
資 料

質的比較分析 (QCA) のソフトの使用方法—fs/QCAと R の QCA・SetMethods パッケージ(1)

森 大 輔

本稿では、質的比較分析⁽¹⁾ (qualitative comparative analysis: QCA) について、ソフトウェアを使用した分析方法を解説する。QCAは、因果関係の分析などのために使われることが多いが、回帰分析などの統計的手法と違い、集合論や論理学等を利用した手法である。QCAでは、原因条件の集合と結果の集合との間の部分集合関係を基にして、因果関係を判断する。統計的手法に比べて個々の事例を重視した方法であり、事例研究を行うには多いが統計学を用いるには少ない事例の数であっても、うまく適用できる利点もある⁽²⁾。

QCAは近年データ分析の一手法として認知されつつあるが、その1つの要因は分析のためのソフトウェアが充実してきたことである。しかし、ソフトウェアの使い方に関する邦語の解説は少ない⁽³⁾。そのため、本稿では、QCAの主なソフトウェアの使い方を、具体例に即して解説する。

QCAの分析を行うことができるソフトウェアは、現在では複数存在する⁽⁴⁾が、本稿ではその中でも、fs/QCA⁽⁵⁾と、R⁽⁶⁾のQCAパッケージ⁽⁷⁾ (およびその発展であるSetMethodsパッケージ⁽⁸⁾) の2つを主に扱う。

解説は2回にわたって行う。(1)である本稿では通常の集合(クリスプ集合)による基本的な分析の方法を扱う。そして、次回(2)ではファジィ集合を用いた方法(ファジィ集合QCA)やその他の発展的事項を扱うことにする。

1. QCA専用ソフトウェア fs/QCA

fs/QCAは、QCAの提唱者であるチャールズ・レイガン(Charles C. Ragin)を

資 料

中心にして開発されたソフトウェアである。その歴史は古く、fs/QCAの前身のソフトウェアは、Windowsが普及する前から開発されていた⁽⁹⁾。fs/QCAでは、レイガンがこれまで確立してきたQCAの手法が一通り組み込まれており、現在、QCAの分析において、最も基本的なソフトウェアであるといつてよい。

fs/QCAの最新版(2017年5月31日現在)であるバージョン3.0は、<<http://www.socsci.uci.edu/~cragin/fsQCA/>>からダウンロードできる。これまでfs/QCAをダウンロードするときにアクセスしていたレイガンのアリゾナ大学のウェブサイト<<http://www.u.arizona.edu/~cragin/fsQCA/software.shtml>>では依然として古いバージョン2.0が提供されているが、基本的には新しいバージョン3.0を使用することを推奨する。

バージョン3.0では、プログラムの安定性が増してフリーズしにくくなった。また、インストールが不要になり、ダウンロードしたフォルダ内のfsqca.exeをダブルクリックするだけでプログラムが立ち上がるようになった。さらに、バージョン3.0からは、WindowsだけでなくMacに対応するようになっている⁽¹⁰⁾。以上のような点で改善がなされているが、ソフトウェアの内容については大きな変更はないため、引き続きバージョン2.0に基づいたマニュアル(森沢2010a)が使用可能である。

1.1. fs/QCAの準備

1.1.1. fs/QCAのダウンロードと起動

- 1) <http://www.socsci.uci.edu/~cragin/fsQCA/software.shtml>にアクセスする。
- 2) 自分のPCのOSがWindowsの場合はfsQCA 3.0 Windowsの隣のclick hereという部分をクリックする。Macの場合はfsQCA 3.0 Macの隣のclick hereという部分をクリックする。以下はWindows版に基づく説明であるがMac版も基本的には同様である。
- 3) ダウンロードしたfsqca30.zipを解凍し、中に入っているfsqca.exeをダブルクリックすると起動する。

1.1.2. ファイルの読み込み

1) データのファイルを用意する。fs/QCAではdat、txt、csvなどのファイルが主に読み込める。ここでは、csvファイル(カンマ区切り)を使用する。ファイル中には、「原因条件」として使用する集合の列⁽¹¹⁾(各値は0か1)と、「結果」として使用する集合の列(各値は0か1)が最低限含まれている。その他に、事例の番号または名前を記した列が含まれていると、後で分析がしやすい⁽¹²⁾。

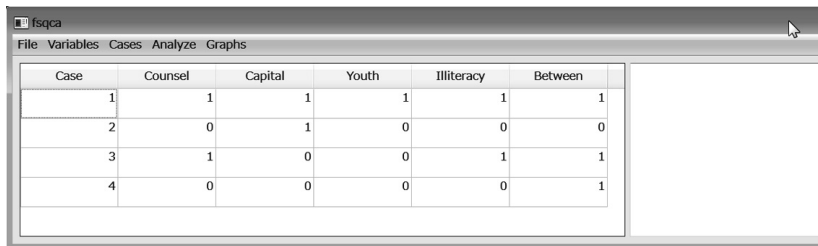
今の例では表1のようなデータのファイルBetts.csvを使用する。これは、森(2016:308)の表4の一部である。これは、1930年代から1960年代前半までの米国連邦最高裁判所の「弁護人の援助を受ける権利」(弁護人依頼権、Right to Counsel)に関する裁判例のデータである。この当時、裁判所は、被告人が抱えている個別の状況を見て弁護人を付すか否かを判断するとしていた。その個別の状況について、具体的にどのような状況であれば弁護人を付すべきと判断されるかを、質的比較分析で分析することを考える。

表1：米国の「弁護人の援助を受ける権利」に関する裁判例のデータの一部(森2016より)

Case	Counsel	Capital	Youth	Illiteracy	Between
1	1	1	1	1	1
2	0	1	0	0	0
3	1	0	0	1	1
4	0	0	0	0	1

注：Case：事例番号、Counsel：弁護人を付すべきと裁判所が判断した、Capital：死刑事件である、Youth：被告人が若年である、Illiteracy：被告人に文字が読めない等の事情がある、Between：罪状認否手続から公判開始までの間弁護人が付いていなかった

2) メニューの中のFileをクリックする。



資 料

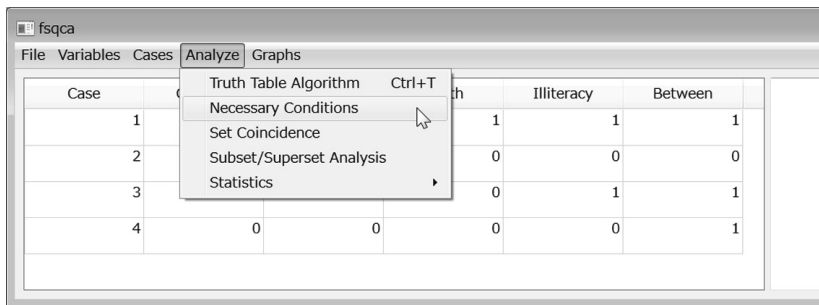
- 3) プルダウンメニューから Open を選択する。
- 4) ファイル選択画面となるので、csv 形式のファイルを選択する。今の例では、Betts.csv を選択する。

1.2. QCAの分析手順（基本）

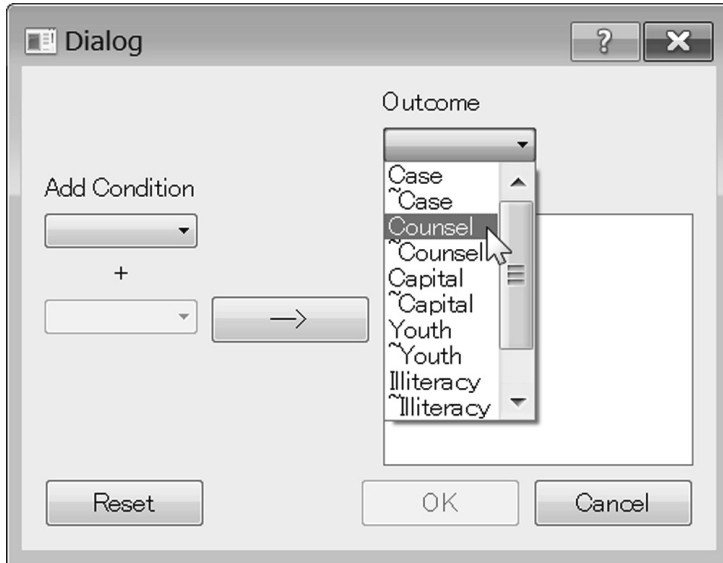
QCA の分析手順では、まず必要条件の分析を行い、次に十分条件の分析を行う (Schneider & Wagemann 2012:278)。

1.2.1. 必要条件の分析

- 1) メニューの Analyze をクリックする。
- 2) 出てくるメニューから Necessary Conditions を選択する。



- 3) Dialog というタイトルのウィンドウが現れる。Outcome の下のプルダウンをクリックする。すると、データに存在する集合 (変数)⁽¹³⁾の一覧が表示されるので、「結果」として使用する集合の名前を選ぶ。今の例の場合は Counsel を選ぶ。



- 4) Add Condition の下のプルダウンをクリックする。すると 3) と同様に、集合の一覧が表示される。~が付いているものは否定を表す。例えば Capital が死刑事件であることの集合なら、~Capital は死刑事件でないことの集合である。集合の一覧の中から、「原因条件」として使用する集合の名前を選び、その横の矢印 (→) のマークをクリックする。すると、右の Conditions の欄に、選んだ条件の集合が表示される。原因条件として使用する集合とその否定すべてについて、これを行う。今の例の場合、Capital、~Capital、Youth、~Youth、Illiteracy、~Illiteracy、Between、~Between を選ぶ⁽¹⁴⁾。
- 5) OKを押す。
- 6) 「原因条件」として使用する変数とその否定すべてについて、必要条件の整合度と被覆度 (カバー度) が表示される。今の例の場合、次のように表示される。

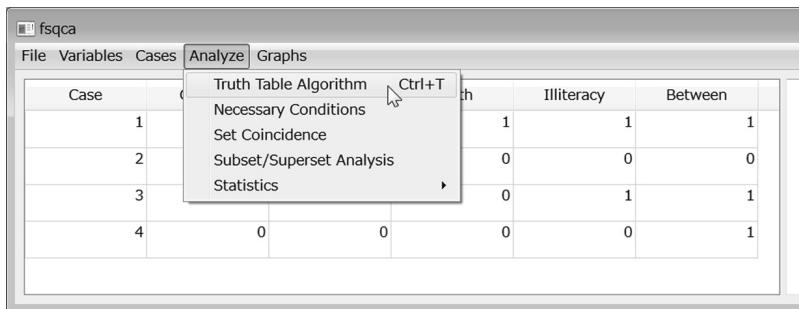
	Consistency	Coverage
Capital	0.500000	0.500000
~Capital	0.500000	0.500000
Youth	0.500000	1.000000
~Youth	0.500000	0.333333
Illiteracy	1.000000	1.000000
~Illiteracy	0.000000	0.000000
Between	1.000000	0.666667
~Between	0.000000	0.000000

