

選好集計理論に基づく認知的モデリング

A Cognitive Modeling Based On the Preference Aggregation Theory

犬童健良(関東学園大学)

Kenryo INDO (Kanto-Gakuen University)

要点:

1. 選好集計理論を限界合理性のモデリングに応用し、誤表象の認知的モデリングを論じる。
2. 従来、選好集計理論は、社会的選択の分析に用いられた。本論文では前診断的プロセス、つまり日常的推論やオフィスにおける断片的な情報の統合に適用する。
3. 例題として条件文推論の検証(選択課題)を取り上げ、論争ジレンマをその実用的スキーマとして再解釈する。
4. またそれを通じて、前診断的プロセスの権利システムとしての側面に光を当てる。

1. はじめに

限界合理性のモデリング

- 知識の欠如: 正しい解法(アルゴリズム)を知らない。
- 推論の失敗, 適応的判断: ときどき誤りを犯すが, 手早く, 節約的な発見的ルール集まり[1]を用いて, さほど悪くない解を手早く見つけ, 環境の変化に適応できる。

満足化と誤表象

- 一方, 誤った知識・信念において, 学習が停止するという意味で, 満足化が誤表象を発生させる。

希望をもった懐疑主義: 進化的認識論[2]との類比

- 反証主義はその処方になりうるだろうか?

前診断的プロセス

- ある知的課題の遂行に役立つように, 断片的な情報を統合するプロセス。日常的な推論からビジネス上の重要な判断や意思決定などにわたる。後者では, 例えば KPI や BSC。
- 本論文では, 単純な条件文の検証(選択課題[3,4])の場合を論じ, 選好集計の背理(論証ジレンマ[5])をこれと結び付ける。またその権利システム[8]としての側面に光を当てる。

2. 条件文と前診断的プロセス

- 今, ある情報システム(認知システム)《A》の業務ロジックは, 単純な条件文 $p \rightarrow q$ の論理的意味 $(\text{not } p) \vee q$ (実質含意)にしたがうものとする。また《B》は, テスト用データを使って, 《A》の入出力を検証するその検証用システム(メタ認知システム)であると仮定する。
- 《B》がプログラム, あるいは人間のいずれであるにせよ, 真理表で与えられた宣言的知識(上記の論理的意味)とは別に, 検証作業を正しく遂行するための何らかの手続きを使っているはずである。
- 課題遂行の手続きに対し, より直接的に役に立つ情報(すなわちその実用的な意味)を作り出すことを, 本論文では前診断的プロセス(pre-diagnostic process)と呼ぶことにする。また実用的スキーマ(pragmatic schema)を, それに適合する選択手続き(choice procedure)とのペアとして定義する。

2.1. 反証主義と満足化

一つの単純な条件文を検証するための前診断的プロセスは, 例えば, 表1のように表せる。

表1 実質含意に対する規範的な前診断的プロセス

observation data	satisficing set	complaint set	concerning set
p	{q}	{not q}	{not q}
q	{p, not p}	ϕ	ϕ
not p	{q, not q}	ϕ	ϕ
not q	{not p}	{p}	{p}

- 観察可能なデータの集合 $D = \{p, \text{not } p, q, \text{not } q\}$
- 満足化集合(satisficing set) $S \subseteq D$: Dの部分集合でポジティブな証拠
- 愁訴集合(complaint set) $C \subseteq D$: 同じくネガティブな証拠。
- 懸念集合(concerning set) $K \subseteq S \cup C$: 選択手続きを左右する部分。

前診断的プロセスは制度(権利システム)[8]とみなせる。

すなわち $K \neq \phi$ のとき, かつそのときのみ検査を実行許可する。

反証主義のスキーマ

主張 1. 反証主義者の懸念集合は愁訴集合に一致する.

反証主義者(Popperian)の懸念集合の要素は矛盾データペア (の片割れ) である. すなわち, 前診断的プロセスと懸念集合の合成を, $p \rightarrow \{\text{not } q\}$; $\text{not } q \rightarrow \{p\}$; それ以外は空集合 ϕ と表せる.

満足化原理

主張 2. エージェントが (反) 満足化にしたがって検査を実行するのは, 満足化集合と愁訴集合に対立する要素のペアがあって, かつ懸案集合がその一方の要素を含むときである.

