Crisp Set QCA in R

Research Design for Causal Inference

Marco Steenbergen

2025-02-24

Introduction

This document shows the R syntax required to perform crisp set qualitative comparative analysis (QCA). To perform the analyses, you should install the `QCA` package (Thiem and Duşa 2013). The use of the package is illustrated with data on female parliamentary representation (Krook 2010).

Steps for Finding Minimally Sufficient Conditions

An analysis of the sufficient conditions in csQCA proceeds in four steps: (a) reading in the data; (b) calibration; (c) construction of the truth table; and (d) application of the Quine-McCluskey algorithm.

Data

The first six observations in the raw data for Krook (2010) are as follows:

```r
load("krook.Rda")
head(krook)
```

```
      country elsyst quotas    welfstate   movement leftstr wompct
1      Sweden     PR    Yes      Soc Dem       None       5   47.3
2     Finland     PR     No      Soc Dem       None       5   42.0
3      Norway     PR    Yes      Soc Dem Autonomous      14   37.9
4     Denmark     PR     No      Soc Dem Autonomous       7   36.9
5 Netherlands     PR    Yes Conservative Autonomous       7   36.7
6       Spain     PR    Yes Conservative Autonomous       0   36.0
```

Calibration

Calibration is the act of rendering the features and outcome as binary variables. We start by creating an empty data frame, which turns the first column of *krook* into a set of row names and adds new capitalized column names:[1]

```
krook_cal <- data.frame(matrix(NA, nrow = 22, ncol = 6, byrow = TRUE))
rownames(krook_cal) <- krook[,1]
colnames(krook_cal) <- c("ES", "QU", "WS", "WM", "LP", "WNP")
head(krook_cal)
```

```
            ES QU WS WM LP WNP
Sweden      NA NA NA NA NA  NA
Finland     NA NA NA NA NA  NA
Norway      NA NA NA NA NA  NA
Denmark     NA NA NA NA NA  NA
Netherlands NA NA NA NA NA  NA
Spain       NA NA NA NA NA  NA
```

Line 1 of the code block initializes the data frame, which has as many rows and columns as the original data. Line 2 assigns the country names as row names. Line 3 assigns column names. Finally, line 4 shows the first six lines of the new data frame.

We now fill this data frame with 0s and 1s according to the following rules:

1. ES=1 if the electoral system is PR; 0 otherwise.
2. QU=1 if quotas is yes; 0 otherwise.
3. WS=1 if welfare state is SocDem; 0 otherwise.
4. WM=1 if women's movement is autonomous; 0 otherwise.
5. LP=1 if the left party seat share > 7%; 0 otherwise.
6. WNP=1 if female representation > 30%; 0 otherwise.

Thus,

```
library(QCA)
krook_cal$ES <- ifelse(krook$elsyst=="PR", 1, 0)
krook_cal$QU <- ifelse(krook$quotas=="Yes", 1, 0)
krook_cal$WS <- ifelse(krook$welfstate=="Soc Dem", 1, 0)
krook_cal$WM <- ifelse(krook$movement=="Autonomous", 1, 0)
krook_cal$LP <- calibrate(krook$leftstr, type = "crisp", thresholds = 7)
krook_cal$WNP <- calibrate(krook$wompct, type = "crisp", thresholds = 30)
head(krook_cal)
```

[1] Capitalization is not essential but I often use it because an older QCA literature used to rely on this notation.

```
            ES QU WS WM LP WNP
Sweden       1  1  1  0  0   1
Finland      1  0  1  0  0   1
Norway       1  1  1  1  1   1
Denmark      1  0  1  1  1   1
Netherlands  1  1  0  1  1   1
Spain        1  1  0  1  0   1
```

Line 1 of the code block loads the `QCA` package, which contains the `calibrate` function. Lines 2-5 are simple `ifelse` statements that map categories onto 0-1 variables. For instance, line 2 takes the countries with a PR electoral system and scores them as 1, whereas countries with majority or mixed electoral systems receive a score of 0. For the continuous features, we use the `calibrate` function. In line 6, we set a seat share threshold of 7% to distinguish between weak (0) and strong (1) left parties. In line 7, we set a female representation threshold of 30% to distinguish between countries with weak (0) and strong (1) female legislative representation. Line 8 again shows the first six observations in the data frame.