7) まず Consistency の列の数字から確認する。Consistency は必要条件の整合度で、これは大まかには、その条件が必要条件である度合いを表す。1 に近いほど、必要条件である度合いが高い⁽¹⁵⁾。今の例では、Illiteracy と Between が、整合度が 1 で必要条件と判断される⁽¹⁶⁾。

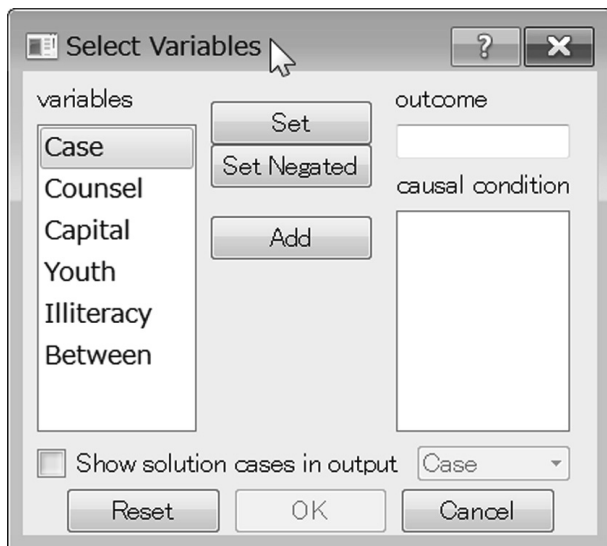
8) 次に 7) で必要条件と判断したものについて、Coverage の列の数字を確認する。Coverage は必要条件の被覆度（カバー度）で、大まかには、その条件が取るに足らない必要条件である度合いを表す⁽¹⁷⁾。0 に近いほど、取るに足らない必要条件である度合いが高い。今の例では、Illiteracy も Between も、0 に近い被覆度ではないので、取るに足らない必要条件ではないと言える。

1.2.2. 十分条件の分析

- 1) メニューの Analyze をクリックする。
- 2) 出てくるメニューから Truth Table Algorithm を選択する。



- 3) Select Variables ウィンドウが現れる。左の variables の欄から、「結果」として使用する集合を選び、Set というボタンを押す。すると、右の outcome の欄に、その集合が移る。今の例では、Counsel を選択して右の欄に移す。



- 4) 左の variables の欄から、「原因条件」として使用する集合を選び、Add というボタンを押す。すると、右の causal condition の欄に、その集合が移る。今の例では、Capital、Illiteracy、Youth、Between を選択して右の欄に移す。
- 5) Show solution cases in output のボックスにチェックを入れて、その横のプルダウンから、事例の番号や名前を記している変数を選択する。今の例では、Case に事例の番号を記しているのをこれを選択する。
- 6) OK を押す。
- 7) Edit Truth Table ウィンドウが現れる。このウィンドウには真理表 (truth table) が表示されている。このウィンドウのメニューの File をクリックして Save as CSV file を選択するとこの真理表が csv ファイルとして保存できる⁽¹⁸⁾。

資 料

Capital	Youth	Illiteracy	Between	number	Counsel	cases	raw consist.	PRI consist.	SYM consist.
1	0	0	0	1 (25%)		cases	0	0	0
0	0	0	1	1 (50%)		cases	0	0	0
0	0	1	1	1 (75%)		cases	1	1	1
1	1	1	1	1 (100%)		cases	1	1	1
0	0	0	0	0 (100%)		cases			
0	1	0	0	0 (100%)		cases			
1	1	0	0	0 (100%)		cases			
0	0	1	0	0 (100%)		cases			
1	0	1	0	0 (100%)		cases			
0	1	1	0	0 (100%)		cases			
1	1	1	0	0 (100%)		cases			
1	0	0	1	0 (100%)		cases			
0	1	0	1	0 (100%)		cases			
1	1	0	1	0 (100%)		cases			
1	0	0	1	0 (100%)		cases			
0	1	0	1	0 (100%)		cases			
1	1	0	1	0 (100%)		cases			
1	0	1	1	0 (100%)		cases			
0	1	1	1	0 (100%)		cases			

8) cases という列の cases ボタンを押すと、真理表の各行に属する事例の番号や名前が、5) で指定した変数に基づき表示される。右から三番目の raw consist. は raw consistency の略で、各行の整合度を表している⁽¹⁹⁾。raw consist. の値が 1 であればその行に属するすべての事例の結果が 1 であることを表し、raw consist. の値が 0 であればその行に属するすべての事例の結果が 0 であることを表す。

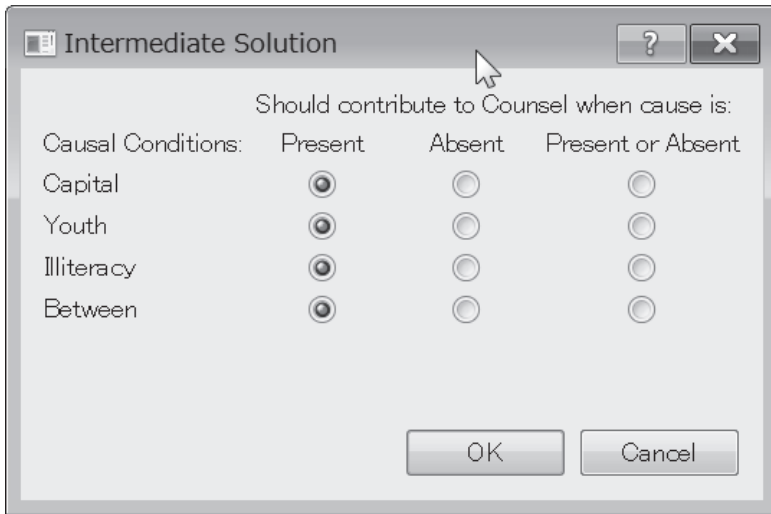
9) 結果の集合の列は、最初の段階では空欄になっている。ここに、右から三番目の raw consist. の列を見ながら、次のように数字を入力する。① raw consist. の値が 1 の行はその行に属する事例の結果が 1 なので、結果の欄に 1 を入力する。raw consist. の値が 0 の行はその行に属する事例の結果が 0 なので、結果の欄に 0 を入力する。② raw consist. が空欄の時は、その行に当てはまる事例はそもそも存在しない。このような行を論理的残余 (logical remainder) の行と呼ぶ。論理的残余の行は、結果の欄に何も入力しない。③ raw consist. の値が 0 と 1 の間の小数になることもあるが、その処理の仕方については 1.3 で説明する。今の例の場合、1 行目と 2 行目は Counsel に 0 を入力し、3 行目と 4 行目は Counsel に 1 を入力し、5 行目以下は Counsel を空欄のままにする。

質的比較分析 (QCA) のソフトの使用方法

- 10) 結果が空欄である行のうち一番上の行の、結果の欄をクリックして選択状態にする。今の例では、5行目の Counsel の欄を選択状態にする。
- 11) メニューの Edit をクリックし、出てくるメニューの中から Delete current row to last を選択する⁽²⁰⁾。

	Between	number	Counsel	cases	raw consist.
	0	0	1 (25%)	0	0
	0	1	1 (50%)	0	0
	0	1	1 (75%)	1	1
	1	1	1 (100%)	1	1
	0	0	0 (100%)		
	0	1	0 (100%)		
	1	1	0 (100%)		
	0	0	0 (100%)		

- 12) 真理表の右下にある Standard Analyses というボタンをクリックする。このときに主項 (Prime Implicant) の選択という画面が現れることがあるが、その場合の対処については1.3で説明する。今の例では、主項の選択の画面は現れない。
- 13) Intermediate solution ウィンドウが現れる。このウィンドウでは、各原因条件の存在が結果の発生に寄与すると予想されるのか (その場合は Present を選択)、それとも各原因条件の欠如が結果の発生に寄与すると予想されるのか (その場合は Absent を選択)、特にそうした予想はできないか (その場合は Present or Absent を選択)、という方向性予想 (directional expectation) を指定する。この方向性予想は、中間解 (intermediate solution) の導出に必要である⁽²¹⁾。今の例では、Capital、Illiteracy、Youth、Between のいずれも、存在が Counsel (弁護士依頼権を裁判所が認める) という結果の発生に寄与すると予想されるので、いずれも Present を選択する。



14) OKを押す。

15) 出力の欄に、上から順に複雑解 (complex solution)、簡潔解 (parsimonious solution)、中間解 (intermediate solution) という3つの解が表示される。

今の例で、複雑解の部分の表示の読み方を説明する。複雑解は、論理的残余を使用せず、事例が存在する真理表の行のみを用いて導出された解である。出力の欄のうち、COMPLEX SOLUTIONと書いてある部分より下でPARSIMONIOUS SOLUTIONと書いてある部分より上が、複雑解についての表示である。

```

--- COMPLEX SOLUTION ---

frequency cutoff: 1
consistency cutoff: 1

                raw          unique
                coverage    coverage    consistency
                -----    -----    -----
~Capital*~Youth*Illiteracy*Between    0.5          0.5          1
Capital*Youth*Illiteracy*Between      0.5          0.5          1

solution coverage: 1
solution consistency: 1

Cases with greater than 0.5 membership in term ~Capital*~Youth*Illiteracy*Between:
3 (1,1)

Cases with greater than 0.5 membership in term Capital*Youth*Illiteracy*Between: 1
(1,1)

```

上の表示のうち真ん中辺りにある、表になっている部分についてまず解説する。表の一番左の列に書いてある ~Capital*~Youth*Illiteracy*Between と Capital*Youth*Illiteracy*Between が、複雑解の項⁽²²⁾である。つまり、~Capital*~Youth*Illiteracy*Between + Capital*Youth*Illiteracy*Between → Counsel が、複雑解である。

consistency は解の項の整合度で、これは大まかには、その項が十分条件である度合い、言い換えれば複雑解の項としてふさわしい度合いを表していると言える。1 に近いほど、十分条件である度合いが高い。今の例では、~Capital*~Youth*Illiteracy*Between の整合度も Capital*Youth*Illiteracy*Between の整合度も 1 のため、どちらも複雑解の項として適切だと言える。

raw coverage は粗被覆度（粗カバー度）で、これは結果が 1 の事例の総数中の、その項でカバーされている事例数の割合を表している。また、unique coverage は固有被覆度（固有カバー度）で、これは結果が 1 の事例の総数中の、その項のみ

