Universality in voting behavior: an ABM to calculate the distribution of votes among the candidates

Enrico Palumbo

June 17, 2014

1 Introduction

1.1 Why universality?

One of the most problematic issues, which has always interested scientists of a diversity of disciplines, is whether human behavior is more determined by our nature or our culture. It is sufficiently evident that what we are and how we interact with each other is a result of our experiences and of the sociocultural background in which we grow up. On the other hand, even the most fanatic empirist would admit that our biology plays a strong role in how we relate to the others. Indeed, we can claim that, as human beings, we all share features which are a consequence of the evolution of our species and which cannot be ignored when we study social systems. These genetic common features give birth to behaviors which are common to the most part of the populations of the world (development of a language, smiles to show happiness, crying to show fear or sorrow, yelling to give alarms...) as I have studied in the course of Prof.Adenzato, "Dal cervello sociale all'emergere della società" held for the students of the "Scuola degli studi superiori dell'università di Torino".
From these considerations we may infer that it is reasonable to try to look for universal patterns in social systems and in this direction goes the research of Santo Fortunato and Claudio Castellano "Scaling and universality in proportional elections" (Phys. Rev. Lett. 99, 138701 (2007), Ref.[1]). In particular, in this paper the authors make a step forward and try to use quantitative methods belonging to statistical physics to show that universality can be found in the distribution of votes among the political candidates. The study regards proportional elections with open lists and multiple-seats constituencies, where the electors can express preferences and the party plays no role in the selection of the candidates.

1.2 Scaling and universality in proportional elections

In this paragraph we try to briefly summarize the results that Fortunato and Castellano have obtained in [1].
In the article the authors study the shape of the distribution of votes among the candidates to find regularities among elections in different countries and years. Several papers have already been written about this topic, but the main novelty introduced in [1] follows from the observation that the popularity of a candidate, namely the number of votes she receives ν, depends on two distinc factors: the appeal of the candidate herself and that of the party she belongs to. They quantify the popularity of the party by the total numbers of votes received by the party N divided by the number of candidates in the list Q, $\nu_0 = \frac{N}{Q}$, i.e. the average number of votes received by a candidate. Therefore, if we simply analysed all the candidates together, we may hide some regularities in the distribution of votes, since a popular candidate of an unpopular party may obtain the same number of votes of an unpopular candidate of a popular party. Since we want the purest possible measure of the social mechanism which leads to the distribution of votes, Fortunato and Castellano decided to rule out the role of parties by separating the candidates corresponding to parties with different popularity. This concept is illustrated in fig.1. Different colors in fig.1 represent different parties with different degrees of popularity. The distributions appear to have a similar shape, but stronger parties' are shifted rightwards, since they have a greater average number of votes. It is clear that the distribution does not depend on N and Q separately, but just on the ratio $\nu_0 = \frac{N}{Q}$, i.e. the popularity of the party. Data corresponding to parties with different N and Q, but same ratio ν_0 show an evident overlap, $\mathrm{P}(\nu, N, Q) = \mathrm{P}(\nu, N/Q)$.

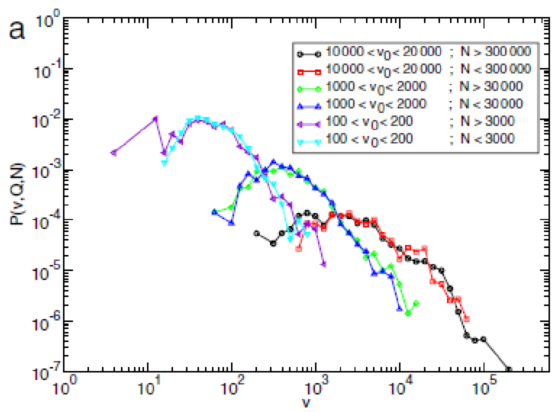

Figure 1: *Distribution of number of votes received by candidates. Curves with the same ν_0 and different N and Q show a remarkable overlap.*

Furthermore, in [1] the authors have shown that the distributions do not depend on ν and ν_0, but better on their ratio $\frac{\nu}{\nu_0}$. This can be interpreted stating that the variable worth representing is not the number of votes a candidate receives ν, but better this number divided by the popularity of the party ν_0.

