Journal of Statistical Software
November 2017, Volume 81, Issue 11. doi: 10.18637/jss.v081.i11

Nash Optimal Party Positions: The nopp R Package

Luigi Curini
University of Milan

Stefano M. Iacus
University of Milan

Abstract

The R package **nopp** enables computing party/candidate ideological positions that correspond to a Nash equilibrium along a one-dimensional space. It accommodates alternative motivations in (each) party strategy while allowing estimation of the uncertainty around their optimal positions through Monte Carlo or bootstrap procedures.

Keywords: spatial theory of voting, Nash equilibrium, voter choice, political science.

1. Introduction

Since Downs' seminal work, the spatial theory of voting has been extremely useful in deepening our understanding of strategic party interactions. However, at least in a multiparty context, the expectations in terms of equilibrium that the theory provides, in particular the high degree of policy convergence (Adams 1999), usually do not fit very well with actual party behavior. Divergence in ideological positions is in fact the rule in the real world of party competition while the cases of spatial "leapfrog" among parties are quite rare (Budge, Klingemann, Volkens, Bara, and Tanenbaum 2001). In this respect, a number of works have recently aimed at filling the gap between the theoretical predictions and the empirical world of party systems. Among the others, Schofield and Sened (2006), Merrill III and Adams (2001) and Adams, Merrill III, and Grofman (2005) have developed models that, by recognizing the importance of non-policy variables in affecting voters' behaviors, are able to produce optimal party strategies that appear much more in line with real-world elections.

Following this logic, the R (R Core Team 2017) package **nopp** (Curini and Iacus 2017) computes party/candidate ideological positions along a one-dimensional space that correspond to a Nash equilibrium. It employs a two-step procedure:

1. It uses survey data to estimate individual respondents' vote choice through an empirical model.

2. It uses the parameter estimates of such an empirical model to search for an equilibrium

configuration in party locations. To this end, **nopp** implements the Merrill and Adams (MA) iterative algorithm (Merrill III and Adams 2001; Adams *et al.* 2005).

The main advantages of package **nopp** are the following:

- It is easy to use as it requires only two lines of commands to run the entire procedure.
- It takes advantage of the flexibility of the **mlogit** package (Croissant 2012) with respect to the choice models that can be estimated.
- It accommodates alternative motivations in (each) party strategy.
- It allows estimation of the uncertainty around the optimal position of parties through two different procedures (bootstrap and Monte Carlo).
- It also incorporates the possibility to produce a graphical representation of the findings.

Package **nopp** is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=nopp`.

2. The algorithm

A typical voter utility function with both policy and non-policy factors can be expressed as follows (see Merrill III and Adams 2001; Adams *et al.* 2005; Calvo and Hellwig 2011):

$$U_{ik}(s_k, a) = -a(x_i - s_k)^2 + \boldsymbol{\beta}\mathbf{t}_{ik} + \boldsymbol{\delta}\mathbf{z}_i + \epsilon_{ik}, \tag{1}$$

where U_{ik} is the utility of voter i to vote for party k. With respect to the policy component in U_{ik}, x_i and s_k are the ideal points of elector i and party k's locations on the underlying policy dimension, respectively. The parameter a describes the weight, or salience, of the voter's proximity preference, and $(x_i - s_k)^2$ is a quadratic term measuring the ideological proximity of voter i to party k. With respect to the non-policy component in U_{ik}, $\mathbf{t}_{ik}$ is a vector that describes characteristics of party k related to voter i, such as partisanship or the assessments about the valence endowment of that party, $\mathbf{z}_i$ is a vector of i's individual attributes (e.g., sex, education), and $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ respectively describe the salience of the previous two vectors in the voter's choice. Finally, ϵ_{ik} is a stochastic error term. One plausible assumption (among others; see below) is that the values of ϵ_{ik} are generated independently from a type-I extreme-value distribution; i.e., the assumption that characterizes the conditional logit (CL) model.

The choice model maximizes the random utility function in Equation 1 to estimate the probability a voter i will select party k (i.e., P_{ik}) in a K-party election:

$$P_{ik}(\mathbf{s}, a) = \frac{\exp\{U_{ik}(s_k, a)\}}{\sum_{j=1}^{K} \exp\{U_{ij}(s_j, a)\}}, \quad \forall i, k, \tag{2}$$

where $\mathbf{s}$ is the vector representing the ideological locations of all parties. Merrill III and Adams (2001) show that the random utility model described in Equation 1 and Equation 2 can be used to search for a Nash equilibrium in party locations, i.e., a combination of strategies can be used such that each party k cannot increase its vote share by altering its strategy. Starting

with the estimated model described in Equation 1, the vote share for party k is given by the expected value as follows:

$$EV_k(\mathbf{s}, a) = \sum_i P_{ik}(\mathbf{s}, a). \tag{3}$$

At a Nash equilibrium, the partial derivative of EV_k with respect to s_k must be zero for all k. That is, for all k:

$$\frac{\partial}{\partial s_k} EV_k(\mathbf{s}, a) = 0, \tag{4}$$

where the expected vote share, EV_k is the sum of the probabilities of voting for party k given all parties' strategic locations. From this and solving for s_k, we can obtain the location at which party k maximizes its vote share:

$$s_k^* = \frac{\sum_i P_{ik}(\mathbf{s}, a)\,(1 - P_{ik}(\mathbf{s}, a))\,x_i}{\sum_i P_{ik}(\mathbf{s}, a)\,(1 - P_{ik}(\mathbf{s}, a))}. \tag{5}$$

By iteratively solving for each party's preferred location, MA provide an algorithm[1] to find each party's equilibrium s^*. In particular, the iterative algorithm updates the spatial location of each party in response to changes in the location of all other parties, based on the fixed proximity and nonproximity factors a, $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ as these parameters result from the empirical model of Equation 1. Therefore estimating the equilibrium strategy of party k does not require information about the actual vote choice of the respondents as in Equation 2.

3. An application: The 2006 Italian general election

3.1. First step: The empirical choice model

The data set `italy2006` included in the **nopp** package contains data from the 2006 Italian General Election survey (source: CSES – Comparative Study of Electoral Systems; `http://www.cses.org/`). In this survey respondents were asked to indicate which party they voted for in the 2006 election. The data concerns 5 parties: UL (Ulivo), RC (Communist Refoundation Party), FI (Forza Italia), AN (National Alliance) and UDC (Union of Christian Democrats).

For application, **nopp** requires the data frame to be in a long format (i.e., each row is an alternative) rather than in a wide format (i.e., each row is an observation). The function `set.data()` allows easy transformation of a wide format data frame into a long one. In the case of `italy2006`, the original dataset presents a wide format.

Once the **nopp** package has been installed, for example using `install.packages("nopp")`, it should be loaded by typing:

```
R> library("nopp")
```

The data set `italy2006` is loaded into the R workspace as follows:

```
R> data("italy2006", package = "nopp")
```

[1] See Adams *et al.* (2005, Appendix 4.1) for a proof of the conditions that guarantee the existence and uniqueness of Nash equilibria for each party k.