資 料

でカバーされている（つまり複数の項でカバーされている事件は除く）事例数の割合を表す。被覆度は割合なので、1が100%で0が0%を表す。被覆度を見ることにより、各解の項の相対的な重要性などを推し量ることができる。今の例では、2つの解の項のどちらも、粗被覆度と固有被覆度は0.5である。つまり全事例の半分ずつが、2つの解の項それぞれでカバーされており、複数の項でカバーされている事件はないことがわかる。

表の下に書いてある solution consistency は解整合度である。これは解全体の整合度で、出力された解が十分条件である度合いを表している（1に近いほど十分条件である度合いが高い）。解整合度が低い場合、解として提示するには問題がある。今の場合、解整合度は1であり問題ない。

solution coverage は解被覆度である。これは解全体の被覆度で、結果が1の事例の総数中の、解全体でカバーされている事例数の割合（1が100%で0が0%）を表している。解被覆度が低い場合、解によって説明できていない事例が多数あることになり、原因条件を増やすなどの対処を場合によっては考えた方がよい。今の例では、解被覆度は1であり問題はない。さらに、解被覆度が1の場合には、解の論理式は単なる十分条件でなく必要十分条件であると言える。なぜならこの場合、解の論理式の左辺の集合と右辺の集合が完全に一致することになるからである⁽²³⁾。今の場合も、解被覆度が1であるので、解の論理式は $\sim\text{Capital}*\sim\text{Youth}*\text{Illiteracy}*\text{Between} + \text{Capital}*\text{Youth}*\text{Illiteracy}*\text{Between} \Leftrightarrow \text{Counsel}$ と書くことができる⁽²⁴⁾（ \Leftrightarrow は必要十分条件の記号である⁽²⁵⁾）。

一番下にある Cases with greater than 0.5 membership in term という部分では、それぞれの解の項でカバーされる事例の番号や名前が、5) で指定した変数に基づき表示される。今の例では、 $\sim\text{Capital}*\sim\text{Youth}*\text{Illiteracy}*\text{Between}$: 3 (1,1)と $\text{Capital}*\text{Youth}*\text{Illiteracy}*\text{Between}$: 1 (1,1)と表示されているが、これは解の項 $\sim\text{Capital}*\sim\text{Youth}*\text{Illiteracy}*\text{Between}$ で事例3がカバーされ、 $\text{Capital}*\text{Youth}*\text{Illiteracy}*\text{Between}$ で事例1がカバーされていることを表している。

簡潔解や中間解についても、今の複雑解と同様に表示を読み取ることができる。簡潔解は、複雑解を構成するのに使った真理表の行に加え、論理的残余の行のう

質的比較分析 (QCA) のソフトの使用方法

ち論理式の形をできるかぎり単純なものにするような行をソフトウェアが探し、そのような行の結果は1になるものと仮定して（これを単純化の仮定 simplifying assumption と呼ぶ⁽²⁶⁾）、これも含めて作った解である⁽²⁷⁾。簡潔解は PARSIMONIOUS SOLUTION と書いてある部分より下に表示される。今の例では、Illiteracy \Leftrightarrow Counsel が簡潔解となる。解の項の粗被覆度、固有被覆度、整合度はすべて1、解整合度や解被覆度も1、含まれる事例は1と3となる。

中間解は、簡潔解に含めている論理的残余の行のうち、13) の方向性予想を基に、含めることがもっともらしい行（容易な反事実的仮定 easy counterfactual）のみを含め、含めることがもっともらしくない行（困難な反事実的仮定 difficult counterfactual）⁽²⁸⁾を除いてでき上がる解である⁽²⁹⁾。中間解は INTERMEDIATE SOLUTION と書いてある部分より下に表示される。なお、中間解の出力については、粗被覆度や固有被覆度や整合度の表の上に、方向性予想が表示される。今の例では、Illiteracy*Between \Leftrightarrow Counsel が中間解となる。解の項の粗被覆度、固有被覆度、整合度はすべて1、解整合度や解被覆度も1、含まれる事例は1と3となる。

1.2.3. 結果の不存在の分析

1.2.1と1.2.2では結果の存在について必要条件と十分条件を求めたが、場合によっては結果の不存在についてもそれらを求めたいことがある。その場合には以下のようにする⁽³⁰⁾。

- 1) まず1.2.1と同様に必要条件から分析する。1.2.1の1)～6)と同様に行う。唯一変更する点は3)で「結果」として選ぶ集合を Counselではなく、その否定の \sim Counselとすることである。今の例の場合、 \sim Youthと \sim Illiteracyが整合度1となり必要条件の可能性がある。
- 2) 次に十分条件の分析をする。1.2.2と同様に行うが、次の3点に注意する。① 1.2.2の3)でSetの代わりにSet Negatedを選択する。②真理表の結果の欄は \sim Counselに関するものになるが、ここは1.2.2と同様にraw consist.が0なら0、1なら1を入力する。論理的残余の扱いも1.2.2と同様である。③ Intermediate

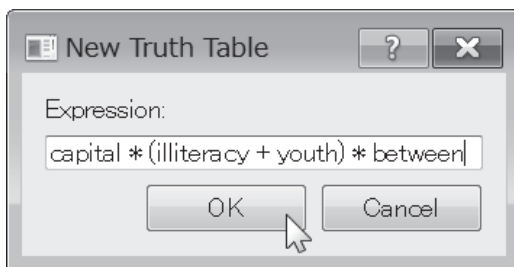
solution ウィンドウでの方向性予想は1.2.2と逆にする。すなわち今の例の場合、1.2.2では方向性予想は Counsel に対するもので全て Present にしたが、ここでは ~Counsel に対するものなので全て Absent にする。

3) Intermediate solution ウィンドウで方向性予想を指定し OK を押すと、解が出力される。今の例の場合、複雑解は $\text{Capital} * \sim \text{Youth} * \sim \text{Illiteracy} * \sim \text{Between} + \sim \text{Capital} * \sim \text{Youth} * \sim \text{Illiteracy} * \text{Between} \Leftrightarrow \sim \text{Counsel}$ 、簡潔解は $\sim \text{Illiteracy} \Leftrightarrow \sim \text{Counsel}$ 、中間解は $\sim \text{Youth} * \sim \text{Illiteracy} * \sim \text{Between} + \sim \text{Capital} * \sim \text{Youth} * \sim \text{Illiteracy} \Leftrightarrow \sim \text{Counsel}$ となり⁽³¹⁾、解整合度・解被覆度はすべて 1 である。

1.2.4. 補足：仮説との比較

QCA では仮説を立てないことが多いが、仮説を立てて実際のデータ分析の結果と照らし合わせることも提案されている (Ragin 1987:118-121; Schneider & Wagemann 2012:295-305)。今の例で $\text{Capital} * (\text{Illiteracy} + \text{Youth}) * \text{Between} \rightarrow \text{Counsel}$ という仮説を立てたとした場合の分析方法を説明する。

- 1) メニューの File をクリックする。
- 2) 出てくるメニューから New from Expression を選択する。
- 3) New Truth Table というタイトルの小さなウィンドウが現れるので、その中の空欄部分に、仮説となる十分条件の論理式の左辺部分を入力する。その際に、条件の集合の名前をすべて小文字で入力するようにする。大文字が混じっていると、結果の 0 と 1 が正しく表示されずすべて 0 になってしまう。例えば、今の例では、 $\text{capital} * (\text{illiteracy} + \text{youth}) * \text{between}$ を空欄に入力する。



質的比較分析 (QCA) のソフトの使用方法

- 4) OK を押す。
- 5) 仮説となる十分条件の論理式に従った真理表が出力される。結果は y の列に表示される。
- 6) 実際のデータから作った真理表と比べることで、複雑解と仮説に含まれる行の違いがわかる。

1.3. QCAの分析手順の詳細（矛盾のある行、主項表）

今度は、表 1 に事例をいくつか追加した表 2 のファイル（White.csv という名前とする）について分析を行うことにする。

表 2：米国の「弁護人の援助を受ける権利」に関する裁判例のデータの一部(森2016より)

Case	Counsel	Capital	Youth	Illiteracy	NoCourt	Between
1	1	1	1	1	1	1
2	0	1	0	0	1	0
3	1	0	0	1	1	1
4	0	0	0	0	0	1
5	1	1	0	0	1	1
6	1	1	0	0	1	1
7	1	0	1	1	0	1
8	1	0	0	0	1	1

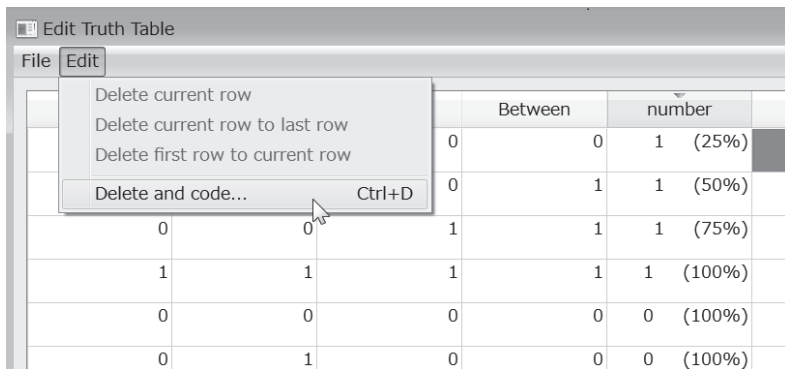
注：NoCourt：被告人に刑事裁判の経験がないか不慣れである、他の変数は表 1 と同様

- 1) 1.2.1と同様に、必要条件の分析を行う。
- 2) 1.2.2の 1)～8)と同様に、十分条件の分析を進めて、真理表を表示する。
- 3) 真理表の結果の集合の列に、右から三番目の raw consist. の列を見ながら、1.2.2の 9)と同様に数字を入力する。ただし、今の例では、raw consist. の値が 0.5という小数になっている行がある。このような行は、結果が 0 の事例もあれば 1 の事例もあるという状態になっている。このような行を矛盾のある行 (contradictory row) と呼ぶ。矛盾のある行の処理の仕方には、主に以下の 2 つがある⁽³²⁾。