主張 2 に合致するエージェントの信念は, 互いに競う形の 2 つ以上の実用的ルールから成り立ち, 典型的には次のようである.

$$(p \rightarrow q) \ \& \ (r \rightarrow s) \ \& \ (p \vee r) \ \& \ \text{not } (p \ \& \ r).$$

cf., 理由に基づく選択

対立する理由が多い選択肢がより選ばれやすい [9].

上記のタイプの満足化に特徴的な傾向

- 理由 (達成動機) p が十分強ければ, 選択肢 q を選び, 別の理由 r が強ければ, 選択肢 s を選ぶ.
- またおうおうにして, ほぼ q と決めているが, p が十分でない. 例えば $q \rightarrow q_1 \vee q_2$, q_1 は好ましいが q_2 は回避したい理由があり, 一方 r についてはほぼ無知であるといった場合.
- そのため, p を裏付ける (あるいは r を否定する) 情報を積極的に獲得し, またそれによって希求水準を高めようとする.
- 一方ルール自体に矛盾する情報には耳を貸そうとしない. 例えば, 「たとえ s を選んでも r が達成できるとは限らない」と 「 q を選んでも s は達成可能である」など.

推測. 日常の推論 (の前診断的プロセス) は, すべてのルールが使えなくなる事態を回避しようとする. そのため誤った信念を温存する (確証バイアス). \Rightarrow 満足化行動の説明

2.2. 選択課題と許可スキーマ

Wason の選択課題(selection task) [3]

一つの面に文字, 別の面に数字が印刷された 4 枚のカードから, 以下のルールが守られているかを確かめるのに裏返す必要があるカードを選びなさい.

「一方の面が A のカードは, その別の面に 9 が記入されている。」

4 枚のカード

A	C	8	9
---	---	---	---

確証バイアス

- 実際の回答パターンでは「 p と q 」や「 p 」がよく観察される. 正解「 p と $\text{not } q$ 」はとても少ない. 前者は確証バイアス, とくに前提と結論の形に一致するものを選ぶマッチングバイアス. 抽象的な出題で生じやすい. 一方, $p \rightarrow \text{not } q$ のように結論を否定形にすると, 検証バイアスが生じ, 正解率が向上する.

許可スキーマ

- 形式的には同じはずの問題でも, 問題の内容や被験者の選び方によって, 正解率が変化する. ところが, 許可スキーマ (permission schema) を与えると, それらによらず安定して正答率が高い.

例) 被験者に対して, 1 組のカードとリストが提示される. 検証されるルール:

IF: カードに「搭乗」と記入されていたら,
THEN: リストの中に「コレラ」が記入されている.

上記例題に対する許可スキーマ:

空港の搭乗手続きの状況であなたが搭乗者の検疫チェックを担当していると想定せよ. カードは人物が搭乗予定者であるかどうかを示し, リストは個人の検疫の記録である. 乗客の安全を守るため, それをチェックしなければならない.

3. 選好集計に基づく認知的モデル

- 重要な意思決定の前診断的プロセスでは、複数の異なる評価や意見を集計することが必要になる。

例) 専門家が集まった委員会による答申作成や、マネージャーが管理データやオフィスの各メンバーの意見を聞いた上で行う判断など。

- しかしたとえ全員が合理的な考えを持っていても、集計の結果得られるグループ全体としての考えが合理的になるとは限らない。

3.1. 論証ジレンマ

- 選好集計は、選好関係(順序)の集計手続きであったが、論理的命題の集計にそれを適用すると、例えば、表 2 に示されるような結果が得られる。
- 近年、こうした論証ジレンマ(discursive dilemma)と呼ばれる判断集計のパズルとその解決が研究された[5]。

例) p : 「TCO 削減率が 50%を超える」、 q : 「システムを前面更新する」、 $\text{not } q$: 「部分的な改良で済みます」とすると、 p と $p \rightarrow q$ はともに多数決で採決され、ゆえに q を推論できる。しかし q は多数決で否決される。

表 2 論証ジレンマ.

member	p	$p \rightarrow q$	q
Taro	Yes	Yes	Yes
Jiro	No	Yes	No
Hanako	Yes	No	No
majority	Yes	Yes	No