The overview of the data set can be obtained using the `head` function as follows

```
R> head(italy2006)
```

```
    country id vote self   prox_FI     prox_UL   prox_AN   prox_UDC
1 Italy2006  1   UL    4 -14.19090  -0.2111659 -18.27562  -3.949044
2 Italy2006  2   UL    4 -14.19090  -0.2111659 -18.27562  -3.949044
3 Italy2006  3   UL    3 -22.72506  -0.2921100 -27.82562  -8.923485
4 Italy2006  4   UL    4 -14.19090  -0.2111659 -18.27562  -3.949044
5 Italy2006  5   UL    2 -33.25922  -2.3730540 -39.37562 -15.897925
6 Italy2006  6   RC    0 -60.32755 -12.5349426 -68.47562 -35.846806
       prox_RC partyID_FI partyID_UL partyID_AN partyID_UDC
1 -4.347948551          0          0          0           0
2 -4.347948551          0          1          0           0
3 -1.177601457          0          0          0           0
4 -4.347948551          0          1          0           0
5 -0.007254523          0          1          0           0
6 -3.666560650          0          0          0           0
  partyID_RC sex age education gov_perf
1          0   1   4         1        3
2          0   0   6         0        4
3          0   1   3         1        3
4          0   0   5         0        3
5          0   0   6         4        3
6          1   1   5         2        4
```

which indicates that `italy2006` has the following columns: `id` (respondent identifier), `vote` (the party voted), `self` (self-placement of respondent on a 0 to 10 left-right scale), `prox_*` (ideological distance between the respondent and a party placement estimated as $-(\texttt{self}_i - s_k)^2$, where s_k is the survey respondents' mean party k placement used as party k's actual position), and `partyID_*` (`partyID_* = 1` if the respondent declares to feel herself close to that party, 0 otherwise).

To transform the data frame into a long format, we use a modified version of the `mlogit.data` function from the **mlogit** package called `set.data`. The package **mlogit** is loaded by **nopp** at startup time:

```
R> colnames(italy2006)
```

```
 [1] "country"     "id"          "vote"        "self"
 [5] "prox_FI"     "prox_UL"     "prox_AN"     "prox_UDC"
 [9] "prox_RC"     "partyID_FI"  "partyID_UL"  "partyID_AN"
[13] "partyID_UDC" "partyID_RC"  "sex"         "age"
[17] "education"   "gov_perf"
```

```
R> election <- set.data(italy2006, shape = "wide", choice = "vote",
+    varying = 5:14, sep = "_")
```

The compulsory arguments are `choice`, which is the variable that indicates the choice made (in our case the variable `"vote"`), the `shape` of the original `data.frame` (in our case: `"wide"`) and, if there are alternative specific variables (in our case: `prox_*` and `partyID_*`), the `varying` argument has to be specified as a numeric vector that indicates which columns contain alternative specific variables. This argument is then passed to reshape the original `data.frame` into long format. Further arguments may be passed to control reshaping. For example, if the names of the variables are of the form `var_alt` (as in our case), one should add `sep = "_"`. We can now look at the transformed data set

```
R> head(election)
```

```
        country id  vote self sex age education gov_perf alt
1.AN  Italy2006  1 FALSE    4   1   4         1        3  AN
1.FI  Italy2006  1 FALSE    4   1   4         1        3  FI
1.RC  Italy2006  1 FALSE    4   1   4         1        3  RC
1.UDC Italy2006  1 FALSE    4   1   4         1        3 UDC
1.UL  Italy2006  1  TRUE    4   1   4         1        3  UL
2.AN  Italy2006  2 FALSE    4   0   6         0        4  AN
             prox partyID chid
1.AN  -18.2756214       0    1
1.FI  -14.1908979       0    1
1.RC   -4.3479486       0    1
1.UDC  -3.9490445       0    1
1.UL   -0.2111659       0    1
2.AN  -18.2756214       0    2
```

The result is a `data.frame` in long format with one line for each alternative. The choice variable is now a logical variable and the individual specific variables (`sex`, `age`, `education` and `gov_perf`) are repeated five times. An index attribute is added to the data, which contains the two relevant indexes: `chid`, the choice index, and the `alt` index.

The first step of the analysis consists in the estimation of Equation 1. Let us assume as an illustrative example that the choice of voter i can be represented as a function of two properties connecting voter i to party k (her ideological proximity to party k (`prox`) and her `partyID`) and four individual characteristics: `sex` (equals to 1 for female), `age` (`1` = "18–24 years", `2` = "25–34", `3` = "35–44", `4` = "45–54", `5` = "55–64", `6` = "65+"), `education` (`0` = "up to primary school", `1` = "incomplete secondary", `2` = "secondary completed", `3` = "post-secondary trade", `4` = "university undergraduate degree inc", `5` = "university undergraduate degree comp") and voter's judgment of the incumbent government performance: `gov_perf` (`1` = "very good job", `2` = "good job", `3` = "bad job", `4` = "very bad job"). We also add an intercept for each party to capture all the other unmeasured non-policy sources of voters' party evaluations, including voters' valence-related evaluations of the parties (such as the judgment related to party leaders' competence, integrity, or charisma).

To estimate individual respondents' vote choice package **nopp** uses functionality provided by the **mlogit** package. We run in this first example a conditional logit model (we also select FI – Forza Italia party – as the reference alternative; this is optional):

```
R> m <- mlogit(vote ~ prox + partyID | gov_perf + sex + age + education,
+    election, reflevel = "FI")
```

```
R> summary(m)


Call:
mlogit(formula = vote ~ prox + partyID | gov_perf + sex + age +
    education, data = election, reflevel = "FI", method = "nr",
    print.level = 0)

Frequencies of alternatives:
      FI       AN       RC      UDC       UL
0.242009 0.178082 0.100457 0.077626 0.401826

nr method
7 iterations, 0h:0m:0s
g'(-H)^-1g = 0.000248
successive function values within tolerance limits

Coefficients :
                 Estimate Std. Error t-value  Pr(>|t|)
AN:(intercept)   0.647570   1.262720  0.5128 0.6080651
RC:(intercept)  -4.927774   2.222642 -2.2171 0.0266177 *
UDC:(intercept) -0.996713   1.375212 -0.7248 0.4685926
UL:(intercept)  -3.668816   1.415796 -2.5913 0.0095602 **
prox             0.080615   0.012912  6.2436 4.276e-10 ***
partyID          4.115841   0.389591 10.5645 < 2.2e-16 ***
AN:gov_perf      0.071072   0.410497  0.1731 0.8625446
RC:gov_perf      1.890192   0.628987  3.0051 0.0026546 **
UDC:gov_perf    -0.054156   0.418196 -0.1295 0.8969623
UL:gov_perf      1.415211   0.405426  3.4907 0.0004818 ***
AN:sex          -0.621747   0.481125 -1.2923 0.1962613
RC:sex          -1.501041   0.783660 -1.9154 0.0554385 .
UDC:sex         -0.340194   0.490517 -0.6935 0.4879692
UL:sex          -0.260747   0.482777 -0.5401 0.5891295
AN:age          -0.196572   0.156697 -1.2545 0.2096702
RC:age          -0.243988   0.246333 -0.9905 0.3219400
UDC:age          0.251847   0.161889  1.5557 0.1197853
UL:age           0.139813   0.158847  0.8802 0.3787672
AN:education     0.071070   0.188558  0.3769 0.7062364
RC:education    -0.450525   0.314049 -1.4346 0.1514092
UDC:education    0.169480   0.184193  0.9201 0.3575103
UL:education     0.176575   0.182907  0.9654 0.3343541
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-Likelihood: -243.73
McFadden R^2:  0.61524
Likelihood ratio test : chisq = 779.45 (p.value = < 2.22e-16)
```

As observed, increasing the ideological proximity between voter i and party k increases the probability to vote for that party. A similar result applies if voter i displays a party identification for party k. However, as the judgment about the performance of the center-right incumbent government worsens, so the chance to vote for any of the opposition parties (relative to FI, the party headed by the incumbent Prime Minister Silvio Berlusconi) increases. Finally, given that the constants are all relative to the score of FI, which is normalized to be zero, we can see that there are unmeasured sources of voters' party evaluations that benefit FI in a statistical significant way while penalizing the opposition parties (UL and RC).

