Gerrymandering under Uncertain Preferences

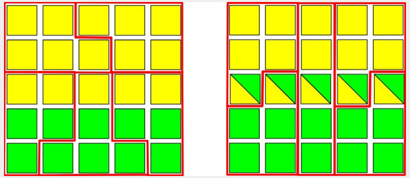
Benjamin Kelly
Rensselaer Polytechnic Institute
kellyb3@rpi.edu

Abstract

Gerrymandering is the manipulating of redistricting for political gain. While many attempts to formalize and model gerrymandering have been made, the assumption of known voter preference, or *perfect information*, limits the applicability of these works to model real world scenarios. To more accurately reason about gerrymandering we investigate how to adapt existing models of the problem to work with *imperfect information*. In our work, we formalize a definition of the gerrymandering problem under probabilistic voter preferences, reason about its complexity compared to the deterministic version, and propose a greedy algorithm to approximate the problem in polynomial time under certain conditions.

Motivation

- ➔ Many “swing” voters don’t have known, constant preferences
- ➔ Can we incorporate this uncertainty into a formal definition of gerrymandering?



Example of a potential district assignment for deterministic voters and non-deterministic voters

Problem Definition

- ➔ We defined the problem over a graph of voters, **G = (V, E)**
- ➔ We ask if it is possible to partition **G** into connected components (districts) subject to certain conditions
  - ◆ Does a given candidate win at least a certain number of districts at least a given likelihood?
  - ◆ Does a given candidate lose at most a certain number of districts with a given likelihood?
  - ◆ The ratio of the size of the largest district and smallest district must not exceed a parameter, **r**

Complexity

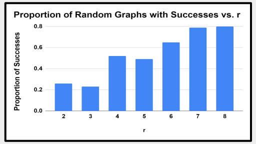
- ➔ The problem is in general NP-Hard
- ➔ With voter weight bounded by *poly*(|**V**|) and candidate number constant, the problem is NP-Complete
- ➔ We developed a greedy algorithm to approximate solutions in polynomial time for the bounded case
  - ◆ Start with a graph of voters with no edges
  - ◆ Greedily add the edge that gives the given candidate the highest chance of winning at least the desired proportion of given districts

Testing of Algorithm

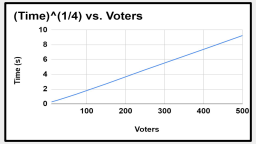
- ➔ We created voters with a 2D coordinate as its “trait” location.
  - ◆ Candidates also have a trait location
  - ◆ A voter’s weight for supporting a candidate is inversely proportional to distance from candidate
  - ◆ The Plackett-Luce model was used to then assign probabilities to preference profiles for each voter
- ➔ The voters were then connected into a graph
  - ◆ Edges were created randomly with a given expected degree for each voter in the graph
  - ◆ Voters close to each other in preference more likely to be connected
    - ● Models people of similar opinions often living near each other
    - ● Effect controlled by a *homophily factor* from 0 to 1

Results

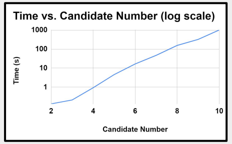
- ➔ We tested our greedy algorithm to analyze both its success rate and its runtime.
- ➔ We confirmed the algorithm scales in polynomial time with increased voters, and exponentially with increased candidates
- ➔ Increasing **r** had a great effect on the success rate of the algorithm



r value vs. against the proportion of graphs where a successful solution was found



Voter count vs the fourth root of runtime (2 candidates)



Candidate number vs runtime (log scale) (100 voters)