表 2 は Condorcet の見出した多数決投票の循環と同等である。

- **命題 1.** 論証ジレンマは、命題間の含意関係についての投票とみなせば、投票の背理である。

証明. 命題間の含意関係は、反射的、推移的、完備的であるから、選好関係(weak order)に読み替えられる。また表 5 における各列の見出しは、それぞれ $\blacksquare \rightarrow p$, $p \rightarrow q$, $\blacksquare \rightarrow q$ と論理的に等しい。ただし \blacksquare は恒真を表す。Taro と Hanako は $\blacksquare \rightarrow p$ にかんして、Taro と Jiro は $p \rightarrow q$ にかんして、そして Jiro と Hanako は $q \rightarrow \blacksquare$ にかんして、それぞれ過半数を確保する。よって多数決の結果は、 $\blacksquare \rightarrow p$, $p \rightarrow q$, $q \rightarrow \blacksquare$ となり循環する。

3.2. 反証主義者のモデリング

- 個人の投票を「真偽」ではなく、それを選択課題における「検査の必要性」とみなせば、前節のジレンマは解消する。

命題 2. 表 2 は反証主義の手続きを構成する。

証明. グループ全体として反証主義の手続きを次のように構成する。各行は「Yes の値のついた列は、見出しの命題にマッチするデータが観察されたら、疑わしいので検査するべきだ」という主張であると解釈する。多数決の結果(最下行)は、 p について検査し、 q については黙認する。 $p \rightarrow q$ については、懸案だが、選択課題では直接観察されることはない。またそれぞれの否定については、各個人の値を反転して $p \rightarrow q$ と組み合わせることによって、同様に $\text{not } p$ については黙認、 $\text{not } q$ については検査となる。

- すなわち、表 2 の最下行は、多数決で Yes となる命題は懸案集合の要素である。あるいは任意の結託がデータ検証手続きにアクセスする権利の設定とみなせる。さらに、それは投票ゲームにおける権力の配分(勝利提携)によって間接的に実現されている。

4. 満足化のモデリング

前節では論証ジレンマを再解釈して科学者の規範的スタンスをモデリングした。本節はこれに対する 2 つの批判を検討しよう。① 3 人のエージェントに当てた順序の下での投票結果がたまさか規範的だったにすぎないのではないか、また② エージェントたちはどのように多数決による集計やその結果に合意できるのか。

3.1. エージェントの選好について

- 第 1 の批判に対しては、次のように、反証主義は基本的にこのパターン(ラテン方格)を含む必要があることを指摘できる。
- $p \rightarrow q$ と推移性に矛盾せず、一つのデータを観測した後の含意関係は以下の 4 パターンに要約できる。
 - ① $\text{not } q \rightarrow \text{not } p \rightarrow \square \rightarrow \blacksquare \rightarrow p \rightarrow q$
 - ② $p \rightarrow \square \rightarrow \{q, \text{not } q\} \rightarrow \blacksquare \rightarrow \text{not } p$
 - ③ $\text{not } q \rightarrow \square \rightarrow \{p, \text{not } p\} \rightarrow \blacksquare \rightarrow q$
 - ④ $p \rightarrow q \rightarrow \square \rightarrow \blacksquare \rightarrow \text{not } q \rightarrow \text{not } p$
- これらは上からそれぞれ p , $\text{not } p$, q , $\text{not } q$ が観察された場合に対応する。①～④で恒偽□と恒真■はつねにこの順で含まれ、また $\{q, \text{not } q\}$ や $\{\text{not } p, p\}$ はその内側の順序が確定しないことを表す。
- 表 2 の論証ジレンマに戻って考えると、①と③④が最初の 2 人の投票者、また 3 人目は次のような順序をもっていた。
 - ⑤ $\{q, \text{not } p\} \rightarrow \square \rightarrow \blacksquare \rightarrow \{p, \text{not } q\}$
- これは前の 2 人の合意するペア $p \rightarrow q$ とは対立し、双方と合意できるペアを 1 つずつもつ。

命題 3. 含意関係が命題 1 の意味で選好関係とみなせるとき、単純多数決によって反証主義的な実用的スキーマが得られるのは、基本的に表 2 のパターンだけである。

証明. 2 項関係が完備的である場合、表 2 とその各列の逆向きの含意について値反転したパターンか、もしくはそのすべての反転パターンの 2 パターン(ラテン方格)だけが強く循環する。また準推移性に違反するのは、Salles の定理 3.7 により、このパターンを部分的に含み、かつある 3 命題と 3 エージェントについて CD (循環的依存性) に違反する場合、かつそのときに限られる。すなわち