Note that by employing **mlogit** we can estimate a large variety of choice models. For example, conditional logit has been criticized in the literature because it imposes the *independence of irrelevant alternatives* (IIA) property on voter choice (meaning that the relative odds of selecting between two parties is independent of the addition or subtraction of other alternatives from the choice set; see Alvarez and Nagler 1998). Dow and Endersby (2004) show that for most applications the IIA assumption is not as restrictive as it might appear. The package **mlogit** allows, however, running a number of models that relax the above IIA assumptions, such as the heteroscedastic logit model, the general extreme value model, the nested logit model and the multinomial probit model.

Alternatively, we could also decide to run a model with party-varying proximity coefficients, rather than with a single proximity coefficient for all parties. Through this we move from a model in which only one parameter for the alternative-specific attribute proximity is estimated – implying that such attribute is valuated identically with regard to all alternatives/parties – to a more general model in which the single generic coefficient for proximity is divided into as many alternative-specific coefficients as there are parties to compete in the election (Ben-Akiva and Lerman 1985). This can be obtained by typing:

```
R> m2 <- mlogit(vote ~ partyID | gov_perf + sex + age + education | prox,
+    election, reflevel = "FI")
R> summary(m2)

Call:
mlogit(formula = vote ~ partyID | gov_perf + sex + age + education |
    prox, data = election, reflevel = "FI", method = "nr", print.level = 0)

Frequencies of alternatives:
      FI       AN       RC      UDC       UL
0.242009 0.178082 0.100457 0.077626 0.401826

nr method
7 iterations, 0h:0m:0s
g'(-H)^-1g = 4.43E-05
successive function values within tolerance limits

Coefficients :
                 Estimate Std. Error t-value  Pr(>|t|)
AN:(intercept)   0.831976   1.267999  0.6561 0.5117384
RC:(intercept)  -5.335890   2.139541 -2.4939 0.0126333 *
UDC:(intercept) -0.715348   1.400122 -0.5109 0.6094083
```

```
UL:(intercept)  -3.125356   1.487789 -2.1007 0.0356698 *
partyID          4.179460   0.395892 10.5571 < 2.2e-16 ***
AN:gov_perf     -0.040633   0.424475 -0.0957 0.9237383
RC:gov_perf      1.758049   0.597427  2.9427 0.0032537 **
UDC:gov_perf    -0.097656   0.432567 -0.2258 0.8213886
UL:gov_perf      1.236409   0.423356  2.9205 0.0034948 **
AN:sex          -0.663719   0.485018 -1.3684 0.1711738
RC:sex          -1.849938   0.788415 -2.3464 0.0189556 *
UDC:sex         -0.452395   0.500718 -0.9035 0.3662654
UL:sex          -0.392115   0.507260 -0.7730 0.4395188
AN:age          -0.215370   0.158295 -1.3606 0.1736538
RC:age          -0.188365   0.242111 -0.7780 0.4365613
UDC:age          0.269604   0.165212  1.6319 0.1027080
UL:age           0.142644   0.165471  0.8620 0.3886622
AN:education     0.066177   0.190695  0.3470 0.7285693
RC:education    -0.391942   0.309933 -1.2646 0.2060148
UDC:education    0.175535   0.186911  0.9391 0.3476613
UL:education     0.202944   0.190928  1.0629 0.2878115
FI:prox          0.103039   0.033629  3.0640 0.0021839 **
AN:prox          0.061046   0.024028  2.5407 0.0110641 *
RC:prox          0.011895   0.020846  0.5706 0.5682712
UDC:prox         0.158010   0.049688  3.1801 0.0014723 **
UL:prox          0.114600   0.030862  3.7133 0.0002046 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-Likelihood: -236.53
McFadden R^2:  0.6266
Likelihood ratio test : chisq = 793.85 (p.value = < 2.22e-16)
```

As observed, now the salience of voter proximity is higher for Ulivo, FI and the centrist UDC, while it is not significant for the radical party RC.

Note that we can also directly estimate a model starting from a wide-format data set by typing

```
R> m3 <- mlogit(vote ~ prox + partyID  | gov_perf + sex + age +
+    education, italy2006, shape = "wide", choice = "vote",
+    varying = 5:14, sep = "_", reflevel = "FI")
```

All the previous models, regardless of their form, can then be passed to **nopp** to estimate the optimal party position configuration.

3.2. Second step: Estimating party optimal positions

Having empirically estimated all parameters of interest related to the individual respondents' vote choices, now we have to feed them into the MA algorithm. To this aim, we have to use the `equilibrium` command in **nopp**. Starting from the previous estimated model `m`, this can be easily performed by typing:

```
R> nash.eq <- equilibrium(model = m, data = election)
```

The mandatory arguments are `model`, which is the **mlogit** model analysis, and `data`, which is the data set (in long format).

```
R> nash.eq

================
Nash equilibrium
================
Party positions:
   FI    UL    AN   UDC    RC
6.408 4.717 6.465 6.291 2.300

Party vote-shares:
   FI    UL    AN   UDC    RC
0.243 0.393 0.183 0.073 0.108
```

nopp returns both the optimal party positions and the vote-shares of parties at that optimal configuration.

The user can also decide to pass to **nopp** a list of *external* party positions (as they arise for example from mass media or expert surveys; in this case, from Benoit and Laver 2006) as well as a list of *external* vote-shares of parties (according to polls or electoral results) and then compare the equilibrium party configuration with the external one. For example, if we decide to use as the external party positions the survey respondents' mean party placements and as the external vote-shares of parties the vote-shares they obtain in the survey, we would type:

```
R> external.pos <- list(FI = 7.76, UL = 3.54, RC = 1.91, AN = 8.27,
+    UDC = 5.99)
R> external.votes <- list(FI = 0.25, UL = 0.38, RC = 0.10, AN = 0.18,
+    UDC = 0.07)
R> nash.eq <- equilibrium(model = m, data = election, pos = external.pos,
+    votes = external.votes)
R> nash.eq

================
External
================
Party positions:
  FI   UL   AN  UDC   RC
7.76 3.54 8.27 5.99 1.91

Party vote-shares:
  FI   UL   AN  UDC   RC
0.25 0.38 0.18 0.07 0.10
```

```
================
Nash equilibrium
================
Party positions:
   FI    UL    AN   UDC    RC
6.408 4.717 6.465 6.291 2.300

Correlation External/Nash: 0.93
Average Absolute Distance: 1.00

Party vote-shares:
   FI    UL    AN   UDC    RC
0.243 0.393 0.183 0.073 0.108

Correlation External/Nash: 1.00
Average Absolute Distance: 0.67%
```

The results make clear that there are marked similarities between actual and optimal positions (average absolute distance from actual party position: 1.00; Pearson-R: .93). Furthermore, the projected vote-shares of parties found in the Nash equilibrium looks remarkably close to actual vote-shares (approximately 0.67% of difference on average). This of course does not imply that all parties are simple-minded vote-maximizers, but still by assuming it we get to a close proxy to the actual (but unobservable) utility function deployed by party leaders in the Italian 2006 electoral context.

nopp is also extremely flexible with respect to alternative motivations in party strategy. For example, two pre-electoral coalitions were competing in the 2006 Italian general election: a center-left coalition (UL and RC) and a center-right coalition (FI, UDC and AN). In such circumstance we could expect that party's utility from the election is not only linked to the vote-share it would obtain but also, to varying degrees, to the electoral success of its own coalition. This could affect party strategy on where to locate in the ideological space during the electoral campaign (Golder 2006).

