^t_{et}} - \alpha T \vec{D} \cdot \overrightarrow{\Delta\beta^*} + \sum_{t=1} \cdot\epsilon_t(\vec{D} \cdot \overrightarrow{\Delta\beta^t_{ep}} - \vec{D} \cdot \overrightarrow{\Delta\beta^t_{et}})]$$
$$\leq 2c\sigma M\sqrt{T} + B^{\times} \sum_{t=1} \cdot\epsilon_t + 1 \leq 2c\sigma M\sqrt{T} + 2B^{\times}\sqrt{T} + 1$$

where $\Delta\beta^t_{ep}$ is the action from exploration, $\Delta\beta^t_{et}$ is the action from exploitation and M is the initial size of ASG.[1]

B. The information loss brought by BLAG

Then, we consider the information loss brought by BLAG. In B.A. graph (short for Barabasi Albert graph), we set n (total number of nodes)= 5K and m (number of edges each new node attaches to existing nodes) = 4 to find the confidence interval of probability.



Form these three figures, we can find that the information loss is decreasing with the increase of the number of round. As to the confidence interval of probability, we speculate that when the probability is larger than 60% and the learning round is more than 1000, the information loss is very small and can be ignored.

IV. REFERENCE

[1]Adaptive Diffusion of Sensitive Information in Social Networks with Partially Known Topology.

[2]Christian Decker and Roger Wattenhofer. Information Propagation in the Bitcoin Network. In 13-th IEEE International Conference on Peer-to-Peer Computing.