- (1) だれも線形順序でない、
 - (2) 上のラテン方格で高々 1 人弱順序に変えたもの、
 - (3) 1 人だけ線形順序のまま、その両極のペアについて後の 2 人がそれぞれ無差別(両方向 Yes)になる
- のうちのいずれかである。しかし(1)と(2)では投票結果はすべて無差別となり、また(3)は後述する命題 3 から反証主義を覆す。

- 残された第 2 の批判に対しては、暫定的にそれが一つの前診断的プロセスなのだと答えよう。本論文はモデルの解釈(メタモデル)の多様性と誤表象の可能性を基本的に認める。実際、検証必要性の解釈を用いなくても、⑤のような一見玉虫色のスタンスが導く循環が、勝者敗者の顕在化を回避する「判断の先延ばし」として解釈しうる。
- 法廷モデル： 論争ジレンマの別の解決を与えるモデルは紛争解決の場としての法廷である。Taro は顧客で Jiro の会社と契約したが、Jiro の契約不履行を訴えているとする。つまり p : 契約不履行があった、 q : 会社はその責任を負う。Hanako は証人として喚問され、法廷が判断を下すものとしよう。この場合、多数決は選択手続きとして不相当であり、法廷は p については Hanako の意見を、また $p \rightarrow q$ は紛争当事者間の合意をそれぞれ尊重し、矛盾なく q を導く。

3.2. 確証バイアスを発生させるメカニズム

- 条件文推論をめぐる認知科学研究が示しているように、人々は必ずしも科学者のように推論しない。そこで、最後に主張 2 で述べた限界合理性のモデリングを試みる。

命題 3-4. 反証主義のスキーマは確証バイアスを派生しやすい。またそれは安定している。

証明. 表 2 における Hanako のスタンスが、「ある条件が真ならば q は検査不要だが、それ以外は検査が必要である」に弱まったとする。これを 0* と書く。このとき Taro と Hanako が結託するとジレンマ(反証主義)を覆す可能性がある。さらに Jiro も 0* に変わると単独の変化では No ($q \rightarrow \blacksquare$) を多数決で導けない。またこの場合、Taro の独裁となる。り、いかなる結託もこれを覆さない。

表 2' 論証ジレンマの変形。

Member	p	$p \rightarrow q$	q
Taro	Yes	Yes	Yes
Jiro	No	Yes	0*
Hanako	Yes	No	0*
majority	Yes	Yes	Yes

5. 関連研究

- 現実の意思決定は、合理的に問題を解こうとしても、複雑でうまくいかないことがある。そこで、最適解を見出す代わりに、より扱いやすい単純な問題に置き換えて、それをできるだけ厳密に解くことによって、満足する (H. A. Simon 流の満足化原理)。
- その具体的なモデリングはさまざまである ([1]を参照)。しかし制約つき最適化、最適停止、非期待効用などの初期のアプローチは適切でないと考えられるようになった。
- エージェントベースモデリングないしエージェントシミュレーション[10]は、合理的選択について演繹するための公理系も、計量経済学的手法による実証分析も必要ない。その代わりに、行動ルールを十分単純化し、相互作用の創発的パターンに注目する。また認知的モデリングとの融合が探求されている[11]。
- 選好集計理論は、公理的アプローチではあるが、本論文で述べたように誤表象とその変化に対するモデリングを提供しうる。

まとめ

- 選好集計理論を限界合理性の認知的モデリングに応用し、前診断的プロセスにおける誤表象発生を論じた。
- 例題として条件文推論の検証 (選択課題) を取り上げ、また論争ジレンマをその実用的スキーマとして再解釈した。
- これらを通じて、前診断的プロセスの権利システムとしての側面に光を当てた。
- 命題 3 と命題 4 より、反証主義は満足化に取って代わられる傾向がある。(反証主義を実現する) 多数決の権利配分は安定ではない。命題 4 の証明で述べた選好変化によって (満足化を実現する) 独裁的ないし寡頭的な権利配分を取って代わられると、安定する傾向があると考えられる。
- 許可スキーマが反証主義手続きを安定させる一つの理由として、上記のレジーム交代を防止するからである。