If we represent the data in function of this new rescaled variable $\frac{\nu}{\nu_0}$ we obtain the pattern of fig.2, $\mathrm{P}(\nu, N/Q) = \mathrm{F}(\frac{\nu}{N/Q})$

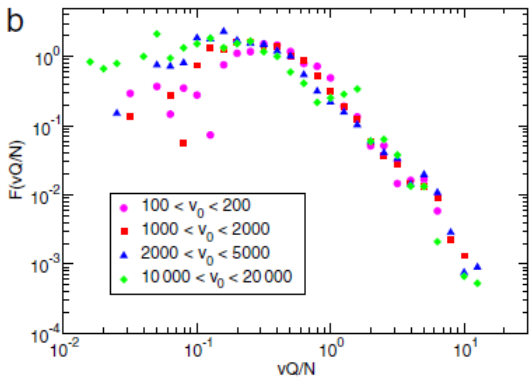

Figure 2: *Distribution of number votes received by candidates, in function of $\frac{\nu}{\nu_0}$. After the role of parties has been ruled out by the scaling, the different curves collapse into one.*

These data are referred to Italian parlamentary elections of 1972, but similar results have been obtained for Poland (2005), Finland (2003) and again Italy (1958, 1987). They all fit well to a lognormal curve and the authors were also able to reproduce the distribution by means of computer simulation.

Further studies (*Chatterjee, Mitrovic, Fortunato "Universality in voting behavior: an empirical analysis" Sci. Rep. 3, 1049; DOI:10.1038/srep01049 (2013),Ref.[2]*) have confirmed the universal pattern also for other years and countries (Denmark, Estonia), as it is showed in fig.3.

Discrepancies from the universal pattern were also found in [2], but they are always associated to differences in electoral rules.

The agreement among data and their regularity is truly remarkable, despite the great changes which have occurred in the sociocultural background of these elections.

We can therefore infer that the spreading of the popularity of a candidate cannot depend on the particular cultural, economic, social features of the country, but just on elementary and deep decision making mechanisms of our brains.

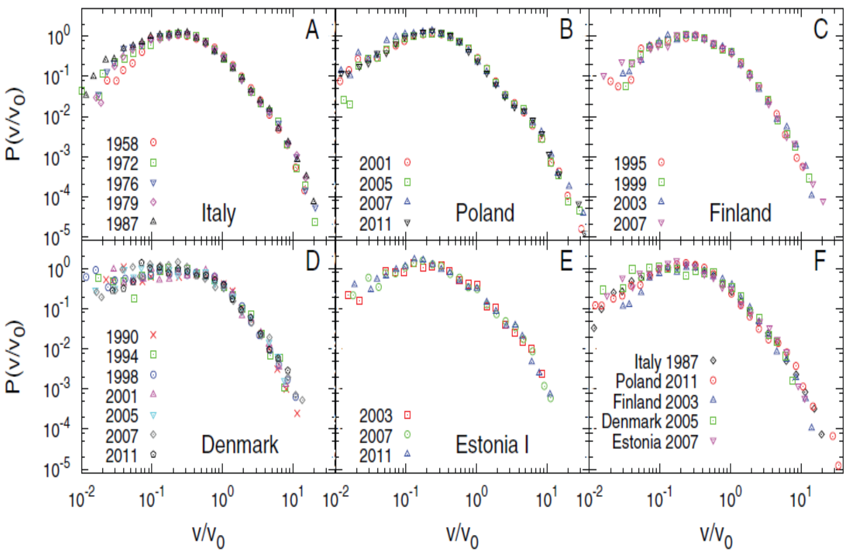

Figure 3: *Frequency of votes received by candidates, in function of* $\frac{\nu}{\nu_0}$

2 The model

Note: in this presentation only the most relevant part of the codes are presented. For details check the files "Distribution of votes.nlogo" for the simulation and "Data analysis cohesion= 10.ipynb", "Data analysis cohesion= 500.ipynb" for the data analysis.

We have built an ABM (Agent Based Model) inspired to that of [1], using Netlogo. Netlogo is a free software which provides a multi-agent programmable modeling environment designed and authored by Uri Wilensky, director of Northwestern University's Center for Connected Learning and Computer-Based Modeling [3]. By means of this powerful software we have simulated the electoral campaign of the candidates of a single party (N and Q are fixed for each simulation) in order to study the distribution of votes among the candidates and compare the results to those obtained in [1] and [2].
The simulation built in [1] is pretty simple: candidates try to convince their acquaintances, with a fixed probability p; the convinced acquaintances become activists and try to convince their acquaintances, until everybody has made up his mind. Every person has a different number of acquaintances, determined by a broad distribution, specifically a power law. Our model took inspiration to what has been done in [1], but it introduces significant and realistic novelties.