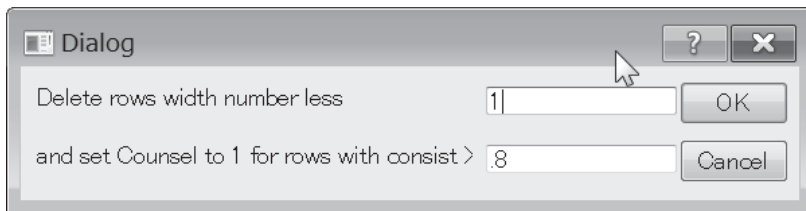
① 原因条件を追加したり、既存の原因条件の値を見直したりする⁽³³⁾。今の例では、表2のNoCourtを原因条件として追加することで、矛盾のある行は解消する。しかしこの方法で、矛盾のある行が解消しない場合や、次回説明するようなファジィ集合の場合⁽³⁴⁾には、次の②の方法も用いられる。

② 整合度 (raw consist.) の値がある閾値 (例えば0.8) 以上の場合には結果の欄に1を入力し、ある閾値より下の場合には結果の欄に0を入力する。その際は真理表のraw consist.と書かれている所をクリックするとraw consist.の値の大きき順にソートされるのでやりやすい。あるいは、次のようにソフトに行わせることもできる。

②-1) Edit Truth Table ウィンドウのメニューのEditをクリックし、出てくるメニューの中からDelete and codeを選択する。



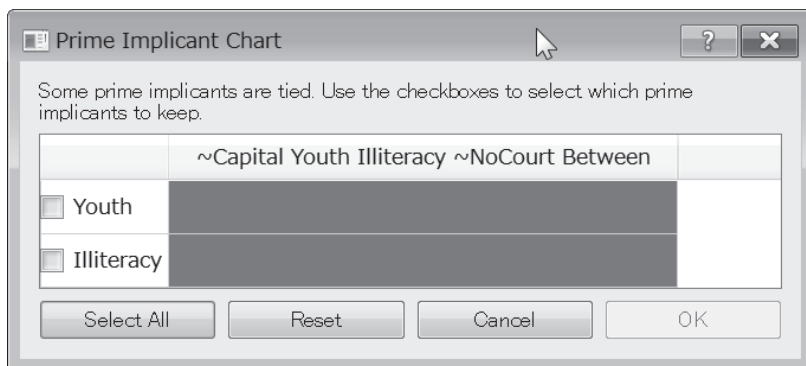
②-2) Dialogというタイトルのウィンドウが出る。and set Counsel to 1 for rows with consist>という欄で、整合度 (raw consist.) の閾値を設定する⁽³⁵⁾。デフォルトは0.8であるが、他の値を設定することもできる⁽³⁶⁾。今はこのままの値とする。



②- 3) OKを押すと、raw consist.の値が0.8以上の行には結果の欄に1、小さい行には0が自動で入り、raw consist.が空欄の行は削除される。

4) ここでは3)で①を採用し、表2のNoCourtを原因条件として追加して分析を行い直す。1.2.2の1)~12)と同様に、十分条件の分析を進める。すなわち、真理表の結果Counselの欄にraw consist.の値(矛盾のある行は解消しているので0か1か空欄のみである)を基にして値を入力し、Counselの欄が空欄の行を削除し、Standard Analysesのボタンを押す。

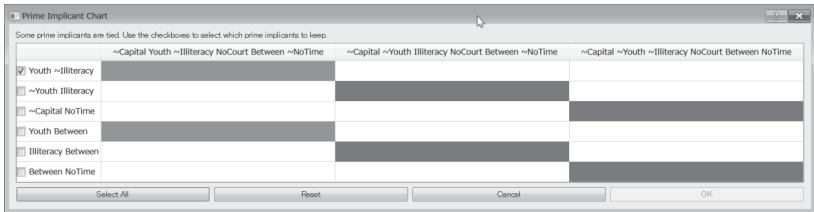
5) すると、1.2.2では出なかったPrime Implicant Chartウィンドウが現れる。これは主項表と呼ばれるものである(森沢2010a: 48-49)。解を導出する過程で、どのような形に解をまとめるかという選択に分析者の自由がある場合、この表が現れる。今の場合、YouthとIlliteracyの2つの行があり、どちらか一方(か両方)のボックスにチェックを入れると、色の付いたセルの色が変わり、右下のOKボタンが押せるようになる⁽³⁷⁾。



資 料

※主項表がもっと複雑になることもあり、例えば下の画像のようなものもある。

この場合、色が付いたセルが同じ列にあるもの（下の画像の場合、例えば2列目に色の付いたセルがある Youth~Illiteracy と Youth Between）はどちらか一方（か両方）にチェックを入れればよい。そうするとその列の色の付いたセルの色がすべて変わる。すべての列の色の付いたセルの色が変わった段階で、右下の OK ボタンが押せるようになる。



6) Intermediate solution ウィンドウが現れるので1.2.2の13)と同様に方向性予想を指定する。今の例では新たな NoCourt を含めすべて Present に指定する。

7) OK を押す。

8) 出力の詳細はここでは省略し、中間解のみ記しておく。中間解の論理式は 5) の主項表でどのような選択をしても NoCourt*Between+Youth*Illiteracy*Between ⇔ Counsel となり、解整合度、解被覆度はともに 1 である。

2. RのQCAパッケージ

Rは、統計分析などのための高機能なフリーソフトである⁽³⁸⁾。Rでは、「パッケージ」という形で自由にソフトに機能を追加できるようになっており、世界中の研究者が、様々なパッケージを提供している。QCAパッケージもそのうちの一つである⁽³⁹⁾。

Rはキーボードからのコマンド入力により操作するので、マウスのクリックでの直感的操作方法（GUI）が備わっている fs/QCA に比べると、慣れない人にとってはとっつきにくいかもしれない。しかし、何度も変数を変えたりして分析する場合や、時間をかけて同じ分析を再現する場合など、コマンド入力を残しておい

質的比較分析 (QCA) のソフトの使用方法

てその一部を書き換えた方が便利な場合は多々ある。

また、他にも利点がある。第一に、現在では、RのQCAパッケージ（およびその発展である SetMethods パッケージ）は機能が豊富になっており、fs/QCAでは行いにくい分析も行うことができるようになっている。第二に、Rは、用意されている様々な統計的手法のパッケージと連動させて分析をしたり、自分で追加のプログラムを組んだり、といった発展性に富んでいる。

2.1. R のQCAパッケージの準備

2.1.1. Rのダウンロードとインストール

- 1) OSがWindowsの場合は <https://cran.r-project.org/bin/windows/base/> にアクセスする。OSがMacの場合は <https://cran.r-project.org/bin/macosx/> にアクセスする。以下の説明はWindows版に基づくが、Mac版も基本的には同様である。
- 2) ページの一番上にある「Download R x.x.x for Windows」というリンク（x.x.xの部分はバージョン番号）をクリックすると、R-x.x.x-win.exeというインストールプログラムがダウンロードされる。
- 3) ダウンロードしたR-x.x.x-win.exeをクリックする。
- 4) 「発行元を確認できませんでした。このソフトウェアを実行しますか」という表示が出た場合は、「実行」を押す。
- 5) 「インストール中に利用する言語を選んでください」と出るので日本語を選択する。
- 6) 「R for Windows x.x.xセットアップウィザードの開始」で「次へ」、「情報」で「次へ」、「インストール先の指定」で「次へ」をクリックする。
- 7) 「コンポーネントの選択」で自分のPCが32bitの場合は32bit、64bitの場合は64bitにチェックを入れ、Message translationsにもチェックを入れたうえで、「次へ」をクリックする。
- 8) 「起動時オプション」で「次へ」、「プログラムグループの指定」で「次へ」、「追加タスクの選択」で「次へ」をクリックする。
- 9) これでRのインストールが完了する。

- 10) Rという青色のアイコンがデスクトップやスタートメニューなどに追加されるので、それをクリックすればRが起動する。

2.1.2. QCAパッケージのインストールと起動

- 1) Rのメニューの「パッケージ」をクリックする。
- 2) 出てくるメニューから「パッケージのインストール」をクリックする。
- 3) Secure CRAN mirrorsというウィンドウが出る。Japan (Tokyo) [https] を選ぶ。
- 4) Packagesというウィンドウが出るので、その中からQCAを選んでOKを押す（アルファベット順に並んでいる）。この4)まででQCAパッケージがインストールされる。以下の5)と6)は、QCAパッケージの起動の仕方である。
- 5) Rのメニューで「パッケージ」をクリックし、「パッケージの読み込み」をクリックする。
- 6) 「1つを選択してください」というウィンドウが出るので、その中からQCAを選んでOKを押す。

2.1.3. ファイルの読み込み

まず、使用するファイルの存在するフォルダに移動しておく。そのためには以下の1)～3)のようにする。

- 1) メニューの「ファイル」をクリックする。
- 2) 出てくるメニューの中から「ディレクトリの変更」を選択する⁽⁴⁰⁾。
- 3) フォルダの選択画面になるので、使用するファイル（今の例ではBetts.csv）の存在するフォルダを選択する。

以上によりフォルダ移動が完了した後に、次の4)を行う。

- 4) 下のようにRに入力する（1番左の>は、Rに最初から表示されている入力促進記号なので入力しない⁽⁴¹⁾）。

```
> Betts <- read.csv("Betts.csv", row.names="Case")
```

read.csv コマンドは、csv ファイルを読み込むためのものである。read.csv

(“Betts.csv”) で Betts.csv というファイルが読み込まれ⁽⁴²⁾、それを Betts という変数に代入している⁽⁴³⁾。row.names の部分は、各行の名前を設定するものである。今の例では、事例の番号を各行の名前にするために、Betts.csv ファイルにおいて事例の番号が記されている変数 Case を、row.names に指定している。

2.2. QCAの分析手順 (基本)

2.2.1. 必要条件の分析

必要条件の分析をするためには、次のように入力する。

```
> superSubset (Betts, outcome="Counsel", conditions=c ("Capital", "Youth",
  "Illiteracy", "Between"), incl.cut=0.9, use.tilde=TRUE)
```

superSubset コマンドは、必要条件を分析するためのものである。Betts の部分には、ファイルを代入した変数名を書く。outcome の部分には、「結果」として使用する集合の名前 (今の例では Counsel) を、ファイル中の変数から選んで入力する。conditions の後の部分には、「原因条件」として使用する集合の名前 (今の例では Capital, Youth, Illiteracy, Between) を、ファイル中の変数から選んで入力する⁽⁴⁴⁾。incl.cut の部分は、必要条件の整合度がいくつ以上のものを出力するかという閾値を指定するもので、0.9 やそれより高い値を指定することが推奨される⁽⁴⁵⁾。use.tilde を TRUE にすると、集合の否定がチルダ (~) を付けた形で表示される。use.tilde を付けない場合や FALSE にした場合は、集合は大文字で集合の否定は小文字、という表示形式になる⁽⁴⁶⁾。