付録. PROLOG を用いた実験

- 実験 1. 選択課題に対する前診断的プロセスのモデリングを、PROLOG を用いて実装し、シミュレーションする (別紙 1)。
- 実験 2. 表 2 の選好集計における多数決や独裁などの任意の権利ないし権力の配分は、単純ゲーム (およびそこから導かれた効力関数) としてモデリングされる。これを PROLOG によって実装する (別紙 2)。
- 実験 3. またその安定性、すなわちコア (勝利提携内の合意によって阻止されない結果の集合) が常に非空であること (優越関係の非循環性)、および準推移性への違反 (循環的依存性条件) について実験する (別紙 3)。

参考文献

- [1] Gigerenzer, G. and Selten, R.(eds.), *Bounded Rationality: The Adaptive Tool Box, Dahlem Workshop Reports*, MIT Press, 2001.
- [2] Popper, K.R., *Objective Knowledge*, Clarendon Press, 1972. (森博訳, 『客観的知識: 進化論的アプローチ』, 木鐸社, 1974)
- [3] Evans, J.St.B.T., *Bias in Human Reasoning: Causes and Consequences*, Lawrence Erlbaum Associates Ltd., 1989.
- [4] Holland, J.H., Holyoak, K.F., Nisbett, R.E. and Thagard, P.R., *Induction: Process of Inference, Learning, and Discovery*, MIT Press, 1986. (市川伸一訳, 『インダクション』, 新曜社, 1991)
- [5] Dietrich, F., "Judgment aggregation: (im)possibility theorems," *Journal of Economic Theory* Vol.126(1): 286-298, 2006.
- [6] Sen, A., *Choice, Welfare and Measurement*, MIT Press, 1982.
- [7] Gaerter, W., *Domain Conditions in Social Choice Theory*, Cambridge University Press, 2001.
- [8] Peleg, B., "Effectivity functions, game forms, games, and rights," *Social Choice and Welfare* Vol.15: 67-80, 1998.
- [9] Shafir, E., Simonson, I. and Tversky, A., "Reason-Based Choice Theory," *Cognitive Science* Vol.49: 11-36, 1993.
- [10] Tesfatsion and K. L. Judd (eds.), *Handbook of Computational Economics, Vol. 2: Agent-Based Computational Economics*, Elsevier, 2006.
- [11] Sun, R., *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*, Cambridge University Press, 2005.

```
% the Wason's selection task
%-----
card(no(1), 'A', 3).
card(no(2), 'D', 4).
card(no(3), 3, 'B').
card(no(4), 7, 'C').
(中略)
% possible observations for the selection task
data_item(p).
data_item(not(p)).
data_item(q).
data_item(not(q)).

% possible types of agent
type_of_agent(poppernian).
type_of_agent(ordinary).

% satisficing set: positive (or affirmative) evidences
satisficing_item(poppernian, p, q).
satisficing_item(poppernian, q, p).
satisficing_item(poppernian, q, not(p)).
satisficing_item(poppernian, not(p), q).
satisficing_item(poppernian, not(p), not(q)).
satisficing_item(poppernian, not(q), not(p)).
satisficing_item(ordinary, p, q).
satisficing_item(ordinary, q, p).
satisficing_item(ordinary, not(p), not(q)).
% satisficing_item(ordinary, not(q), not(p)).

% complaint set: negative (or denial) evidences
complaint_item(poppernian, p, not(q)).
complaint_item(poppernian, not(q), p).
complaint_item(ordinary, p, not(q)).
complaint_item(ordinary, q, not(p)).
% complaint_item(ordinary, not(p), q).
complaint_item(ordinary, not(q), p).

% the prediagnostic process
satisficing_set(T,A,B):-
    type_of_agent(T),
    data_item(A),
    findall(X, satisficing_item(T,A,X), B).
complaint_set(T,A,C):-
    type_of_agent(T),
    data_item(A),
    findall(X, complaint_item(T,A,X), C).
concerning_set(T,A,D):-
    type_of_agent(T),
    data_item(A),
    findall(X, concerning_item(T,A,X), D).
```

```
% concerning set: directly affects the choice procedure
concerning_item(poppernian,A,B):-
    complaint_item(poppernian, A, B).
concerning_item(ordinary,A,B):-
    satisficing_item(ordinary, A, B),
    complaint_item(ordinary, A, not(B)).
concerning_item(ordinary,A,B):-
    satisficing_item(ordinary, A, not(B)),
    complaint_item(ordinary, A, B).

% the choice procedure
choice_procedure(T,A,C):-
    data_item(A),
    possible_choice(C),
    rule_for_choice_procedure(T,A,C).
possible_choice(inspect).
possible_choice(not(inspect)).
rule_for_choice_procedure(T,A,inspect):-
    type_of_agent(T),
    concerning_item(T,A,_B),
    !.
rule_for_choice_procedure(T,A,not(inspect)):-
    type_of_agent(T),
    %+ concerning_item(T,A,_B).

% demo
/*
?- type_of_agent(T), choice_procedure(T,A,C),
nl, write(T: (A->C)), fail.

poppernian: (p->inspect)
poppernian: (not(p)->not(inspect))
poppernian: (q->not(inspect))
poppernian: (not(q)->inspect)
ordinary: (p->inspect)
ordinary: (not(p)->not(inspect))
ordinary: (q->inspect)
ordinary: (not(q)->not(inspect))

No
?-
*/
```