This is basically what happens in the model:

1) Electors and candidates are represented by patches of Netlogo's world;

2) At the beginning, candidates are defined as random placed patches (fig.4). Every candidate starts with just his own vote. Every candidate has a different basin of potential supporters and a different "strength", namely the capacity to convince voters;

```
patches-own [vote group  strength]
globals [list-of-candidates list-of-votes]

...
to setup-patches

;; define the undecided voters


ask patches [set vote 0

  set strength 0

  set pcolor white


  set group random n-of-groups ;;every patch belongs to a group

  ]

;;define the candidates

...

while [i <= candidates] [ ask one-of patches with [vote = 0] [set vote i ;;n of the candidate

...

    set strength ( random-float 1) ;; every candidate has a different strength

...
```

3) Social networks exist among voters. Every voter belongs to a group, representing his friends, colleagues, family. Friends tend to imitate each other, as suggested in the paper of Epstein J., Axtell R. "Coordination in Transient Social Networks: An Agent-Based Computational Model of

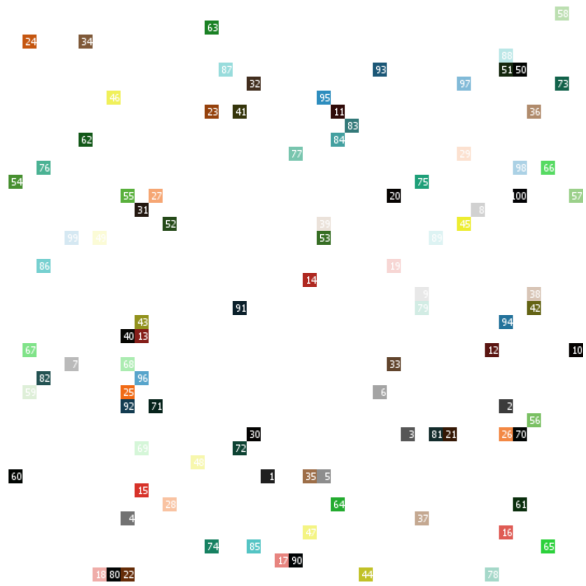

Figure 4: *Candidates are the colored patches, the world is mainly undecided (white)*

the Timing of Retirement" (In "Behavioral Dimensions of Retirement Economics", edited by Henry Aaron, 161-186, Ref.[4]). The number of groups present in the simulation can be controlled by the slider "n-of-groups";

```
...
  set group random n-of-groups ;;every patch belongs to a group
...
```

4) Candidates start to convince their undecided neighbors. If they succeed, the convinced neighbors become activists and start convincing their undecided neighbors and so on (fig.5). The probability of success of a candidate (or of an activist, who simply assumes the features of the supported candidate) depends on three factors: the strength of the candidate, the fraction of friends of the undecided voter who supports the candidate, the fraction of votes the candidate already has. The first factor models the ability of persuasion, which depends on the appeal of the candidate (or on the passion of the activist), the second deals with the imitating behavior among friends, the third represents a rich-gets-richer mechanism, which we believe to be realistic in these kinds of phenomena, since more popular candidates will get easier access to information media and people are usually not willing to support lost causes. The parameter named "cohesion" regulates the trade-off among these factors: the smaller its value, the weaker the interaction among social groups, the stronger is the rich-gets-richer mechanism (consenquently we will observe a more unequal distribution of votes, fig.6).

The value of "cohesion" can be easily changed using a slider;

```
...

ask patches with [vote != 0] [convince] ;; candidates and activists try to convince

...
to convince [

let pvote vote ;; the number of the candidate the activist votes

let pstrength strength ;; the strength of the candidate(or of the activist
;; which assumed the strength of the candidate he or she supports)

let nvote item (pvote - 1) list-of-votes ;; the number of votes received by
;;the candidate(or by the candidated supported by the activist)

let totvotes sum list-of-votes ;; total number of votes


if any? neighbors with [vote = 0] ;;only undecided voters can be convinced

[ask one-of neighbors with [vote = 0]

 [ let mygroup group ;; group of the undecided voter

   let group-decision 1000 * cohesion * (count patches with [group = mygroup and vote = pvote] + 0.1)
   / count patches with [group = mygroup]
   ;;the fraction of votes of friends who support the candidate

   ;;note: 1000 is a scale constant, +0.1 is introduced to avoid probability 0 of convincing

   if random-float 1 <= pstrength * (nvote / totvotes) * group-decision ;;the probability
   ;;of convincing is: strength of the activist * popoularity of the candidate * friends decision

   [set vote pvote ;;undecided voter becomes an activist

   set strength pstrength ;;she gets the strength of the candidate they support

...
```

5) At every iteration we plot the bar chart of votes received by the candidates, the number of undecided voters, the bar chart of the frequency of candidates with a certain number of votes. The latter is the one we will utilise in our analysis (fig.7);

6) The simulation stops when there are no more undecided voters (fig.8).