More formally, the utility that party k belonging to coalition c_j attached to an electoral outcome can be considered as the weighted sum of its expected vote-share (EV_k) and the expected vote of its coalition partners:

$$U_k = EV_k + \alpha \sum_{i=c_j} EV_i, \tag{6}$$

where i are the parties – other than k – belonging to coalition c_j (see Adams *et al.* 2005, p. 113). When $\alpha = 0$, parties are purely vote-seeking (like in the previous scenario), while if α increases parties start to weight both their own votes and those of their coalition partners. At the extreme, when $\alpha = 1$, parties weight their own votes exactly as those of their coalition partners. An example:

```
R> coal1 <- list(FI = 1, UL = 2, RC = 2, AN = 1, UDC = 1)
```

With the above line of command, the parties are assigned to two different coalitions.

```
R> alpha1 <- list(FI = 0.7, UL = 0.8, RC = 0.1, AN = 0.5, UDC = 0.5)
```

With this second line of command, the values of α are defined for each party. Note that α can be different across coalitions as well as within the same coalition.

```
R> nash.eq <- equilibrium(model = m, data = election, coal = coal1,
+    alpha = alpha1)
R> nash.eq

================
Nash equilibrium
================
Party positions:
   FI    UL    AN   UDC    RC
5.581 5.422 6.088 5.603 2.349

Party vote-shares:
   FI    UL    AN   UDC    RC
0.243 0.386 0.186 0.074 0.112
```

Also in this case, we could add to our command a list specifying the external party position and the external vote-shares of parties.

```
R> nash.eq <- equilibrium(model = m, data = election,
+    pos = c(FI = 7.76, UL = 3.54, RC = 1.91, AN = 8.27, UDC = 5.99),
+    votes = c(FI = 0.25, UL = 0.38, RC = 0.10, AN = 0.18, UDC = 0.07),
+    coal = coal1, alpha = alpha1)
R> nash.eq

================
External
================
Party positions:
  FI   UL   AN  UDC   RC
7.76 3.54 8.27 5.99 1.91

Party vote-shares:
  FI   UL   AN  UDC   RC
0.25 0.38 0.18 0.07 0.10


================
Nash equilibrium
================
Party positions:
   FI    UL    AN   UDC    RC
5.581 5.422 6.088 5.603 2.349
```

```
Correlation External/Nash: 0.81
Average Absolute Distance: 1.41

Party vote-shares:
   FI    UL    AN   UDC    RC
0.243 0.386 0.186 0.074 0.112

Correlation External/Nash: 1.00
Average Absolute Distance: 0.69%
```

Compared to the previous scenario in which parties are expected to be just vote-maximizers, an equilibrium in which parties care also about the votes obtained by their own coalition (as depicted above) produces equilibria positions that resemble less well the actual positions of Italian parties during the 2006 election (average absolute distance: 1.00 vs. 1.41)[2].

A party k could also be motivated to maximize its margin relative to party j. The probability that parties act as margin-maximizers rather than vote-maximizers can be due for example to the type of electoral system employed, e.g., plurality vs. proportional systems (Cox 1990). Let us assume that UL is interested in maximizing its margin relative to FI. Then, we would type:

```
R> nash.eq <- equilibrium(model = m, data = election,
+    margin = list(UL = "FI"))
R> nash.eq

================
Nash equilibrium
================
Party positions:
   FI    UL    AN   UDC    RC
6.372 4.937 6.426 6.222 2.311

Party vote-shares:
   FI    UL    AN   UDC    RC
0.243 0.392 0.183 0.073 0.109
```

`margin` is not a symmetric command. Therefore, if both UL and FI are interested maximizing their vote margins relative to each other, we would write:

```
R> nash.eq <- equilibrium(model = m, data = election,
+    margin = list(FI = "UL", UL = "FI"))
R> nash.eq
```

[2]In this sense, one could also infer that the existence of pre-electoral coalitions seems to have played no relevant role in the parties' spatial behaviors during the 2006 Italian electoral stage, given that members of both blocks appear unfocused on how their policy strategies influence the collective appeal of their coalition block (for more on this point see Curini and Iacus 2008).

```
================
Nash equilibrium
================
Party positions:
   FI    UL    AN   UDC    RC
5.925 4.891 6.495 6.349 2.309

Party vote-shares:
   FI    UL    AN   UDC    RC
0.242 0.392 0.184 0.074 0.109
```

Untill now we have imposed no restrictions on party positioning, leaving parties completely free to manipulate their policy images. This is an option that we can restrict in **nopp**. That is, party k can be modeled as being stuck (to a varying degree) to a specific party location for various reasons, for example, fear of losing reputation among electors, uncertainty about popular preferences, activists' concern for established ideology, etc. (Curini and Hino 2105; Budge 1994; Plümper and Martin 2008). When this happens, in the final Nash configuration of party positions, the position of party k can be expressed as: $\texttt{fixed} \cdot (1 - \gamma) + \texttt{nash} \cdot \gamma$, where `nash` is the optimal party position when that party is free to move and `fixed` is the "stuck" party position.

When $\gamma = 1$, a party's policy choices can span with no spatial limits. When $\gamma = 0$, a party is fixed to the spatial position specified by the user. When $0 < \gamma < 1$, the final position of party k will depend on both considerations (maximizing its vote-shares on one hand and the concern to protect its ideological reputation on the other). For example, let us fix RC party at 1.95 and assume that γ is equal to 0.5 for that party. Then, we would type:

```
R> nash.eq <- equilibrium(model = m, data = election,
+    fixed = list(RC = 1.95), gamma = 0.5)
R> nash.eq

================
Nash equilibrium
================
Party positions:
   FI    UL    AN   UDC    RC
6.409 4.726 6.466 6.291 2.094

Party vote-shares:
   FI    UL    AN   UDC    RC
0.243 0.393 0.183 0.073 0.108
```

Notice that through **nopp** we can also easily combine different party motivations in a variety of ways. Below we report two hypothetical scenarios:

Example 1:

```
R> nash.eq <- equilibrium(model = m, data = election,
+    margin = list(FI = "UL", UL = "FI"),
+    fixed = list(RC = 1), gamma = 0.2)
```

Example 2:

```
R> coal1 <- list(FI = 1, UL = 2, RC = 2, AN = 1, UDC = 1)
R> alpha1 <- list(FI = 0.7, UL = 0.8, RC = 0.5, AN = 0.5, UDC = 0.5)
R> nash.eq <- equilibrium(model = m, data = election, coal = coal1,
+    alpha = alpha1, fixed = list(RC = 1), gamma = 0.6)
```

3.3. Linear utility function

In Equation 1 we have assumed a quadratic utility function for the voters with respect to the policy component of U_{ik}. An alternative would be to assume a linear utility function, i.e.,

$$U_{ik}(s_k, a) = -a|x_i - s_k| + \boldsymbol{\beta}\mathbf{t}_{ik} + \boldsymbol{\delta}\mathbf{z}_i + \epsilon_{ik}. \tag{7}$$