出力は次のようになる。

	incl	RoN	cov.r
1 ILLITERACY	1.000	1.000	1.000
2 BETWEEN	1.000	0.500	0.667
3 ILLITERACYBETWEEN	1.000	1.000	1.000
4 ~CAPITAL+YOUTH	1.000	0.500	0.667
5 CAPITAL+~YOUTH	1.000	0.000	0.500

資 料

一番左に、集合の名前が書かれている。なお、名前はすべて大文字で書かれ、また論理積の記号*は省略される。すなわち、3行目の ILLITERACYBETWEEN は Illiteracy*Between という意味である。

inclの列は必要条件の整合度、cov.rの列は必要条件の被覆度を表している⁽⁴⁷⁾。なお、このRのQCAパッケージの場合、fs/QCAと違い、あらかじめ設定した必要条件の整合度の閾値（今の例の場合は0.9）を超えるものは、論理和や論理積を含めてすべて出力される⁽⁴⁸⁾。今の例では論理和や論理積のものを除けば、IlliteracyとBetweenが、整合度が1で必要条件と判断されることは、fs/QCAでの必要条件の分析と同様である。

2.2.2. 十分条件の分析

1) まず真理表 (truth table) を表示するには、次のように入力する。

```
> BettsTable <- truthTable (Betts, outcome="Counsel", conditions=c("Capital",
" Youth", "Illiteracy", "Between"), complete=TRUE, show.case=TRUE, sort.by=
"incl")
> BettsTable
```

truthTable コマンドは、真理表を表示するためのものである。その真理表の表示を1つ目の入力促進記号>の部分で BettsTable という変数にいったん代入し、2つ目の入力促進記号>の部分で改めてその変数の中身を表示させている。このようにいったん代入しているのは、この真理表の表示を代入した変数を後で利用する可能性があるためである⁽⁴⁹⁾。

Bettsの部分には、ファイルを代入した変数名を書く。outcomeの部分には、「結果」として使用する集合の名前（今の例では Counsel）を、ファイル中の変数から選んで入力する。conditionsの後の部分には、「原因条件」として使用する集合の名前（今の例では Capital、Youth、Illiteracy、Between）を、ファイル中の変数から選んで入力する。

complete を TRUE にすると、論理的残余も含めて真理表のすべての行が表示される。それに対して、complete を書かなかつたり FALSE にしたりすると、論理

質的比較分析 (QCA) のソフトの使用方法

的残余の行は表示されない。

show.case を TRUE にすると、真理表の各行に帰属する事例の番号が表示される。この事例の番号には、2.1.3の4)のファイルの読み込みのときに row.names で指定した変数が用いられる。

sort.by は、真理表の行の並び順をソートする機能である。例えば今の例のように sort.by="incl" とすると、整合度が大きい行から順にソートされる。

truthTable コマンドによる真理表の出力は次のようになる。一番左の列の数字は、ソートしない場合に出力される真理表の行番号を表している。今の場合、ソートをしているので、行番号が1から順に並んでいない。OUT は結果の集合（今の場合 Counsel）の列である。fs/QCA と違いこの欄に最初から数字が入っている。OUT の欄が?となっているのは論理的残余の行である。n は事例の数を表す。incl は整合度を表しており fs/QCA の raw consist. に当たる⁽⁵⁰⁾。cases では、事例の名前が、2.1.3の4)のファイルの読み込みのときに row.names で指定した変数を基に付けられている。今の例では、2.1.3の4)で Case を行の名前に指定しているので、事例の名前が Case の番号で表されている。

	CAPITAL	YOUTH	ILLITERACY	BETWEEN	OUT	n	incl	PRI	cases
4	0	0	1	1	1	1	1.000	1.000	3
16	1	1	1	1	1	1	1.000	1.000	1
2	0	0	0	1	0	1	0.000	0.000	4
9	1	0	0	0	0	1	0.000	0.000	2
1	0	0	0	0	?	0	-	-	
3	0	0	1	0	?	0	-	-	
5	0	1	0	0	?	0	-	-	
6	0	1	0	1	?	0	-	-	
7	0	1	1	0	?	0	-	-	
8	0	1	1	1	?	0	-	-	
10	1	0	0	1	?	0	-	-	
11	1	0	1	0	?	0	-	-	
12	1	0	1	1	?	0	-	-	
13	1	1	0	0	?	0	-	-	
14	1	1	0	1	?	0	-	-	
15	1	1	1	0	?	0	-	-	

資 料

2) 複雑解 (complex solution) を表示するには次のように入力する。

```

> BettsCS <- eqmcc(Betts, outcome="Counsel", conditions=c("Capital", "Youth",
  "Illiteracy", "Between"), details=TRUE, show.case=TRUE, use.tilde=TRUE)
> BettsCS

```

eqmcc コマンドは、解を導出して表示するためのものである。その解の表示を1つ目の入力促進記号>の部分でBettsCSという変数にいったん代入し、2つ目の入力促進記号>の部分で改めてその変数の中身を表示させている。このようにいったん代入しているのは、この解の表示を代入した変数を後で利用する可能性があるためである。

Bettsの部分には、ファイルを代入した変数名を書く。また、Bettsの代わりにBettsTableという、1)で真理表の表示を代入した変数名を書いても同じ結果となる。outcomeとconditionsの所には、真理表コマンドのとくと同様に「結果」と「原因条件」を書く。detailsはTRUEにすると整合度や被覆度が表示される。detailsを書かなかったりFALSEにしたりすると整合度や被覆度が表示されない。show.caseをTRUEにすると、解の各項でカバーされる事例の番号が表示される。use.tildeをTRUEにすると、集合の否定がチルダ (~) を付けた形で表示され、use.tildeを付けない場合やFALSEにした場合は、集合は大文字で集合の否定は小文字という表示形式になる。

出力は、最初に真理表が論理的残余を除いた形で表示され、その後に複雑解が、次のように表示される。

```

M1: ~CAPITAL*~YOUTH*ILLITERACY*BETWEEN+CAPITAL*YOUTH*ILLITERACY*BETWEEN
    <=>COUNSEL

```

	incl	PRI	cov.r	cov.u	cases
1 ~CAPITAL*~YOUTH*ILLITERACY*BETWEEN	1.000	1.000	0.500	0.500	3
2 CAPITAL*YOUTH*ILLITERACY*BETWEEN	1.000	1.000	0.500	0.500	1
M1	1.000	1.000	1.000		

質的比較分析 (QCA) のソフトの使用方法

一番上の M1 の所に、複雑解の論理式が書かれている。<=> (両矢印) は、必要十分条件関係を表す記号である。1.2.2の15)でも説明したように、解被覆度が 1 の場合は、解の論理式は必要十分条件関係となる。十分条件関係の場合は、この部分に表示される論理式は=> (右矢印) になる。

その下の表の incl は整合度 (解の項の行のものは解の項の整合度、M1 の行のものは解整合度)、cov.r は粗被覆度 (解の項の行のものは解の項の被覆度、M1 の行のものは解被覆度)、cov.u は固有被覆度を表す。cases にはその項でカバーされる事例の番号が表示されている。

3) 簡潔解 (parsimonious solution) を表示するには、次のようにする。

```
> BettsPS <- eqmcc (Betts, outcome="Counsel", conditions=c("Capital", "Youth",
"Illiteracy", "Between"), include="?", details=TRUE, show.case=TRUE, use.
tilde=TRUE)
> BettsPS
```

複雑解とほとんど同じであるが、唯一異なるのは include="?" という部分が加わったことである (上ではわかりやすいように太字にしてある)。これは、論理的残余を含めて解を導出するという意味である。

出力は、複雑解の時と同様に、最初に真理表が論理的残余を除いた形で表示され、その後に簡潔解が表示される。出力の読み方は複雑解と同様である。

4) 中間解 (intermediate solution) を表示するには、次のようにする。

```
> BettsIS <- eqmcc (Betts, outcome="Counsel", conditions=c("Capital", "Youth",
"Illiteracy", "Between"), include="?", dir.exp="1,1,1,1", details=TRUE, show.
case=TRUE, use.tilde=TRUE)
> BettsIS
```

今度は簡潔解に、dir.exp="1,1,1,1" という部分が加わっている (上ではわかりやすいように太字にしてある)。これは、conditions に書かれた各原因条件の、方向性予想を指定するものである。条件の存在が結果の発生に寄与すると予想する

資 料

場合は1、条件の欠如が結果の発生に寄与すると予想する場合は0、特にそうした予想はできない場合は-（ハイフン）を書く。今の場合、conditionsにこの順で書かれている4つの原因条件Capital、Youth、Illiteracy、Betweenについて、各条件の存在が結果の発生に寄与するという方向性予想を立てている。

出力は、複雑解の時と同様に、最初に真理表が論理的残余を除いた形で表示され、その後に中間解が表示される。出力の読み方は複雑解と同様である。

5) 簡潔解の単純化の仮定を確認する。簡潔解では、解を複雑解よりも単純な形にするため、論理的残余の行をいくつか使用している（これを単純化の仮定 *simplyfing assumption* と呼ぶ）。これらが具体的にどの行であるかはfs/QCAでは確認できないが、RのQCAパッケージでは確認できる。次のように入力すればよい。

```
> BettsPS$SA
```

BettsPSは、3)で簡潔解の表示を代入した変数である。その変数の後ろにSAを付けることで、簡潔解の単純化の仮定が次のように表示される。

\$M1	CAPITAL	YOUTH	ILLITERACY	BETWEEN
3	0	0	1	0
7	0	1	1	0
8	0	1	1	1
11	1	0	1	0
12	1	0	1	1
15	1	1	1	0

一番左の列の番号は、1)の真理表の一番左の列の行番号と対応している。1)の真理表と照らし合わせることで、どの行が単純化の仮定として使われているかがわかる。

6) 中間解の容易な反事実的仮定と困難な反事実的仮定を確認する。中間解では、

質的比較分析 (QCA) のソフトの使用方法

5)の単純化の仮定の中で、もっともらしい行（容易な反事実的仮定 easy counterfactual）のみ使用し、もっともらしくない行（困難な反事実的仮定 difficult counterfactual）は使用しない。これらも具体的にどの行であるかは fs/QCA では確認できないが、R の QCA パッケージでは簡単に確認できる。次のように入力すればよい。

```
> BettsIS$i.sol
```