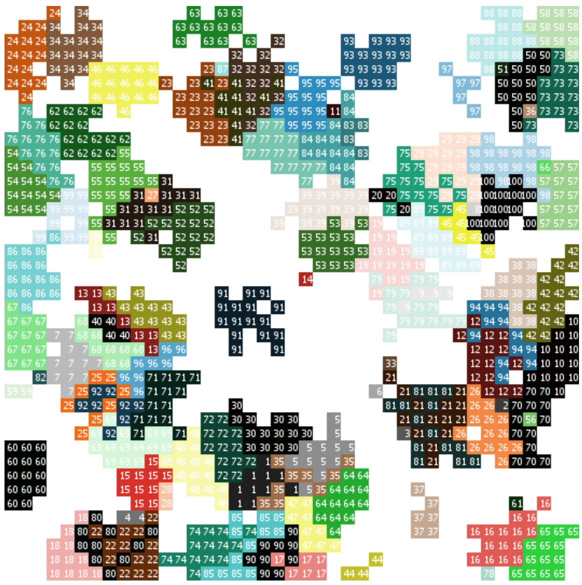

Figure 5: *Candidates expand their popularity, cohesion = 500*

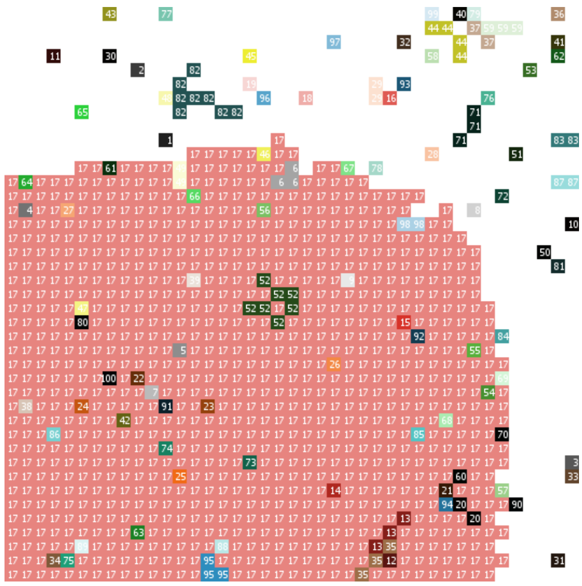

Figure 6: *Candidates expand their popularity, cohesion = 1, the rich-gets-richer mechanism becomes evident*

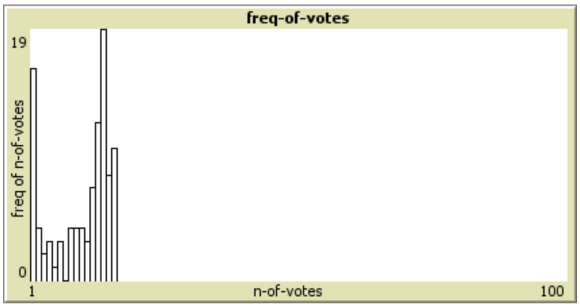

Figure 7: *Histogram corresponding to the $P(\nu,\ N,\ Q)$*

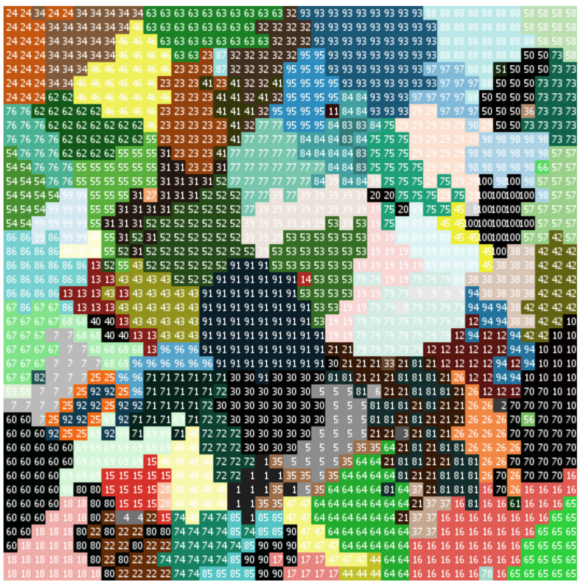

Figure 8: *Every voter has decided*

3 The experimental plan

Our aim is to analyse the bar chart of votes P(ν, N, Q) and compare it with the results obtained in [1] and [2]. In order to do this, we have used Behavior Space and IPython [5] for the data analysis. The values of the parameters corresponding to the simulations we have made are reported in tab.1. We control the total number of electors of the party N changing the size of Netlogo's world and the number of candidates of the party Q by means of a slider named "candidates".