In the `italy2006.lin` dataset included in **nopp** the ideological proximity variable (labelled `proxlin_*`) has been computed as $-|\texttt{self}_i - s_k|$. This variable allows the estimation of the empirical model by employing a linear utility function. From the empirical model, we can do the same with respect to the computation of the Nash equilibrium in party locations. Once reshaped in a long format `italy2006.lin`, we would type two lines of commands to run the entire procedure:

```
R> data("italy2006.lin", package = "nopp")
R> election3 <- set.data(italy2006.lin, shape = "wide",
+    choice = "vote", varying = 5:14, sep = "_")
R> m3 <- mlogit(vote ~ proxlin + partyID | gov_perf + sex + age +
+    education, election3, reflevel = "FI")
R> nash.eq3 <- equilibrium(model = m3, data = election3,
+    quadratic = FALSE)
```

The `FALSE` option after the `quadratic` option tells **nopp** to adopt the MA algorithm for linear utility

```
R> nash.eq3

================
Nash equilibrium
================
Party positions:
   FI    UL    AN   UDC    RC
6.913 4.335 7.140 5.978 2.125

Party vote-shares:
   FI    UL    AN   UDC    RC
0.242 0.402 0.178 0.078 0.100
```

3.4. Estimating uncertainty

The **nopp** package can also estimate the confidence intervals around the optimal party positions (as well as the party vote-shares at that optimal configuration) in two different (and mutually excludable) ways:

(1) By means of a bootstrap procedure. The observations are resampled and the Nash equilibrium is re-evaluated. The bootstrap procedure is as follows:

1. Estimate the statistical model m with coefficient matrix $\mathbf{b}$ using the original data, $\mathbf{D}$.
2. Draw $\mathbf{D}^*$ randomly with replacement from the original data $\mathbf{D}$.
3. Re-estimate anew the model m^* with coefficient matrix $\mathbf{b}^*$ on the new data $\mathbf{D}^*$.
4. Calculate Nash-optimal party placements using $\mathbf{b}^*$ and $\mathbf{D}^*$.
5. Repeat steps 2–4, R times.

(2) By means of a Monte Carlo simulation starting from the empirical estimation. The coefficients of the model fitted with **mlogit** are resampled independently from their asymptotic Gaussian distribution and the Nash equilibrium is re-evaluated each time. Compared to the former method, this alternative appears as more "Bayesian" in spirit, given that it takes the estimated coefficients of the model fitted with **mlogit** as its priors before simulation. The Monte Carlo method implemented in **nopp** is as follows:

1. Estimate the statistical model m with coefficient matrix $\mathbf{b}$ and variance-covariance matrix $\mathbf{V}$ using the original data, $\mathbf{D}$.
2. Draw new coefficients $\mathbf{b}^*$ from the Gaussian distribution $\mathcal{N}(\mathbf{b}, \mathbf{V})$.
3. Calculate Nash-optimal party placements using $\mathbf{b}^*$ and the original data $\mathbf{D}$.
4. Repeat steps 2–3, R times.

The above strategies are in line with what is suggested in King, Tomz, and Wittenberg (2000).

Note that for each simulation, there is a unique Nash equilibrium that is independent of the randomly generated starting points used in the algorithm for parties' initial placements.

To run a bootstrap procedure, we only have to set the `boot` argument specifying the number of boostrap replications as follows:

```
R> set.seed(123)
R> nash.eq.boot <- equilibrium(model = m, data = election, boot = 100)
R> nash.eq.boot

================
Nash equilibrium
================
Party positions:
   FI    UL    AN   UDC    RC
6.408 4.717 6.465 6.291 2.300

Party vote-shares:
```

```
   FI    UL    AN   UDC    RC
0.243 0.393 0.183 0.073 0.108

================
Bootstrap
================
Party positions:
                      FI    UL    AN   UDC    RC
2.5% lower bound  5.797 4.013 5.678 4.962 1.416
mean estimate     6.424 4.616 6.415 6.246 2.354
97.5% upper bound 7.040 5.094 7.220 7.386 3.332

Party vote-shares:
                      FI    UL    AN   UDC    RC
2.5% lower bound  0.207 0.338 0.154 0.049 0.081
mean estimate     0.243 0.391 0.188 0.070 0.108
97.5% upper bound 0.288 0.442 0.230 0.095 0.139

Bootstrap replications: 100
```

Similarly, to run a Monte Carlo procedure, we need to add the `MC` option specifying the number of replications.

```
R> set.seed(123)
R> nash.eq.mc <- equilibrium(model = m, data = election, MC = 100)
R> nash.eq.mc

================
Nash equilibrium
================
Party positions:
   FI    UL    AN   UDC    RC
6.408 4.717 6.465 6.291 2.300

Party vote-shares:
   FI    UL    AN   UDC    RC
0.243 0.393 0.183 0.073 0.108

================
Monte Carlo
================
Party positions:
                      FI    UL    AN   UDC    RC
2.5% lower bound  5.953 4.376 5.783 5.018 1.819
mean estimate     6.419 4.703 6.457 6.237 2.402
97.5% upper bound 6.870 5.086 7.002 6.868 3.034
```

```
Party vote-shares:
                      FI    UL    AN   UDC    RC
2.5% lower bound  0.219 0.359 0.159 0.053 0.095
mean estimate     0.242 0.391 0.185 0.074 0.108
97.5% upper bound 0.271 0.428 0.212 0.097 0.126
```

The confidence level can be specified by the argument `conf.int` (by default 0.95) in the `equilibrium` function. The complete series of simulated party positions and party shares is contained in the output object of `equilibrium` for later user manipulation.

Note that, as the typical standard error of both Monte Carlo and bootstrap experiments are proportional to $1/\sqrt{R}$ to get at least the first decimal digit right, one should use at least $R = 10000$ replications but in our runs above we set $R = 100$ just as an example of use.

4. Graphical display of the results

Through **nopp** it is also possible to display graphically the results of the analysis with the `plot` command (see Figure 1):

```
R> nash.eq <- equilibrium(model = m, data = election)
```

and passing the output of the function `equilibrium` to the `plot` function

```
R> plot(nash.eq)
```

In this case, **nopp** reports parties' equilibrium positions (upper panel of Figure 1), while in the lower panel of Figure 1 the height of each line centered on each party's optimal position is proportional to the amount of votes that each party gets in the equilibrium. Had we passed to **nopp** a list of external party positions and external vote-shares of parties, we could also display the optimal party positions/party vote-shares against the external party positions/party vote-shares.

We first create two list objects containing the quantities of interest.

```
R> external.pos <- list(FI = 7.76, UL = 3.54, RC = 1.91, AN = 8.27,
+    UDC = 5.99)
R> external.votes <- list(FI = 0.25, UL = 0.38, RC = 0.10, AN = 0.18,
+    UDC = 0.07)
```

and then pass these lists of votes and external positions as arguments to the `equilibrium` function

```
R> nash.eq <- equilibrium(model = m, data = election,
+    pos = external.pos, votes = external.votes)
```