BettsIS は、4)で中間解の表示を代入した変数である。その変数の後ろに \$i.sol を付けることで、中間解の容易な反事実的仮定や困難な反事実的仮定などが次のように表示される。

\$C1P1				
\$C1P1\$EC				
	CAPITAL	YOUTH	ILLITERACY	BETWEEN
8	0	1	1	1
12	1	0	1	1
\$C1P1\$DC				
	CAPITAL	YOUTH	ILLITERACY	BETWEEN
3	0	0	1	0
7	0	1	1	0
11	1	0	1	0
15	1	1	1	0

\$C1P1\$EC の下の表が容易な反事実的仮定で、\$C1P1\$DC の下の表が困難な反事実的仮定である。一番左の列の番号は、1)の真理表の一番左の列の行番号と対応している。

2.2.3. 結果の不存在の分析

R での結果の不存在の分析は fs/QCA 以上に簡単である。結果の存在でのコマンド内の outcome="Counsel" を outcome="~Counsel" に書き換えるだけでよい。すなわち次のようにする。

資 料

- 1) 必要条件の分析は、2.2.1の superSubset コマンド内の outcome=“Counsel”を outcome=“~Counsel”に書き換えればよい。
- 2) 十分条件の分析における真理表の作成は、2.2.2の 1) の truthTable コマンド内の outcome=“Counsel”を outcome=“~Counsel”に書き換えればよい。
- 3) 十分条件の分析における複雑解、簡潔解、中間解の導出は、2.2.2の 2)～4) の eqmcc コマンド内の outcome=“Counsel”を outcome=“~Counsel”に書き換えればよい⁽⁵¹⁾。また、中間解については方向性予想を dir. exp=“0,0,0,0”とする。これにより1.2.3と同じ解が導かれる⁽⁵²⁾。
- 4) 単純化の仮定や容易・困難な反事実的仮定の確認は2.2.2の 5) 6) と同様である。

2.2.4. 補足：仮説との比較

Rでは、仮説との比較について、fs/QCA よりもはるかに詳しく行うことができる。しかし、その機能は、QCA パッケージではなく、その発展である SetMethods パッケージに含まれている。そのため、以下の分析を行う前に、2.1.2 と同様の方法で SetMethods パッケージのインストールと起動を行っておくものとする。

- 1) 仮説の論理式の左辺を代入した変数を作る。今の例での仮説は、1.2.4と同様 Capital *(Illiteracy + Youth)* Between→Counsel だとして、それを変数 t に代入するとすると、次のように書くことになる。

```
>t<-“CAPITAL *ILLITERACY* BETWEEN + CAPITAL *YOUTH* BETWEEN”
```

仮説の論理式を書くにあたって、エラーが出ないようにするためには、次のようなことに注意する必要がある。論理式の左辺のみを書き、原因条件はすべて大文字で表し、括弧は使わず展開した形にし、否定がある場合は~ (チルダ) を使う。

- 2) 仮説と解の比較をするためには、次のように書く。

```
>theory.evaluation (theory = t, empirics = BettsCS, outcome = “Counsel”, intermed = FALSE)
```

質的比較分析 (QCA) のソフトの使用方法

theory.evaluation コマンドは、仮説（理論 theory）と解（経験的データ empirics）の比較を行うものである。theory には 1) で作った仮説の変数を入れる。empirics には 2.2.2 の 2) 3) 4) で求めた解の表示を代入した変数を入れる。ここでは 2.2.2 の 2) で求めた複雑解の変数 BettsCS を入れている。outcome には 2.2.2 の 2) 3) 4) などと同様の「結果」の変数を書く。intermed は、比較する解が中間解である場合には TRUE とし、他の場合には書かないか FALSE とする。

出力は次のようになる。

~CAPITAL*~YOUTH	CAPITAL*YOUTH	Sol.Formula	Theory	T*E	!*E	T*e	!*e	Out
*ILLITERACY*BET	*ILLITERACY*BE							come
WEEN	TWEEN							
1	0	1	1	1	1	0	0	0
2	0	0	0	0	0	0	0	1
3	1	0	1	0	0	1	0	0
4	0	0	0	0	0	0	0	1

一番左の列は、2.1.3 の 4) のファイルの読み込みのときに row.names で指定した変数を基に付けられた、事例の名前である。今の例では、2.1.3 の 4) で Case を行の名前に指定しているため、事例の名前が Case の番号で表されている。一番右の列 Outcome は、「結果」（今の例では Counsel）を表している。

左から 2 番目と 3 番目の列は、解の項である。今の例では 2 番目の列は 3 行目が 1 となっているが、これは解の項 ~CAPITAL*~YOUTH*ILLITERACY*BETWEEN でカバーされるのは、事例番号 3 の事例であるという意味である。

4 番目の Sol.Formula の列は、解（今の例では複雑解）全体でカバーされる事例を表しており、要するにいずれかの解の項でカバーされている事例を表す。今の例では Sol.Formula の列は 1 行目と 3 行目が 1 となっているが、これは 2 番目の列の解の項で事例番号 3 の事例がカバーされ、3 番目の列の解の項で事例番号 1 の事例がカバーされているからである。

Theory の列は、仮説の論理式でカバーされる事例を表している。今の例では仮説 Capital *(Illiteracy + Youth)* Between→Counsel でカバーされるのは、事例

資 料

番号1の事例であることがわかる。

T*eの列は、仮説（Theory）でカバーされかつ解（Empirics）でもカバーされている事例を表している⁽⁵³⁾。t*eの列は、仮説ではカバーされていないが解ではカバーされている事例を表している。T*eの列は、仮説ではカバーされているが解ではカバーされていない事例を表している。t*eの列は、仮説でカバーされておらず解でもカバーされていない事例を表している。T*eやt*eは仮説とデータが一致した部分であるが、t*eやT*eは予想されていなかった部分だと言える。そのため、後者の部分を検討することで、さらに理論を発展させることができる可能性がある。

2.3. QCAの分析手順の詳細（矛盾のある行、主項表）

1.3と同様に、表2のファイル（White.csvという名前とする）について分析を行う。以下の分析を行う前に、2.1.3の4）と同様の方法で、White.csvを代入した変数Whiteを作っておくものとする。

- 1）2.2.1と同様に、superSubsetコマンドで必要条件の分析を行う。
- 2）2.2.2の1）と同様にtruthTableコマンドで真理表を表示する。真理表が表示されるので、整合度（incl）の欄を確認して、この欄が0と1の間の小数になっている行があれば、その行は矛盾のある行である。1.3の3）と同様の真理表なので、inclが0.5の行がある。

矛盾のある行への対処は1.3の3）で説明したのと同様、次の2つの方法がある⁽⁵⁴⁾。①原因条件を追加したり、既存の原因条件の値を見直したりする。②整合度（incl）の値がある閾値以上の場合には結果の欄に1を入力し、ある閾値より下の場合には結果の欄に0を入力する。まず②の方法を説明する。

Rの場合はfs/QCAと違い、真理表のOUTの欄に数字が最初から入っているが、整合度（incl）が0.5の行の結果OUTの数字は0になっている。これは、truthTableコマンドでは整合度の閾値1がデフォルトなので、何も指定しないと、inclが1よりも小さい行は全てOUTの欄が0になるためである。整合度の閾値を別の値、例えば0.8に変えたいときは、truthTableコマンド内にincl.cut=0.8を付け加える。

すなわち、以下のようにする。

```
> WhiteTable <- truthTable (White, outcome="Counsel", conditions=c ("Capital", "Youth",
"Illiteracy", "Between"), complete=TRUE, show.case=TRUE, sort.by= "incl", incl.cut=0.8)
> WhiteTable
```

- 3) 矛盾のある行への処理が2)の②の方法の場合に、解を求める。2.2.2の2) 3) 4)と同様に eqmcc コマンドで複雑解、簡潔解、中間解を求められる。eqmcc コマンドでも、2)の truthTable コマンドと同様に整合度の閾値は1がデフォルトである。つまり incl.cut を書かない場合には、整合度の閾値は1として解が導出される。整合度の閾値を例えば0.8に設定するには eqmcc コマンド内に incl.cut=0.8を付け加える。例えば今の例で、整合度の閾値0.8で複雑解を導出するには以下のようにする。

```
> WhiteCS <- eqmcc (White, outcome="Counsel", conditions=c ("Capital", "Youth", "Illiteracy",
"Between"), details=TRUE, show.case=TRUE, use.tilde=TRUE, incl.cut=0.8)
> WhiteCS
```

- 4) 矛盾のある行への処理が2)の①の方法の場合を考える。今の例では、1.3の4)と同様に、表2の NoCourt を原因条件として追加して分析を行い直す。必要条件の分析や真理表の作成までは2.3の1)や2)と同様である。解の導出も、2.3の3)と基本的に同様であり、複雑解は表示に異なる部分はない。しかし、簡潔解や中間解の表示に多少異なる部分があるので、その点を説明する。簡潔解を導出するには次のようにする。

```
> White2PS <- eqmcc (White, outcome="Counsel", conditions=c ("Capital", "Youth", "Illiteracy",
"Between", "NoCourt"), include="?", details=TRUE, show.case=TRUE, use.tilde=TRUE)
> White2PS
```

資 料

すると、簡潔解が次のよう出力される。

M1: NOCOURT*BETWEEN + (ILLITERACY) <=> COUNSEL							
M2: NOCOURT*BETWEEN + (YOUTH) <=> COUNSEL							
	incl	PRI	cov.r	cov.u	(M1)	(M2)	cases
1 NOCOURT*BETWEEN	1.000	1.000	0.833	0.500	0.500	0.667	8;3;5;6;1
2 ILLITERACY	1.000	1.000	0.500	0.000	0.167		3;7;1
3 YOUTH	1.000	1.000	0.333	0.000		0.167	7;1
M1	1.000	1.000	1.000				
M2	1.000	1.000	1.000				

この出力では、M1とM2という2つの論理式が書かれている。これは、主項表での選択によって、簡潔解が2つありうるからである。1.3の5)で説明しているように、同じ例についてfs/QCAで解を導くときは、主項表で選択をする必要があった。それに対してこのRのQCAパッケージでは、主項表での選択はさせず、ありうる論理式をすべて表示するようになっている。分析者はその中から、分析者から見て最も妥当な形の論理式を選択して解とすればよい。