Sim	N	Q	cohesion	N/Q
1	2809	35	10	80
2	3721	35	10	80
3	4225	53	10	80
4	2809	100	500	28
5	3249	116	500	28
6	3721	133	500	28
7	6561	80	500	82
8	7921	97	500	82
9	5929	72	500	82
10	1369	150	500	9
11	1089	121	500	9
12	1681	187	500	9

Table 1: *Simulations*

Behavior Space is a tool of Netlogo which allows to perform multiple experiments from different initial random conditions of the same model. For each one of the simulations in tab.1, we have done a hundred experiments. In the Netlogo's code, at the end of the simulation we have inserted the "export-plot" function in order to export the points of the bar chart P(ν, N, Q) we want to analyse.

```
...
export-plot "freq-of-votes" (word "freq-of-votes" runs ".txt")
...
```

"Runs" is a so called "dummy variable" that we put in the name of the export file. Using Behavior Space we made the slider go from 1 to 100, avoiding the overwriting of the export files (Netlogo outputs freq-of-votes1, freq-of-votes2...).
Once we obtained, for each simulation, a hundred files indicating the (x,y) of the plot P(ν, N, Q), we have used IPython to read and analyse them. For each simulation of tab.1, we have put the data corresponding to the 100 experiments into a Python dictionary. For every number of votes we had a list of values, corresponding to the different experiments, which we averaged in

order to reduce the fluctuations owed to the stochastic nature of our model. The result, named "meanhistogram" in the following code, is our P(ν, N, Q).

Caveat: if you want to use this code, remember to put Netlogo's output files in the same folder of the Python's program

```
...
histograms1 = col.defaultdict(list) #histograms is a dict which has value = list by default


f = [open("./freq-of-votes1/freq-of-votes%d.txt" % i, "rU") for i in range(1, 101)] #open files of the experiment
N1 = 2809 #total number of electors of the party
Q1 = 35 #total number of candidates of the party
for i in f: #for every of the 100 files
    var = 0
    for line in i: #for every line in the file

        var += 1

        if var >= 18: #first 17 lines are comments

             s = line.strip().split(',') #erase the spaces before and after, split where there's a comma

             s[0] = s[0].replace( '"', ' ') #erase the ""
             s[0] = int(s[0]) #the key is the first valule
             s[1] = s[1].replace( '"', ' ')
             s[1] = int(s[1]) #the value is the second value

             histograms1[s[0]].append(s[1]) #associate the key to the value

for j in f: #close the file
    j.close()

...
meanhistograms = dict()

for i in histograms1.iterkeys(): #for every number of votes

        meanhistograms[i] = (array(histograms1[i]).mean())/Q1 #average n of candidates with that
        # n of votes, normalized by total number of candidates


...
```

4 The results

4.1 The distribution of votes

The first thing to check is the shape of the distribution of votes. We have performed the analyses using two different values of the parameter "cohesion" and obtained very different results (keeping number of groups = 10). For the first three simulations of tab.1, corresponding to a value of cohesion = 10, the model delivers very broad distributions, similar to power laws in the first part and almost stationary in the tails (fig.9). We have plotted the values of the P(ν, N,Q) for three different values of N and Q, keeping costant their ratio, using IPython again. The core of the code consists in copying the keys and the values of the "meanhistogram", corresponding to P(ν, N, Q), into two lists and then using the function plot.

```
x=[]
y=[]
for i in sorted (meanhistograms.iteritems(),key = itemgetter(0)):#the histogram must be ordered
# iteritems = (key, value), itemgetter(0) means that we sort the keys and not the values
    x.append(i[0]) #key
    y.append(i[1]) #value

plot(x,y, 'r')
```

Since this shape is very different from the experimental data of [1] and [2], we have performed the rest of the analyses for a value of cohesion = 500 (see tab.1). As we said before, a higher value of "cohesion" yields a less broad distribution (see fig.5 and fig.6), which in this case seems to be more similar to the behavior observed in [1] and [2] (fig.10).

4.2 The first scaling

Assumed the value of cohesion = 500, the first thing we want to verify is that our histograms do not depend on N and Q separately, but vary with the ratio N/Q, namely that $P(\nu, Q, N) = P(\nu, \frac{N}{Q})$, as observed in the study of Fortunato and Castellano. To do this, we have plotted nine different curves, each one corresponding to different values Q and N, but grouped for the value of the ratio N/Q (see tab.1, from exp 4 to exp 12). What we have observed is that plots belonging to the same ratio appear to follow a common pattern, but it would be a premature conclusion to define it a good overlap (fig.11). In particular, the black and red curves, corresponding to a smaller number of candidates, are very noisy and it is not easy to draw conclusions about their behavior.