and then plot the estimates via

```
R> plot(nash.eq)
```

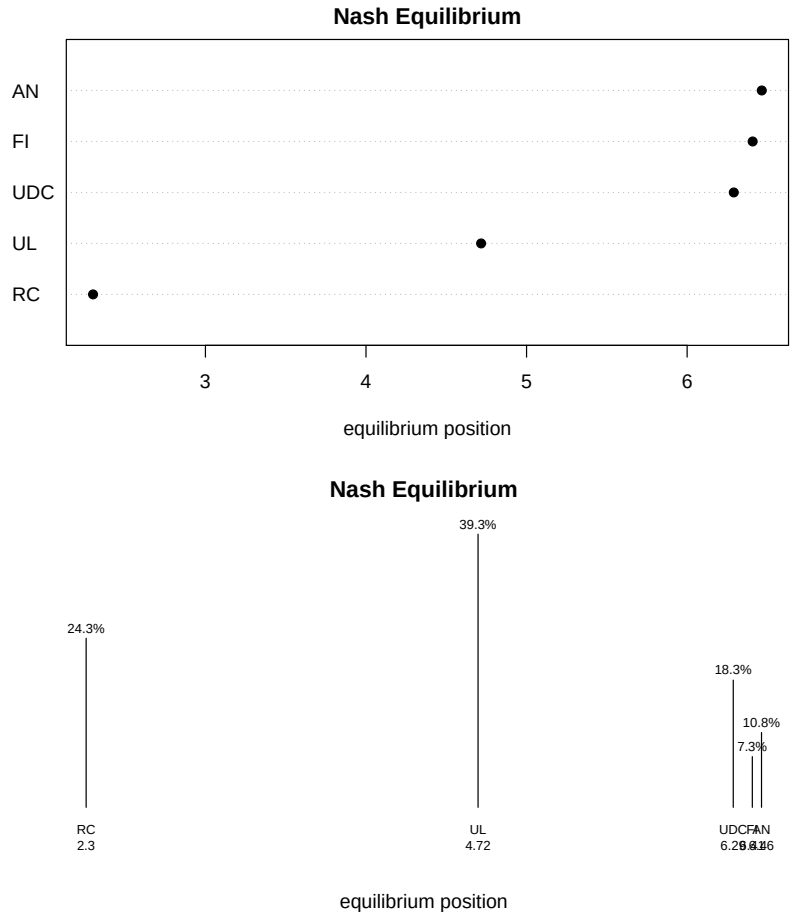

Figure 1: Plot of Nash equilibrium party positions alone (up) with the corresponding party vote-shares (down).

The results are shown in Figure 2. Alternatively, to obtain the same graphs we could type:

```
R> nash.eq <- equilibrium(model = m, data = election)
R> plot(nash.eq, pos = list(FI = 7.76, UL = 3.54, RC = 1.91, AN = 8.27,
+    UDC = 5.99))
R> plot(nash.eq, votes = list(FI = 0.25, UL = 0.38, RC = 0.10, AN = 0.18,
+    UDC = 0.07))
```

After running a bootstrap procedure, the `plot` command also displays parties' equilibrium positions with their corresponding confidence intervals, as well as a graph contrasting the equilibrium positions as they arise from the normal Nash procedure and from the bootstrap Nash procedure (see Figure 3), with

```
R> plot(nash.eq.boot)
```

The same happens after running a Monte Carlo procedure.

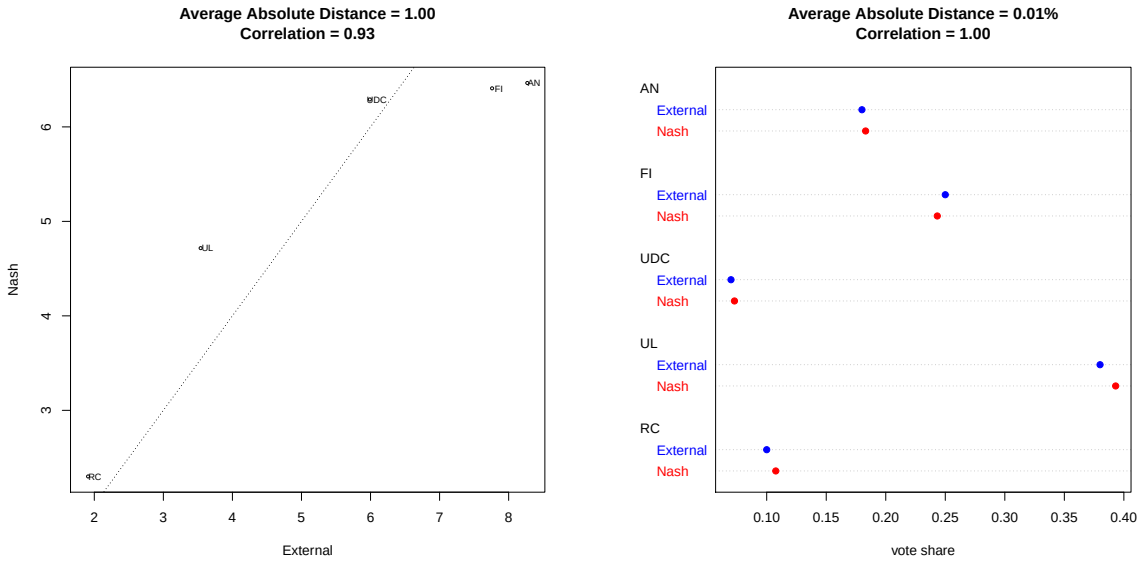

Figure 2: Comparing Nash equilibrium and external party positions (left) and external party vote-shares (right).

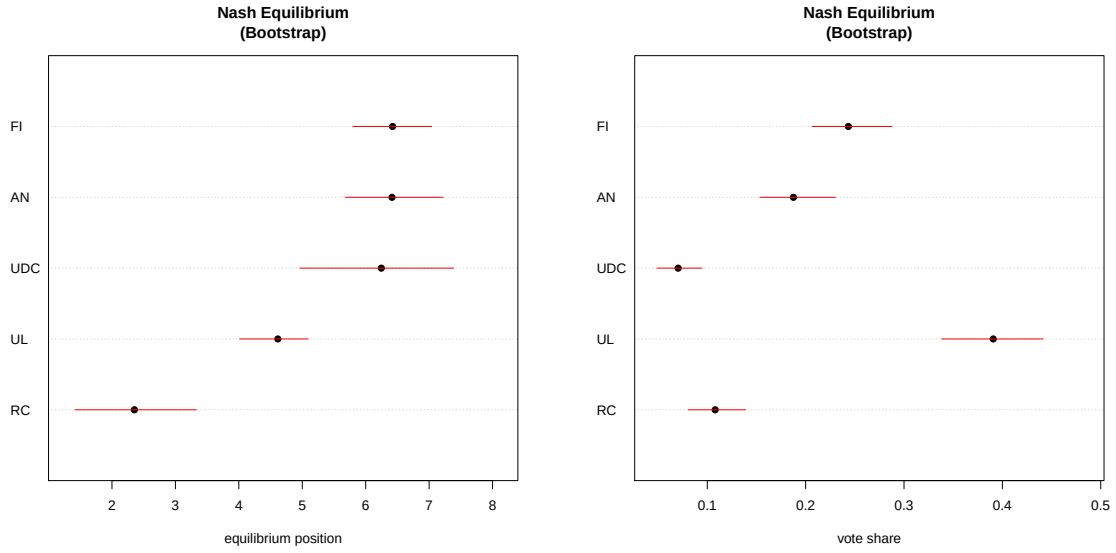

Figure 3: Plot of Nash equilibrium party positions (up) and vote-shares (down) with 95% bootstrap confidence intervals.

5. Conclusions

Comparing real data with a clear counterfactual scenario is always useful to derive (possibly new) insights on the phenomenon that we want to understand. This type of proto-theoretical exercise (Axelrod 1997) is what **nopp** allows the researcher to do. In this respect, we consider the flexibility of **nopp**'s in relation to the different motivations in each party's strategies it can accommodate (partly or fully vote seeking, party or fully coalition seeking, etc.) and the estimation procedures it can include (linear or quadratic utility functions, conditional logit or models that relax the IIA assumption, different estimation of uncertainty, etc.) as useful

tools for scholars interested in modeling party competition within a spatial framework. By estimating parties' equilibrium policy positions, it becomes possible, for example, to contrast the actual positions of the parties with a theoretical benchmark. The aim of such comparison is to understand the nature of the electoral incentives facing parties under different scenarios and to derive intuitions on real party system competition based on this understanding (see for example Curini 2015; Guarnieri 2014; Mauerer, Thurner, and Debus 2015).