Truth Table

The truth table is generated by applying the `truthTable` function to the calibrated data. Two key decisions have to be made. First, what outcome (1 or 0) should be analyzed? By setting `outcome = "WNP"`, I focus on countries with strong female representation.[2] Second, what should be done with the logical remainders? By setting `ncut = 1` we declare the minimum number of cases to be 1 as the threshold for logical remainders. There must be at least one instance of a constellation not to be called a logical remainder. The remaining parts of the code affect what is displayed. Specifically, `show.cases = TRUE` causes the country names to be shown. Further, `sorty.by` influences how the truth table entries are sorted, in our cases first by the inclusion value and then by the number of cases that are found in a particular constellation.

```
krook_truth <- truthTable(krook_cal,
                           outcome = "WNP",
                           n.cut = 0,
                           show.cases = TRUE,
                           sort.by = c("incl", "n"))
krook_truth
```

```
  OUT: output value
    n: number of cases in configuration
```

[2] To focus on countries with weak female representation, the outcome should have been specified as `~WNP`.

```
 incl: sufficiency inclusion score
  PRI: proportional reduction in inconsistency

    ES QU WS WM LP   OUT    n  incl  PRI
28  1  1  0  1  1     1     2  1.000 1.000
32  1  1  1  1  1     1     2  1.000 1.000
 4  0  0  0  1  1     1     1  1.000 1.000
12  0  1  0  1  1     1     1  1.000 1.000
21  1  0  1  0  0     1     1  1.000 1.000
24  1  0  1  1  1     1     1  1.000 1.000
26  1  1  0  0  1     1     1  1.000 1.000
27  1  1  0  1  0     1     1  1.000 1.000
29  1  1  1  0  0     1     1  1.000 1.000
11  0  1  0  1  0     0     4  0.000 0.000
25  1  1  0  0  0     0     3  0.000 0.000
 3  0  0  0  1  0     0     2  0.000 0.000
 9  0  1  0  0  0     0     1  0.000 0.000
18  1  0  0  0  1     0     1  0.000 0.000
    cases
28  Netherlands,Belgium
32  Norway,Iceland
 4  New Zealand
12  Germany
21  Finland
24  Denmark
26  Austria
27  Spain
29  Sweden
11  Australia,United Kingdom,France,Ireland
25  Switzerland,Portugal,Greece
 3  Canada,USA
 9  Italy
18  Luxembourg
```

Here is how to interpret the truth table.

1. Column 1 indexes a particular constellation. This is done in lexicographic order. For instance, when all predictors are zero, then the index is 1. When the first four predictors are zero, but the last one is one, then the index is 2, etc.
2. Columns 2-6 indicate whether a predictor applies (1) or not (0).
3. Column 7 indicates the outcome, 1 for strong female representation and 0 for weak representation

4. Column 8 indicates how many cases are covered by the constellation. For instance, constellation 28 applies to two countries.
5. Column 9 gives the *sufficiency* inclusion score (incl). When we say a condition is "sufficient" for an outcome, all cases with that condition should also exhibit the outcome. In our case, the sufficiency inclusion score is 1 for all constellations where female representation is strong. That means that the outcome is present when the constellation is present. This is what we would expect with crisp set QCA. Notice that the cases where female representation is low receive a sufficiency inclusion score of 0.[3] Note that sufficiency inclusion is by definition 1 in crisp set QCA, at least once contradictions have been eliminated.
6. Column 10 gives the proportional reduction in inconsistency (PRI). This is not relevant for crisp set QCA and shall be discussed when we show how the `QCA` package can be used to perform fuzzy set QCA.
7. Column 11 shows the cases to which a constellation applies. For instance, the constellation index as 28 covers Belgium and the Netherlands.

Minimization via Quine-McCluskey

The truth table list all conjunctions of sufficient conditions for strong female representation. There are nine such conjunctions. Using the algorithm developed by Quine (1952) and McCluskey (1956), we can now see if this number can be reduced.

In the `QCA` package, the algorithm is implemented through the `minimize` function, which is applied to the truth table. Line 1 of the syntax invokes the function. Line 2 selects the minimization method, in this case `qm` for Quine-McCluskey. There are other algorithms that could be used; a detailed discussion can be found in Thiem and Duşa (2013). Line 3 calls for a detailed output. Finally, line 4 of the `minimize` function calls for the cases to be shown for a particular solution.

```
krook.qm <- minimize(krook_truth,
                     method = "qm",
                     details = TRUE,
                     show.cases = TRUE)
krook.qm
```

```
M1: ~ES*~WS*WM*LP + ES*QU*~WS*WM + ES*QU*~WS*LP + ES*WS*~WM*~LP + ES*WS*WM*LP
    <-> WNP

                  inclS   PRI   covS   covU   cases
```