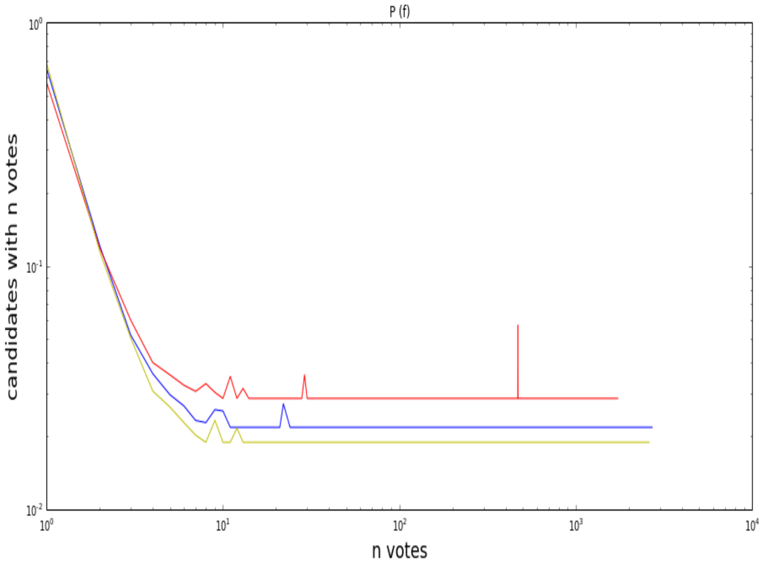

Figure 9: *Distribution of votes for cohesion = 10. Red:* $P(\nu, N = 2809, Q = 35, \nu_0 = 80)$*, Blue:* $P(\nu, N = 3721, Q = 46, \nu_0 = 80)$*, Black:* $P(\nu, N = 4225, Q = 53, \nu_0 = 80)$*. Curves resemble power laws in the first part and are almost stationary in the tails.*

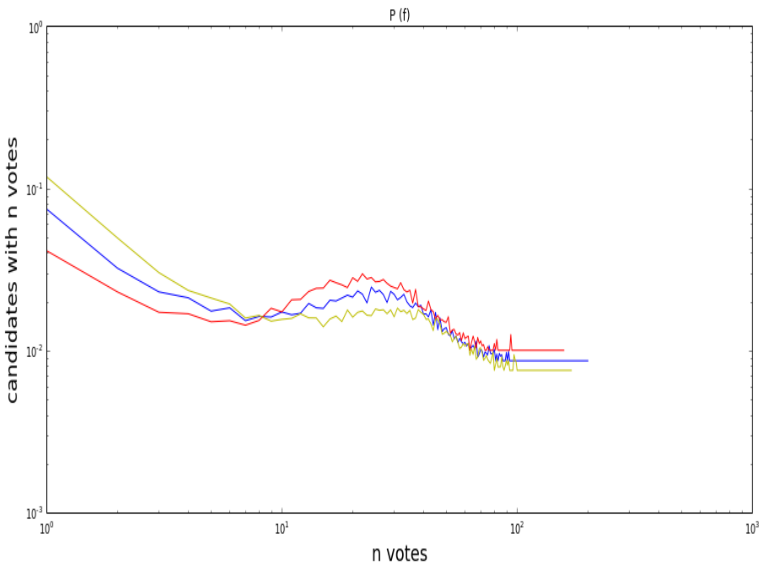

Figure 10: *Distribution of votes for cohesion = 500. Red:* $P(\nu, N = 2809, Q = 100, \nu_0 = 28)$*, Blue:* $P(\nu, N = 3249, Q = 116, \nu_0 = 28)$*, Black:* $P(\nu, N = 3721, Q = 133, \nu_0 = 28)$*. Curves are less broad, they have a peak in the center.*

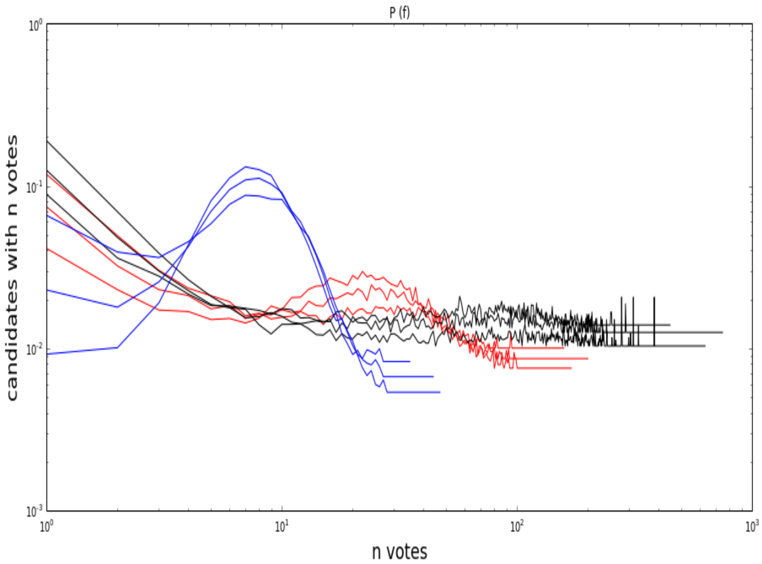

Figure 11: *Distribution of votes for cohesion = 500. Red: N/Q = 28 , Black: N/Q = 82, Blue: N/Q = 9. There is a common pattern for curves of the same color.*