Future alternative specifications for U_{ik} that may be used in place of the utility model defined in Equation 1 includes a post-electoral policy preference model (Kedar 2005), a policy discounting model (Adams *et al.* 2005), a model that mixes directional and proximity theory (Merrill III and Grofman 1999), a model in which the utility attached to voting for a party is affected by expectations of the transmission of votes into representation in government (Calvo and Hellwig 2011), and others.

References

Adams J (1999). "Policy Divergence in Multicandidate Probabilistic Spatial Voting." *Public Choice*, **100**(492), 103–122. doi:10.1023/a:1018301007077.

Adams JF, Merrill III S, Grofman B (2005). *A Unified Theory of Party Competition*. Cambridge University Press, Cambridge. doi:10.1017/CBO9780511614453.001.

Alvarez MR, Nagler J (1998). "When Politics and Models Collide: Estimating Models of Multiparty Elections." *American Journal of Political Science*, **42**, 1349–1363. doi:10.2307/2991747.

Axelrod R (1997). *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton University Press, New Jersey.

Ben-Akiva ME, Lerman SR (1985). *Discrete Choice Analysis: Theory and Application to Travel Demand*. MIT Press, Cambridge.

Benoit K, Laver M (2006). *Party Policy in Modern Democracies*. Routledge, London. doi:10.4324/9780203028179.

Budge I (1994). "A New Theory of Party Competition: Uncertainty, Ideology, and Policy Equilibria Viewed Comparatively and Temporally." *British Journal of Political Science*, **24**(4), 443–467. doi:10.1017/s0007123400006955.

Budge I, Klingemann HD, Volkens A, Bara J, Tanenbaum E (2001). *Mapping Policy Preferences*. Oxford University Press, Oxford.

Calvo E, Hellwig T (2011). "Centripetal and Centrifugal Incentives under Different Electoral Systems." *American Journal of Political Science*, **55**(1), 27–41. doi:10.1111/j.1540-5907.2010.00482.x.

Cox G (1990). "Centripetal and Centrifugal Incentives in Electoral Systems." *American Journal of Political Science*, **34**(4), 903–935. doi:10.2307/2111465.

Croissant Y (2012). *Estimation of Multinomial Logit Models in R : The* **mlogit** *Package*. R package version 0.2-2, URL https://CRAN.R-project.org/package=mlogit.

Curini L (2015). "Explaining Party Ideological Stances." *Public Choice*, **162**, 79–96. doi:10.1007/s11127-014-0199-6.

Curini L, Hino A (2105). "Missing Links in Party-System Polarization: How Institutions and Voters Matter." *Journal of Politics*, **74**(2), 460–473. doi:10.1017/s0022381611001721.

Curini L, Iacus SM (2008). "Italian Spatial Competition between 2006 and 2008: A Changing Party System?" In *XXII Congress of the Italian Political Science Society (SISP), Pavia, 5-8 September*. SISP.

Curini L, Iacus SM (2017). ***nopp**: Nash Optimal Party Positions*. R package version 1.1.0, URL https://CRAN.R-project.org/package=nopp.

Dow JK, Endersby JW (2004). "A Comparison of Conditional Logit and Multinomial Probit Models in Multiparty Elections." *Electoral Studies*, **23**, 107–122. doi:10.1016/s0261-3794(03)00040-4.

Golder SN (2006). *The Logic of Pre-Electoral Coalition Formation*. The Ohio State University Press, Columbus.

Guarnieri F (2014). "Comportamento Eleitoral e Estratégia Partidária Nas Eleçiões Presidenciais No Brasil (2002–2010)." *Opinião Pública*, **20**(2), 157–177. doi:10.1590/1807-01912014202157.

Kedar O (2005). "When Moderate Voters Prefer Extreme Parties: Policy Balancing in Parliamentary Elections." *American Political Science Review*, **99**(2), 185–99. doi:10.1017/s0003055405051592.

King G, Tomz M, Wittenberg J (2000). "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." *American Journal of Political Science*, **44**, 341–355. doi:10.2307/2669316.

Mauerer I, Thurner PW, Debus M (2015). "Under Which Conditions Do Parties Attract Voters' Reactions to Issues? Party-Varying Issue Voting in German Elections 1987–2009." *West European Politics*, pp. 1–23. doi:10.1080/01402382.2015.1026562.

Merrill III S, Adams J (2001). "Computing Nash Equilibria in Probabilistic, Multiparty Spatial Models with Nonpolicy Components." *Political Analysis*, **9**, 347–361. doi:10.1093/oxfordjournals.pan.a004874.

Merrill III S, Grofman B (1999). *A Unified Theory of Voting: Directional and Proximity Spatial Models*. Cambridge University Press, Cambridge.

Plümper T, Martin CW (2008). "Multiparty Competition: A Computational Model with Abstention and Memory." *Electoral Studies*, **27**, 424–441. doi:10.1016/j.electstud.2008.02.010.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Schofield N, Sened I (2006). *Multiparty Democracy: Parties, Elections and Legislative Politics in Multiparty Systems*. Cambridge University Press, Cambridge.

A. Preparing the data

In this appendix we briefly explain how to build a dataset in R that can be used in **nopp** starting from a format displayed by the majority of electoral survey data available online. The dataset that we will use is called `italy2006.wide` (source: http://www.cses.org/).

```
R> data("italy2006.wide", package = "nopp")
R> head(italy2006.wide)
```

```
    country id self FI DS AN DL UDC RC pID age sex education vote gov_perf
1 Italy2006  1    3 NA NA NA NA  NA NA   0   1   0         1   UL       NA
2 Italy2006  2    4 NA NA NA NA  NA NA   0   4   1         1   UL        3
3 Italy2006  3    4 NA NA NA NA  NA NA  23   6   0         0   UL        4
4 Italy2006  4    3 NA NA NA NA  NA NA   0   3   1         1   UL        3
5 Italy2006  5    4 NA NA NA NA  NA NA  23   5   0         0   UL        3
6 Italy2006  6   10  1  9  9  7   6  1   3   4   1         4   AN       NA
```

where `vote` is a variable that identifies the party voted for by the respondent in the 2006 Italian general election, `self` is the self-placement of respondent on a 0 to 10 left-right scale, `pID` is a variable that identifies the partisanship of the respondent (where 0 = stands for no party ID, 1 = FI party ID, 23 = UL party ID, 3 = AN party ID, 4 = UDC party ID, 6 = RC party ID), and the variables from `FI` to `RC` identify the placement of those parties as perceived by the respondent.