出力の表の中の(M1)の列は、M1の論理式における各項の固有被覆度を表している⁽⁵⁵⁾。今の例ではM1はNOCOURT*BETWEEN + ILLITERACY ⇔ COUNSELでNOCOURT*BETWEENの固有被覆度が0.500、ILLITERACYの固有被覆度が0.167となる。(M2)の列も同様である。

casesの列にはそれぞれの解の項でカバーされる事例が書かれている。カンマでなくセミコロン (;) で区切られている事例は、真理表の違う行に属する事例である。今の例では、解の項NOCOURT*BETWEENに属する事例は、事例5と事例6が同じ行に属し、事例8、事例3、事例5・6、事例1がそれぞれ違う行に属することがわかる。

表の下の方のM1の行には、M1の解の解整合度や解被覆度が書かれており、M2の行には、M2の解の解整合度や解被覆度が書かれている。

5) 矛盾のある行への処理が2)の①の方法の場合において、今度は中間解を次

質的比較分析 (QCA) のソフトの使用方法

のコマンドで導出する。

```
> White2IS <- eqmcc (White, outcome="Counsel", conditions=c ("Capital", "Youth", "Illiteracy",
"Between", "NoCourt"), include="?", dir.exp="1,1,1,1", details=TRUE, show.case=TRUE,
use.tilde=TRUE)
> White2IS
```

すると、中間解が次のように出力される。

```
p.sol: BETWEEN*NO COURT + ILLITERACY
```

```
Number of multiple-covered cases: 1
```

```
M1: BETWEEN*NO COURT + YOUTH*ILLITERACY*BETWEEN <=> COUNSEL
```

	incl	PRI	cov.r	cov.u	cases
1 BETWEEN*NO COURT	1.000	1.000	0.833	0.667	8;3;5,6;1
2 YOUTH*ILLITERACY*BETWEEN	1.000	1.000	0.333	0.167	7;1
M1	1.000	1.000	1.000		

```
p.sol: BETWEEN*NO COURT + YOUTH
```

```
Number of multiple-covered cases: 1
```

```
M1: BETWEEN*NO COURT + YOUTH*ILLITERACY*BETWEEN <=> COUNSEL
```

	incl	PRI	cov.r	cov.u	cases
1 BETWEEN*NO COURT	1.000	1.000	0.833	0.667	8;3;5,6;1
2 YOUTH*ILLITERACY*BETWEEN	1.000	1.000	0.333	0.167	7;1
M1	1.000	1.000	1.000		

p.sol は簡潔解を表しており、実際、4) で求めた 2 つの簡潔解と同じものが表示されている。中間解は、簡潔解に含めている論理的残余の行のうち、方向性予想を基に、含めることがもっともらしくないと判断された行 (困難な反事実的仮定) を除いてでき上がる解であった。そのため、前提となる簡潔解が違えば、そこから導かれる中間解も違ったものになる可能性がある。今の例では、簡潔解が 2 種類あったので、それぞれの簡潔解ごとにそこから導かれる中間解を表示しているのである。ただ、今の例では、結局どちらの簡潔解から導かれた中間解も同

資 料

じであることが、表示を見てみるとわかる。

6) 単純化の仮定や容易・困難な反事実的仮定の確認を2.2.2の5) 6)と同様に行う。今の例では、簡潔解の論理式は4)で見たように2つの可能性があったので、それぞれについての単純化の仮定が表示される。また、中間解は5)で見たように基になる2つの簡潔解ごとに別々に表示されるため、容易・困難な反事実的仮定もそれぞれについて表示される。

参考文献

石田淳・齋藤圭介監訳 (2016) 『質的比較分析 (QCA) と関連手法入門』 晃洋書房
(B. Rihoux & C. C. Ragin (2008) *Configurational Comparative Methods: Qualitative Comparative Analysis (QCA) and Related Techniques*, Sage Publications, Inc.)

鹿又伸夫 (1993) 「質的比較分析プログラムQCAについて」 立命館産業社会論集29 (2) , 85-132.

田村正紀 (2015) 『経営事例の質的比較分析：スモールデータで因果を探る』 白桃書房.

舟尾暢男 (2016) 『The R Tips 第3版：データ解析環境Rの基本技・グラフィックス活用集』 オーム社.

森大輔訳 (2010a) 「Fuzzy-Set/Qualitative Comparative Analysis ユーザーガイド」
(<http://park18.wakwak.com/~mdai/qca/software.html>) (C. C. Ragin 2008 “User's Guide to Fuzzy-Set/Qualitative Comparative Analysis 2.0.” Tucson, Arizona : Department of Sociology, University of Arizona.<http://www.u.arizona.edu/~cragin/fsQCA/software/fsQCAJapanese.pdf>)

森大輔訳 (2010b) 「Tosmana ユーザーマニュアル」 (<http://park18.wakwak.com/~mdai/qca/software/TosmanaJapanese.pdf>) (L.Cronqvist 2007 “Tosmana User Manual.” http://www.tosmana.net/tosmana_manual1_3beta.pdf)

森大輔 (2016) 「判例研究への質的比較分析 (QCA) の応用の可能性：米国の弁護士依頼権に関する判例の分析を例に」 熊本法学136、318-262.

Quaranta, M. & C. Q. Schneider (2013) “How To... Use Software for Set-Theoretic

質的比較分析 (QCA) のソフトの使用方法

- Analysis: Online Appendix to C. Q. Schneider and C. Wagemann (2012) ‘Set-Theoretic Methods for the Social Sciences’, Cambridge University Press,” Version 1.2, http://www.cambridge.org/download_file/857936
- Ragin, C. C. (1987) *The Comparative Method: Moving Beyond Qualitative and Quantitative Strategies*. University of California Press (鹿又伸夫監訳 1993 『社会科学における比較研究：質的分析と計量的分析の統合にむけて』 ミネルヴァ書房).
- Ragin, C. C. (2008) *Redesigning Social Inquiry: Fuzzy Sets and Beyond*. University of Chicago Press.
- Schneider, C.Q. & C. Wagemann (2012) *Set-Theoretic Methods for the Social Sciences: A Guide to Qualitative Comparative Analysis*, Cambridge University Press.
- Thiem, A. & A. Dusa (2012) *Qualitative Comparative Analysis with R: A User's Guide*, Springer.

注

- (1) QCAの邦語の解説については、森訳 (2010a)、田村 (2015)、石田・齋藤監訳 (2016)、森 (2016)などを参照。
- (2) ただし、大規模データの分析などにもQCAは使用されうる。Ragin (2008)など参照。
- (3) 英語文献では、例えばSchneider & Wagemann (2012)の付録であるQuaranta & Schneider (2013)において、様々なソフトウェアの使い方が解説されている。
- (4) 他に例えばTosmana, Kirq、統計ソフトStataのfuzzy adoファイル等がある。これらも有用であり、かつこれらでしか提供していない機能もあるが、いくつか重要な機能（特に中間解の導出）が備わっていないこともあり、本稿ではこれらは扱わないことにした。Compassのウェブサイトのソフトウェア紹介ページ (<http://www.compass.org/software.htm>) や筆者のwebページ (<http://park18.wakwak.com/~mdai/qca/software.html>) に主なソフトウェアの一覧がある。また、Tosmanaについては、旧バージョンのマニュアルの翻訳として森訳 (2010b) がある。

資 料

- (5) 本稿では fs/QCA は Windows 版のバージョン 3.0 を使用している。fs/QCA の使い方については、バージョン 2.0 のマニュアルの翻訳である森沢 (2010a) がある。また、田村 (2015) にも、使い方の記載がある。
- (6) 本稿では R は Windows 版のバージョン 3.4.0 を使用している。
- (7) 本稿では QCA パッケージはバージョン 2.6 を使用している。マニュアル (英語) などは <https://CRAN.R-project.org/package=QCA> にある。
- (8) 本稿では SetMethods パッケージはバージョン 2.1 を使用している。マニュアル (英語) などは <https://CRAN.R-project.org/package=SetMethods> にある。このパッケージは主に、Schneider & Wagemann (2012) に記載された分析方法を実行するためのものである。
- (9) この前身のソフトウェアの邦語の解説として鹿又 (1993) がある。
- (10) <http://www.socsci.uci.edu/~cragin/fsQCA/> で Windows 版と Mac 版の両方のプログラムの配布が行われている。
- (11) 一般に、表の縦を「列」、横を「行」と呼ぶ。
- (12) csv ファイルは、データをカンマで区切った形式ファイルであり、テキストエディタで作成・表示できるほか、Microsoft Office の Excel や LibreOffice の Calc 等でも作成・表示することができる。csv ファイルは、後に説明する R においても読み込みやすい形式であり、汎用性の高いファイル形式であると言えるため、ここではこれを用いることにする。
- (13) 統計データでは「変数」と呼ばれるものが、QCA では「集合」と呼ばれることになる。ただ、統計データも扱う研究者が多い影響から「変数」と呼んでしまうことも多い。実際、fs/QCA にも Variables (変数) というメニューがある。本稿でも「集合」「変数」両方を用いる。ただし、「集合の否定」「集合の論理和」とは言うが、「変数の否定」「変数の論理和」とは通常言わないので注意が必要である。
- (14) なお、2 つ以上の集合の論理和を、「原因条件」として使用する集合に設定することもできる。そのためには、Add Condition の下のプルダウンをクリックして集合を選んだ後、すぐ下の + という表示の下のプルダウンをクリックし

質的比較分析 (QCA) のソフトの使用方法

てもう1つの集合を選び、そして矢印のマークをクリックすればよい。ただし、これを行うのは、2つ以上の集合の論理和が必要条件になるかを確かめたい理論的根拠がある場合に限る方がよい。なぜなら、論理和を取ると集合が大きくなるため、必然的に必要条件になりやすくなってしまいう性質があるからである (Schneider & Wagemann 2012: 74)。

