Improving Delegated Proof-of-Stake

July 22, 2019

Abstract

1 Introduction

See https://github.com/Luker501/DPoSVoterRepresentation for an introduction to this problem.

2 Notation

We consider an election (N, C) where there is a set of nominators N and a set of candidates C, which we can represent as a bipartite graph $(N \cup C, E)$. There is an edge $(n, c) \in E$ whenever nominator $n \in N$ votes for candidate $c \in C$.

We look to elect $1 \leq k \leq |C|$ candidates into the committee S. To do so, each nominator $i \in N$ provides a list of approved candidates $A_i \subseteq C$ where the size is restricted by a variable x, i.e. $|A_i| \leq x$. We refer to the collection of approval ballots as A, where $\forall n \in N$ then $A_n \in A$. To understand who is elected from these approval ballots, we will use three 'helper' vectors. Firstly, we have a vector $b \in \mathbb{R}^N_{\geq 0}$, recording the *budget* of each nominator, i.e. the amount of cryptocurrency held. Secondly, a vector $w \in \mathbb{R}^C_{\geq 0}$ records the vote *weight* assigned to each candidate. And thirdly a vector $r \in \mathbb{R}^N_{\geq 0}$ records the number of *representatives* each nominator has in the elected committee.

3 Committee Selection Rules

Given an election (N, C) and the nominator's budget b, we define the `PAV-score` of a committee $W \subseteq C$ as:

$$\texttt{PavSc} \;=\; \sum_{i=1}^{|N|} \Big(\sum_{j=1}^{|A_i \cap W|} \frac{1}{j} \Big) \cdot b_i$$

Proportional Approval Voting (PAV) is the committee selection rule that, given an election (N, C) and a committee size k, outputs all size-k committees with the highest `PavSc`; finding a committee in the output of this rule is NP-hard.

4 Committee Quality

Different committees have different possible qualities of representation. To define if a committee represents the electorate well, we introduce the following axioms:

Definition 1. *A committee S of size k satisfies* Justified Representation *if there does not exist a group of nominators $M \subseteq N$ where:*

- $\sum_{m \in M} b_m \geq \sum_{n \in N} b_n$
- $|\bigcap_{m \in M} A_m| \geq 1$, *and*
- *For all $m \in M$, $|S \cap A_m| < 1$.*

The rationale behind the justified representation definition is that if k candidates are to be selected, then intuitively, each group of nominators that has a budget of $\frac{\sum_{n \in N} b_n}{k}$ deserves a representative. So this group should not be completely unrepresented.

We can also generalise justified representation into the following axiom:

Definition 2. *A committee S of size k satisfies* Extended Justified Representation *if for any integer $1 \leq j \leq k$, there does not exist a group of nominators $M \subseteq N$ where:*

- $\sum_{m \in M} b_m \geq \frac{j}{k} \cdot \sum_{n \in N} b_n$
- $|\bigcap_{m \in M} A_m| \geq j$, *and*
- *For all $m \in M$, $|S \cap A_m| < j$.*

The rationale behind the extended justified representation definition is that if k candidates are to be selected, then intuitively, each group of nominators that has a budget of $\ell \cdot \frac{\sum_{n \in N} b_n}{k}$ deserves ℓ representatives. So there should be a member of this group with at least ℓ representatives.

5 Local Search - PAV Algorithm

We have currently made the heuristic decision to start the algorithm from the committee that contains the block producer candidates with the most votes and generate new committees by replacing the least popular block producer candidate first. This starting point is subject to change because having a fixed starting committee probably means that block producer candidates would still be able to strategically vote (perhaps using vote buying). This issue will be tackled by replacing the fixed committee starting point with a randomly chosen committee and deciding on a method to compare different returned committees produced by the algorithm. Given different starting points, the algorithm could return different committees due to: (a) the algorithm's approximate nature; and (b) it is possible that more than one committee could have the same score.

Algorithm 1: The Local Search Proportional (LS-PAV) Algorithm

function `LS-PAV (N, C, A, S, b)` **returns** a committee S;
Input: $\langle N, C, A, S, b\rangle$, $S \subseteq C$, $b \in \mathbb{R}^N_{\geq 0}$; where N is the set of nominators, C is the set of candidates, A is the set of approval ballots, S is the currently selected committee (indexed from $0, ..., |S|-1$) and b is the vector of nominator budgets.
Output: $\langle S^*\rangle$; where S^* is the definitive selected committee.
begin:
double $\texttt{PavSc} = 0.0$; //initialising PAV score counter
vector r, w; //declaring helping vectors: representatives and vote weight
bipartite graph $G := (N \cup C, E)$; //saves the nominator/candidate relationships
int $\texttt{SPointer} := |S| - 1$; //counter that tracks attempts to improve the committee starting from the end. Everytime we replace someone, the algorithm goes back to the end (i.e. back to $|S|-1$).
for all $(n \in N)$ **do**
  for all $(c \in A_n)$ **do**
    add edge (n, c) **in** E; //connect each candidate with it's nominators in the graph
    $w_c = w_c + b_n$; //update the maximum vote weight of a candidate
  end for
  $r_n := |S \cap A_n|$; //the number of current representatives for each nominator
  $\texttt{PavSc} := \texttt{PavSc} + ((\sum_{q=1}^{r_n} \frac{1}{q}) \cdot b_n)$; //update current PAV score
end for
//now we have the PAV score of the most popular committee
//Now we attempt to find new committees that improve the PAV score
while $(\texttt{SPointer} >= 0)$ **do**
  $c^- = S_{|\texttt{SPointer}|}$; //the candidate to remove
  double $\texttt{lostPavSc} := 0.0$; //the PAV score lost by removing a candidate
  vector $r' = r$; //keep a temporary number of representatives r': if we find a worst committee, we will just keep r instead r'
  for all $((n, c^-) \in E)$ **do**
    $\texttt{lostPavSc} := \texttt{lostPavSc} + (\frac{1}{r_n} \cdot b_n)$; //updating the lost PAV score
    $r'_n - -$; //updating the temporary vector
  end for
  for all $(c^+ \in C \backslash S$, where $w_{c^+} \geq \texttt{lostPavSc} + \frac{|N|}{|S|^2})$ **do**
    double $\texttt{addedPavSc} := 0.0$; //PAV score gained by adding this candidate
    for all $(n, c^+) \in E$ **do**
      $\texttt{addedPavSc} := \texttt{addedPavSc} + (\frac{1}{r'_n + 1} \cdot b_n)$; //updating the added PAV
      $r'_n + +$; //updating the temporary vector
    end for
    if $(\texttt{PavSc} - \texttt{lostPavSc} + \texttt{addedPavSc} \geq \texttt{PavSc} + \frac{|N|}{|S|^2})$ **then**
      $S := (S \backslash \{c^-\}) \cup \{c^+\}$; $r := r'$; //save the improved committee
      $\texttt{PavSc} := \texttt{PavSc} - \texttt{lostPavSc} + \texttt{addedPavSc}$; //update the score
      $\texttt{SPointer} := |S|$; //counter is reset to check again
      break for;
    end if
  end for
  $\texttt{SPointer}--$; //continue the search to look for an elected candidate to replace
end while
return $\langle S\rangle$;
end.