As the first step, we estimate the survey respondents' mean in each party placement used as the party's actual position.

```
R> varlist <- c("FI", "DS", "AN", "DL", "UDC", "RC")
R> for (var in varlist) {
+    assign(sprintf("mean_%s", var),
+      mean(italy2006.wide[, var], na.rm = TRUE))
+  }
```

Note that in the 2006 Italian general election the two main parties of the center-left coalition (DS and DL) presented a joint list for the Chamber of Deputies (under the name of "Ulivo – UL"). Therefore, in our analysis to estimate the placement of UL we compute the average between DS and DL.

```
R> mean_UL <- (mean_DS + mean_DL) / 2
R> mean_UL
```

```
[1] 3.540472
```

To compute the ideological distance between the respondent and each party placement according to a quadratic utility function (i.e., $-(\texttt{self}_i - s_k)^2$), we would type:

```
R> varlist <- c("FI", "UL", "AN", "UDC", "RC")
R> for (var in varlist) {
+    italy2006.wide[[sprintf("prox_%s", var)]] <-
+      - (get(sprintf("mean_%s", var)) - italy2006.wide$self)^2
+  }
```

On the other hand, to compute the same ideological distance but according to a linear utility function (i.e., $-|\texttt{self}_i - s_k|$.), we would type:

```
R> for (var in varlist) {
+    italy2006.wide[[sprintf("proxlin_%s", var)]] <-
+      - abs(get(sprintf("mean_%s", var)) - italy2006.wide$self)
+  }
R> head(italy2006.wide)

    country id self FI DS AN DL UDC RC pID age sex education vote
1 Italy2006  1    3 NA NA NA NA  NA NA   0   1   0         1   UL
2 Italy2006  2    4 NA NA NA NA  NA NA   0   4   1         1   UL
3 Italy2006  3    4 NA NA NA NA  NA NA  23   6   0         0   UL
4 Italy2006  4    3 NA NA NA NA  NA NA   0   3   1         1   UL
5 Italy2006  5    4 NA NA NA NA  NA NA  23   5   0         0   UL
6 Italy2006  6   10  1  9  9  7   6  1   3   4   1         4   AN
  gov_perf    prox_FI    prox_UL    prox_AN   prox_UDC    prox_RC
1       NA -22.725059  -0.292110 -27.825625  -8.923486  -1.177602
2        3 -14.190897  -0.211166 -18.275625  -3.949045  -4.347949
3        4 -14.190897  -0.211166 -18.275625  -3.949045  -4.347949
4        3 -22.725059  -0.292110 -27.825625  -8.923486  -1.177602
5        3 -14.190897  -0.211166 -18.275625  -3.949045  -4.347949
6       NA  -4.985928 -41.725502  -2.975625 -16.102400 -65.370031
  proxlin_FI proxlin_UL proxlin_AN proxlin_UDC proxlin_RC
1  -4.767081  -0.540472     -5.275    -2.98722  -1.085174
2  -3.767081  -0.459528     -4.275    -1.98722  -2.085174
3  -3.767081  -0.459528     -4.275    -1.98722  -2.085174
4  -4.767081  -0.540472     -5.275    -2.98722  -1.085174
5  -3.767081  -0.459528     -4.275    -1.98722  -2.085174
6  -2.232919  -6.459528     -1.725    -4.01278  -8.085174
```

Eventually, we could compute the quadratic proximity between `self` and the placement of each party as perceived by the same respondent, by typing:

```
R> italy2006.wide$UL <- (italy2006.wide$DS + italy2006.wide$DL) / 2
R> varlist <- c("FI", "UL", "AN", "UDC", "RC")
R> for (var in varlist) {
+    italy2006.wide[[sprintf("proxego_%s", var)]] <-
+      - (italy2006.wide[[var]] - italy2006.wide$self)^2
+  }
```

Finally, we could want to compute a partisanship variable for each party that takes the value of 1 if a respondent is identified with a specific party. In this particular data set, the ID that identifies each party is a numerical value, so the following procedure is needed to create the `partyID` variables

```
R> italy2006.wide$partyID_FI <- as.numeric(italy2006.wide$pID == 1)
R> italy2006.wide$partyID_UL <- as.numeric(italy2006.wide$pID == 23)
```

```
R> italy2006.wide$partyID_AN <- as.numeric(italy2006.wide$pID == 3)
R> italy2006.wide$partyID_UDC <- as.numeric(italy2006.wide$pID == 4)
R> italy2006.wide$partyID_RC <- as.numeric(italy2006.wide$pID == 6)
```

Now the dataset is ready to be used in **nopp**

```
R> colnames(italy2006.wide)
```

```
 [1] "country"     "id"          "self"        "FI"
 [5] "DS"          "AN"          "DL"          "UDC"
 [9] "RC"          "pID"         "age"         "sex"
[13] "education"   "vote"        "gov_perf"    "prox_FI"
[17] "prox_UL"     "prox_AN"     "prox_UDC"    "prox_RC"
[21] "proxlin_FI"  "proxlin_UL"  "proxlin_AN"  "proxlin_UDC"
[25] "proxlin_RC"  "UL"          "proxego_FI"  "proxego_UL"
[29] "proxego_AN"  "proxego_UDC" "proxego_RC"  "partyID_FI"
[33] "partyID_UL"  "partyID_AN"  "partyID_UDC" "partyID_RC"
```

```
R> election <- set.data(italy2006.wide, shape = "wide",
+    choice = "vote", varying = c(16:25, 27:36), sep = "_")
R> head(election)
```

```
        country id self FI DS AN DL UDC RC pID age sex education
1.AN  Italy2006  1    3 NA NA NA NA  NA NA   0   1   0         1
1.FI  Italy2006  1    3 NA NA NA NA  NA NA   0   1   0         1
1.RC  Italy2006  1    3 NA NA NA NA  NA NA   0   1   0         1
1.UDC Italy2006  1    3 NA NA NA NA  NA NA   0   1   0         1
1.UL  Italy2006  1    3 NA NA NA NA  NA NA   0   1   0         1
2.AN  Italy2006  2    4 NA NA NA NA  NA NA   0   4   1         1
       vote gov_perf UL alt       prox   proxlin proxego partyID chid
1.AN  FALSE       NA NA  AN -27.825625 -5.275000      NA       0    1
1.FI  FALSE       NA NA  FI -22.725059 -4.767081      NA       0    1
1.RC  FALSE       NA NA  RC  -1.177602 -1.085174      NA       0    1
1.UDC FALSE       NA NA UDC  -8.923486 -2.987220      NA       0    1
1.UL   TRUE       NA NA  UL  -0.292110 -0.540472      NA       0    1
2.AN  FALSE        3 NA  AN -18.275625 -4.275000      NA       0    2
```

Note that in the `election` object there are also several missing values. This does not constitute a problem for the estimation of the empirical model. However, the missing values, where present, should be dropped from the dataset before calling the equilibrium routine, in the following way:

```
R> m <- mlogit(vote ~ prox + partyID  | gov_perf + sex, election,
+    reflevel = "UL")
R> election2 <- election
R> idx <- which(is.na(election2$gov_perf) | is.na(election2$prox) |
+    is.na(election2$partyID))
```

```
R> election2 <- election2[-idx, ]
R> nash.eq <- equilibrium(model = m, data = election2)
R> nash.eq

================
Nash equilibrium
================
Party positions:
   FI    UL    AN   UDC    RC
6.420 4.763 6.378 6.456 2.295

Party vote-shares:
   FI    UL    AN   UDC    RC
0.243 0.393 0.183 0.073 0.108
```

Affiliation:

Luigi Curini
Department of Social and Political Sciences
University of Milan
Via Passione 13, I-20123 Milan, Italy
E-mail: luigi.curini@unimi.it

Stefano M. Iacus
Department of Economics, Management and Quantitative Methods
University of Milan
Via Conservatorio 7, I-20123 Milan, Italy
E-mail: stefano.iacus@unimi.it

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
November 2017, Volume 81, Issue 11 *Submitted:* 2015-09-27
doi:10.18637/jss.v081.i11 *Accepted:* 2016-10-28