4.3 The convolution and the second scaling

As suggested in the paper of Fortunato and Castellano, the best way to rule out finite size effects on the model is to convolute the curves belonging to the same ratio N/Q. Indeed, although the distribution should be theorically independent from the values of N and Q separately, it is intuitive that this is exactly true only in the limit of N and Q infinite (with N = 10 and Q = 5, you cannot have the same distribution as with N = 10000 and Q= 5000). The convolution of distributions of independent variables is the distribution of the sum of these variables, so this operation is similar to calculate the distribution of the average of random variables having the same ratio $P(\nu(N1, Q1) + \nu(N2, Q2) + \nu(N3, Q3))$ (to have exactly the distribution of the average we should divide by three).

What we obtained after this operation (done using the Numpy function “convolve” in IPython) is quite similar to what happens in fig.1, the black curve corresponds to the maximum average number of votes (N/Q = 82) and it is shifted rightwards, the red curve is in the middle (N/Q = 28) and the blue curve is on the left (N/Q = 9) since it corresponds to the minimum average number of votes (fig.12).

The last operation we have done is to see if the curves of our model do not depend on ν and N/Q independently, but just on the ratio $\frac{\nu}{N/Q}$, i.e. $P(\nu, Q, N) = F(\frac{\nu}{N/Q})$, as observed in [1] and [2]. We can see that the curves cross with each other, there is a decent overlap in the tails of the distribu-

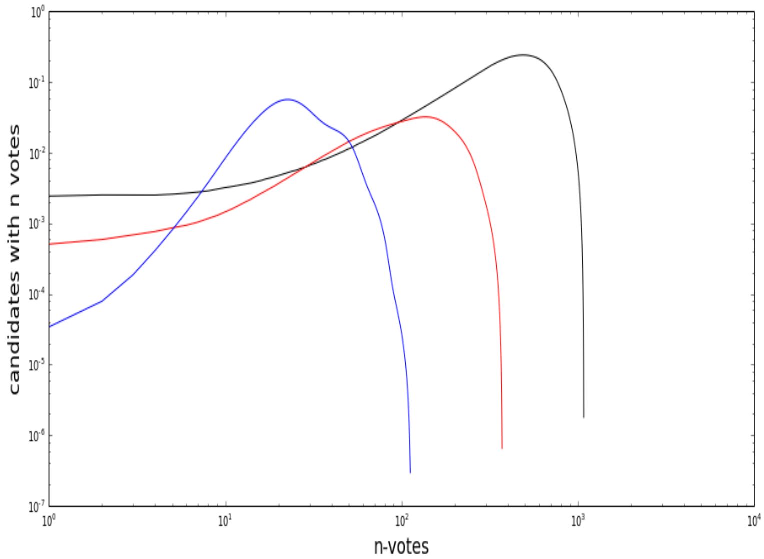

Figure 12: *Convolution of the three curves with: Red: $N/Q = 28$, Black: $N/Q = 82$, Blue: $N/Q = 9$*

tions, but it is not a striking coincidende as in [1] and [2](fig.13).

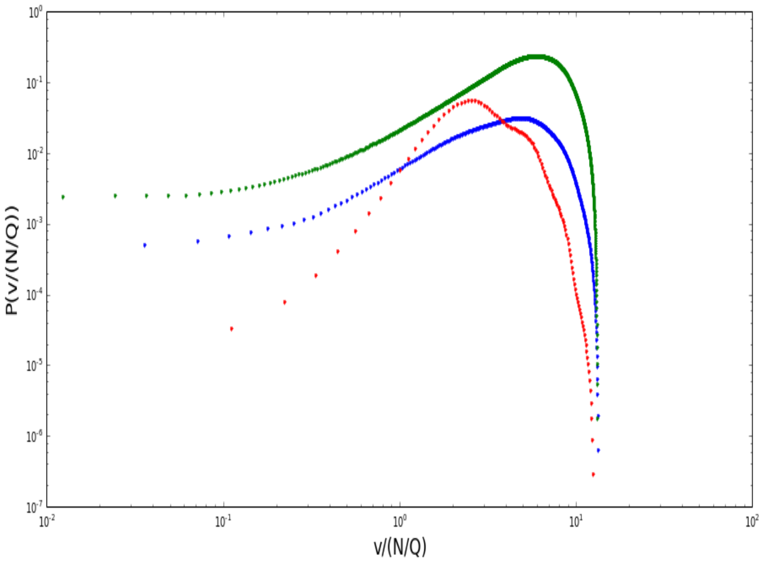

Figure 13: *Frequency of votes received by candidates, in function of* $\frac{\nu}{\nu_0}$

5 Conclusions

In this work we have built an ABM model, inspired to the computer simulation of Fortunato and Castellano. Thanks to the power of ABM models, we could easily insert important and realistic novelties like the existence of relationships among the electors and the fact that people are more likely to be convinced by more popular candidates. Afterwards, we have devolped a practical and automatic way to analyse the data from Netlogo's simulation by Python, without doing the work of cleaning the files by hand. The core of this code, mutatis mutandis, can be reutilised in many different situations. Generally, the results we have obtained from our analysis show a good, but not striking, agreement with the ideas presentend in [1].