- (15) 閾値はかなり高くすべきで、Consistencyが例えば0.9を上回るような場合に初めて必要条件と判断すべきとされている (Schneider & Wagemann 2012: 143)。
- (16) ただし、森 (2016: 303) では事例数がまだ少ないことなどから、この段階で Illiteracy については必要条件と判断することを避けている。
- (17) 取るに足りない必要条件とは、例えば、次のようなものである。米国大統領になるという結果を考えるにあたって、米国出生であるという条件は確かに必要条件であるが、当てはまる例が多すぎて (結果の集合に比して条件の集合が大きすぎる)、多くの場合には、当たり前取るに足りない必要条件である。ただし、もちろん分析者の関心などによっては、取るに足りないと言えない場合もある。
- (18) ただし、現在 (2017年5月31日) の段階では、事例の番号や名前についての情報は保存されないという欠点がある。そのため、事例の番号や名前については、casesのボタンを押して表示されたものを、ファイルに手作業で転記していくしかない。後に紹介するRのQCAパッケージの方がこの点では楽である。
- (19) 右から二番目の PRI consist. はPRI整合度を表し、一番右のSYM consistはraw consist. と PRI consist. の積の値を表す。これらはraw consist. の代わりに使われることのある値である。基本的にはraw consist. を使用すればよいが、raw consist. とこれらの値が大きく乖離する場合には注意を要する。PRI整合度の詳しい意味については、森訳 (2010a: 96-98) 参照。
- (20) ただ、Delete current row to lastがグレーになっていて選択できないことがある。その場合には、以下の操作を試すとよい。今の例でいえば、5行目のCounselの欄をダブルクリックして入力状態にしたうえで何も入力せずに他の

資 料

行に移り、その後5行目の Counsel の欄をクリックし直して選択状態にする。

- (21) 詳しくは森訳 (2010a : 58-61)、Carsten & Schneider (2012 : 160-177)、森 (2016 : 297-302) などを参照。
- (22) ここでは、論理積でつながれた集合を項と呼んでおり、それが論理和でつながれたものが解の式である。
- (23) ベン図でいえば二つの集合の円が完全に重なるということになる。
- (24) この点は森 (2016) では触れなかった。
- (25) \Leftrightarrow の代わりに $=$ が用いられることもある。
- (26) 単純化の仮定が具体的にどの行についてなされているかは fs/QCA では表示されず、R の QCA パッケージでは表示可能である。2.2.2の 5) を参照。
- (27) 簡潔解は「とにかく論理式が簡単になるように」との観点で論理的残余を式に含めるか否かを決めており、含めることがもっともらしくない行まで含めてしまっている可能性があるため、これを最終的な解として提示することには慎重になるべきである (Schneider & Wagemann 2012 : 171)。中間解が通常最も解釈しやすいが、簡潔解は、必要不可欠な原因条件はどれかといったことを考える際に役に立つ (森訳2010a : 78)。
- (28) これらが具体的にどの行かは fs/QCA では表示されず、R の QCA パッケージでは表示可能である。2.2.2の 6) を参照。容易・困難な反事実的仮定について詳しくは、森訳 (2010a : 58-61)、Carsten & Schneider (2012 : 160-177)、森 (2016 : 297-302) などを参照。
- (29) 中間解の論理的残余の行の含め方はあくまで一つの方法であり、これ以外の含め方もありうるし、中間解における論理的残余の行の含め方では問題のある場合 (その一部は次回 (2) で紹介する) もある。Kirq が中間解を導出しないのは、そうした理由からのようである。http://grundrisse.org/qca/docs/kirq.html を参照。
- (30) 解の論理式が必要十分条件である場合には、結果の存在の論理式が分かると、結果の不存在の論理式はド・モルガンの法則で簡単に求めることができるように思うかもしれない。例えば、1.2.2の15) では中間解の論理式が Illiteracy *Between \Leftrightarrow Counsel だったので、ド・モルガンの法則を使うと \sim Illiteracy+

質的比較分析 (QCA) のソフトの使用方法

~Between⇔~Counselとなる。しかし、ド・モルガンの法則の単純な適用による論理式と、データから導かれる解の論理式はずれることが多い（実際、今の論理式も1.2.3で導かれる論理式と異なる）。そのため、結果の不存在についての解の論理式が知りたい場合には、以下で説明するようなソフトウェアを用いる方法で別個に分析するべきである（Schneider & Wagemann 2012 : 279）。

- (31) これら3つの論理式は、1.2.2の15)の3つの論理式にド・モルガンの法則を適用したものとは異なっているものがあることに注意する。
- (32) より詳しく様々な処理の仕方について解説したものとして、Schneider & Wagemann (2012 : 120-123)、石田・齋藤監訳 (2016 : 65-67) 参照。
- (33) 森 (2016 : 292) では、この処理の仕方と法学の判例研究における「区別」という方法との類似性について指摘している。
- (34) ファジ集合の場合には、raw consist. は小数のことがほとんどである。
- (35) 他に Delete rows width number less という欄もある。これは事例の数についての閾値を設定する欄であるが、今はこの欄は1のままでよい。多数事例の研究で、ある行に1つの事例が存在しているだけでは、測定誤差のせいである可能性があると考えられる場合に、閾値を1でなく2にしたりすることがある。Schneider & Wagemann (2012 : 153, 182) 参照。
- (36) 閾値をいくつにしたらいかがというはっきりした基準はないが、Schneider & Wagemann (2012 : 127-128) では次のようなことが述べられている。まず、0.5のような低い値にすべきではなく、0.75でも十分ではない。また、先行研究等からの理論的な予想が確たるものであるほど高い閾値を使った方がよく、事例数が少ないほど低い閾値を使った方がよいといったことも言える。さらに、真理表の行を、整合度が高いものから低いものへ、という順に並び替えると、高い値と低い値の間に、整合度の閾値の決定の指針となりうるような、明白なギャップができることが多い、ということも指摘されている。
- (37) この際 Youth と Illiteracy のどちらを選ぶべきかが問題になるが、これは森訳 (2010a : 61) ではユーザーが理論的・実質的知識を用いて行うとされている。しかし、実際上これは困難である。代わりに取りうる方法の一つは、全ての

可能性を試して、出てくる解を比べてみて、解が分析者から見て最も妥当な形となるものを選択するというものである（森2016：264）。実は主項表での選択で解が変わるのは多くの場合簡潔解のみで、複雑解や中間解はかなりの場合、どれを選択しても最終的に同じ解になる（今の場合の Youth と Illiteracy もどちらを選択しても同じ解になる）ということもあるので、このように試してみるというのは有効な方法である。こうしたこともあるので、Kirq では主項表での選択をユーザーにさせず、可能な解を最初から全て表示する、という方法を取っている。<http://grundrisse.org/qca/docs/kirq.html> を参照。また、後に見るように R でも同様である。

- (38) R の使い方の解説として、例えば舟尾（2016）参照。
- (39) Stata にも R のパッケージに当たる ado ファイルと呼ばれるものがあり、fuzzy ado ファイルにより QCA の分析を行うことができる。Schneider & Wagemann（2012：320）は、R や Stata といった量的研究者が頻繁に使用するソフトに QCA がこのように実装されたことは、量的研究者の間で QCA が認知されたことの証であり、量的研究者の間でのさらなる認知の引き金にもなると評価している。
- (40) ディレクトリとは、Windows においてフォルダと呼ばれるもののことである。
- (41) つまり、`> Betts <- read.csv ("Betts.csv", row.names="Case")` と書かれているときは、`Betts <- read.csv ("Betts.csv", row.names="Case")` の部分を入力する。また、R では半角空白は無視されるため、`<-` の前後の半角空白は見やすさ重視で入れているだけで、実際は半角空白はあってもなくてもよい。
- (42) 最初にフォルダの移動をしていない場合は、ファイルのパス名（ファイルの場所を含む形での名前）を記入する必要がある。例えば、Betts.csv が C ドライブ直下にある場合、“Betts.csv”の部分は“C:\Betts.csv”とする。
- (43) R での代入の記号は `<-` である。
- (44) 複数の変数を使用する場合には、`c ()` で囲む。例えば今の例で、Capital のみを「原因条件」に指定するときは `conditions="Capital"` でよいが、Capital、Youth、Illiteracy、Between の 4 つを「原因条件」に指定するときは、

質的比較分析 (QCA) のソフトの使用方法

conditions=c (“Capital”, “Youth”, “Illiteracy”, “Between”)とする。

- (45) 注15参照。また、incl.cutを指定しないと閾値 1 として出力される。
- (46) 例えば、Capital という集合は CAPITAL と表され、その否定は capital と表される。
- (47) RoN の列は、Relevance of Necessity を表す。これは、Schneider & Wagemann (2012: 236) において、必要条件の被覆度では表しきれない種類の瑣末さがあるとして、必要条件の被覆度の代わりに提案された、必要条件の瑣末さの度合いの指標である。通常は incl と cov.r の値を見れば十分だが、RoN と cov.r の値が大きく乖離する場合には注意を要する。
- (48) 注14で述べたように、特に論理和を必要条件とすることには注意を要する。ただ、このように閾値を超えたものを全て出力してくれることで、分析者が気づいていなかった必要条件が見つかることもありうるため、この点は fs/QCA と比べて利点である。
- (49) したがって本来は、truthTable (Betts, outcome = “Counsel”, conditions = c (“Capital”, “Youth”, “Illiteracy”, “Between”), complete=TRUE, show.case=TRUE, sort. by = c (“n”, “incl”)) という部分だけで、真理表は表示できる。
- (50) 同様に PRI は PRI 整合度を表しており fs/QCA の PRI consist. に当たる。
- (51) 代わりに eqmcc () の括弧内に、explain=“0” を付け加えてもよい。なお、これと outcome=“~Counsel” を併用しても、否定同士でキャンセルされ合って肯定になったりすることはなく、1つの効果のみが適用され否定と解釈される。
- (52) 注30と31で述べたように、2.2.2の解にド・モルガンの法則を適用したものと、ここで導いた解は異なるものになる。なお、ド・モルガンの法則を適用した論理式は deMorgan コマンドで得ることができる。例えば Illiteracy*Between にド・モルガンの法則を適用した論理式は deMorgan (“ILLITERACY*BETWEEN”, use.tilde=TRUE) で得ることができる (変数は全て大文字で書く必要がある。また、use.tilde=TRUE は、集合の否定をチルダの形式で表示してほしい場合に用いる)。

資 料

- (53) この分析方法については、Schneider & Wagemann (2012 : 295-305) 参照。
- (54) R の QCA パッケージでは他に二項検定を使った方法も使用できる。ただし、これは事例数が多くないと適切に機能しないうえ、ファジィ集合では使用できない。また、石田・齋藤監訳 (2016 : 66) でも指摘されているように、こうした確率的な方法は事例指向的な QCA の特徴と合わない面もある。こうした理由から、ここではこの方法は解説しない。
- (55) 解の論理式が複数存在する場合、cov.u の列には、その複数全てに共通する項のみ固有被覆度の値が書かれ、それ以外の項は cov.u の欄には 0 が表示される。今の例でも、2 つの論理式に共通する NOCOURT*BETWEEN のみ固有被覆度 0.500 が書かれている。