[3] There also is a necessity inclusion score, which we discuss later in this document.

```
--------------------------------------------------------------------------
1  ~ES*~WS*WM*LP  1.000  1.000  0.182  0.182  New Zealand; Germany
2   ES*QU*~WS*WM  1.000  1.000  0.273  0.091  Spain; Netherlands,Belgium
3   ES*QU*~WS*LP  1.000  1.000  0.273  0.091  Austria; Netherlands,Belgium
4  ES*WS*~WM*~LP  1.000  1.000  0.182  0.182  Finland; Sweden
5    ES*WS*WM*LP  1.000  1.000  0.273  0.273  Denmark; Norway,Iceland
--------------------------------------------------------------------------
              M1  1.000  1.000  1.000
```

We observe there is equifinality among five different constellations of prime implicants. To understand the solution, the following legend may help.

1. * is identical to AND or ∧. It means that we have a conjunction of factors.
2. ~ means negation. In Germany and New Zealand, for example, high female representation is obtained in the presence of strong left parties and independent women movements, but in the *absence* of proportional representation and a social-democratic welfare state.
3. `inclS` is another sufficiency inclusion score, in this case for the minimal conjunctions of sufficient conditions.
4. `covS` is the raw coverage of a constellation. Here, it fluctuates between 0.182 and 0.273.
5. `covU` is the unique coverage of a constellation. In our case, the highest unique coverage is for `ES*WS*WM*LP`. This is a reason to favor that solution over others.
6. `PRI` again can be ignored in crisp set QCA.

What about Necessity?

So far, I have shown how to use the `QCA` package to explore sufficiency. We can also use the package to explore necessity. A conjunction of conditions becomes necessary for the outcome when that outcome does not occur in the absence of those conditions. One way to explore necessity is by computing the *necessity* inclusion score. Where the sufficiency inclusion score takes the conjunctions of conditions and outcomes present in the truth table and divides by the instances displaying the conditions, the necessity inclusion score divides by the instances displaying the outcome. The idea is simple. If a constellation of factors is necessary, then all instances of the outcome should occur in conjunction with the factors.

One way to compute the necessity inclusion score is to use the homemade `nec_inc` function below. This function takes the truth table as an input and selects all those cases where the outcome is 1. It then computes and returns the necessity inclusion as the number of cases with a particular configuration and the outcome divided by the total number of cases with the outcome. If necessity holds then all instances of the outcome should also have the constellation, so that necessity inclusion is 1 for one of the constellations. That is not the case here (see the column labeled "necessity").

```r
nec_inc <- function(truth) {
  temp <- subset(truth$tt, truth$tt$OUT==1)
  out <- nrow(temp)
  temp$necessity <- temp$n/out
  return(temp)
}
nec_inc(krook_truth)
```

```
   ES QU WS WM LP OUT n incl PRI                cases necessity
4   0  0  0  1  1   1 1    1   1          New Zealand 0.1111111
12  0  1  0  1  1   1 1    1   1              Germany 0.1111111
21  1  0  1  0  0   1 1    1   1              Finland 0.1111111
24  1  0  1  1  1   1 1    1   1              Denmark 0.1111111
26  1  1  0  0  1   1 1    1   1              Austria 0.1111111
27  1  1  0  1  0   1 1    1   1                Spain 0.1111111
28  1  1  0  1  1   1 2    1   1 Netherlands,Belgium 0.2222222
29  1  1  1  0  0   1 1    1   1               Sweden 0.1111111
32  1  1  1  1  1   1 2    1   1       Norway,Iceland 0.2222222
```


Krook, Mona Lena. 2010. "Women's Representation in Parliament: A Qualitative Comparative Analysis." *Political Studies* 58 (5): 886–908. https://doi.org/10.1111/j.1467-9248.2010.00833.x.

McCluskey, E. J. 1956. "Minimization of Boolean Functions." *Bell System Technical Journal* 35 (6): 1417–44. https://doi.org/10.1002/j.1538-7305.1956.tb03835.x.

Quine, W. V. 1952. "The Problem of Simplifying Truth Functions." *The American Mathematical Monthly* 59 (8): 521–31. https://doi.org/10.1080/00029890.1952.11988183.

Thiem, Alrik, and Adrian Duşa. 2013. *Qualitative Comparative Analysis with R: A User's Guide*. Springer Briefs in Political Science. New York, NY: Springer.