These discrepancies might be owed to different causes, but I believe that these are the three most important factors:

1) Parameters: in our study we have explored just two possible values of "cohesion" and just one of the number of groups. Different choices of the parameters might yield better results;

2) Size effects: the numbers N and Q we utilise in our study are, for computational reasons, orders of magnitude inferior to those utilised by Fortunato and Castellano. This may be a problem because it generates noisier and more fluctuating data;

3) Quantitative measuring: our estimations of the similarity of the curves were mainly visual.

Therefore, we suggest that further studies might rule out these factors using the software Behavior Search,a Netlogo's extension which allows to search the entire space of parameters to find the values which best reproduce the results obtained in [1] and [2]. Moreover, more powerful computers or more efficient simulations should be used in order to treat with bigger values of N and Q and finally it should be controlled analytically if the distribution of votes matches well with the log-normal curve and measured the distance between the distributions in the scaling analysis.

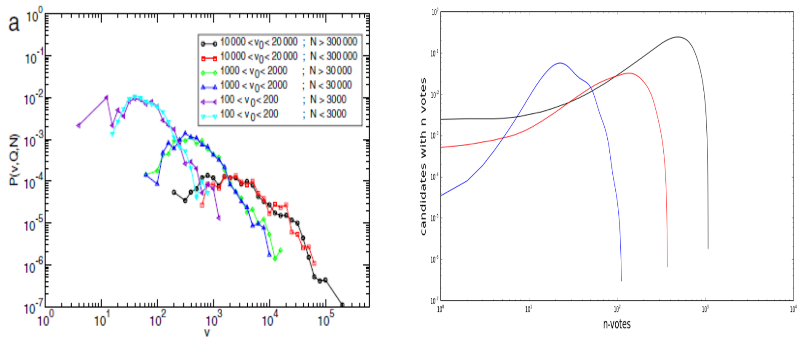

Figure 14: *Comparisons $F(\nu)$, experimental data deliver broader curves, while our results appear to decay faster.*

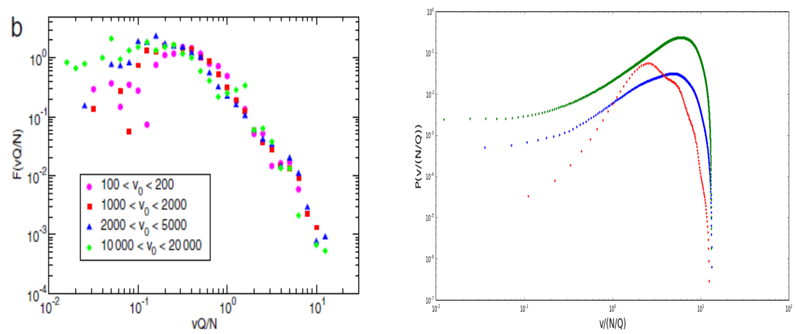

Figure 15: *Comparisons $F(\frac{\nu}{\nu_0})$. The curves of our model appear to follow a common patter, but the overlap is not striking as in the experimental data.*

6 References

[1]: Fortunato, S. , Castellano, C., Scaling and universality in proportional elections. Phys. Rev. Lett. 99, 138701 (2007).
[2]: Chatterjee, A., Mitrovic, M. , Fortunato, S., Universality in voting behavior: an empirical analysis. Sci. Rep. 3, 1049; DOI:10.1038/srep01049 (2013).
[3]: NetLogo itself: Wilensky, U. 1999. NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.
[4]: Epstein, J., Axtell, R. , "Coordination in Transient Social Networks: An Agent-Based Computational Model of the Timing of Retirement" (In "Behavioral Dimensions of Retirement Economics", edited by Henry Aaron, 161-186. New York: Russell Sage / Washington: Brookings Press, 1999) [5]: Fernando Prez, Brian E. Granger, IPython: A System for Interactive Scientific Computing, Computing in Science and Engineering, vol. 9, no. 3, pp. 21-29, May/June 2007, doi:10.1109/MCSE.2007.53. URL: http://ipython.org