表3 満足化を行う前診断的プロセスの一例

observation data	satisficing set	complaint set	concerning set
p	{q}	{not q}	{not q}
q	{p}	{not p}	{p}
not p	{not q}	φ	φ
not q	φ	{p}	φ

(別紙2) ページ1 / 1

以下は Prolog プログラム sp06d.pl のコードと実験結果から、一部を抜粋して示す。

表4 Prolog プログラム sp06d.pl で扱われる選好集計のモデリング

モデリング	人数	生成	主な項目
選好関係	2, 3, ...	可	各種の選好関係
領域制限	2, 3, ...	可	価値制限, 循環的依存等
社会的選好	2, 3	可	SWF, SDF (社会的選好とそれに基づく選択)
単純ゲーム	2, 3	可	固有性, 強性, 弱性, コア, 安定性等
効力関数	2, 3	可	超加法性, 極大性, 凸性, コア, 安定性等

```
% simple game and the winning coalitions
```

```
%-----
```

```
:- dynamic win/2.
```

```
/*
% example 1
win( [], no).
win( [1], yes).
win( [2], no).
win( [1,2], yes).
*/
```

```
% example 2
win( [], no).
win( [1], yes).
win( [2], no).
win( [3], no).
win( [1,2], yes).
win( [1,3], no).
win( [2,3], no).
win( [1,2,3], yes).
```

```
% demo for a simple game (example 2)
```

```
/*
?- verify_win.
```

```
game:[[1], [1, 2], [1, 2, 3]]
is proper
is weak with veto players:[1]
is inessential but nondictatorial
```

```
Yes
?-
*/
```

```
% effectivity function
```

```
%-----
```

```
:- dynamic eff/2.
```

```
set_eff(M):-
    eff_model_base(M,E),
    abolish(eff/2),
    forall(member(eff(A,B),E),assert(eff(A,B))).
```

```
% example eff(1): a reproduction of foregoing example of
% a dictatorial simple game
```

```
eff( [1], [x]).
eff( [1], [y]).
eff( [1], [z]).
eff( [1], [x,y]).
eff( [1], [x,z]).
eff( [1], [y,z]).
eff( S, [x,y,z]):- group(S,_).
eff( [1,2], E):- event(E,_),EY=[],EY=[x,y,z].
```

```
% demo
```

```
/*
?- switch_mode_effectivity(A).
```

```
A = win->eff
```

```
Yes
?- verify_eff.
```

```
group:[ ] is_effective_for
group:[3] is_effective_for
group:[2] is_effective_for
group:[2, 3] is_effective_for
group:[1] is_effective_for
[x]
[y]
[z]
[x, y]
[x, z]
[y, z]
group:[1, 3] is_effective_for
group:[1, 2] is_effective_for
group:[1, 2, 3] is_effective_for
is superadditive
is regular
is stable
is strongly stable
```

```
Yes
?-
*/
```

単純ゲーム, 効力関数, 安定性: 権利システムの数理的基礎として

- 単純ゲーム (simple game) は 1 人以上の可能なサブグループ (結託) のすべてのうち, 単調性や固有性などの性質を満たすように特定の部分集合を勝利提携として選んだもの. 権利 (ないし権力) の配分を意味する. 優越関係が成り立つ代替案同士の比較, すなわちその結託の内部のエージェントが強選好で合意すると, 選好集計ないし社会的選択はそれに一致しなければならない.
- 特徴ゲームフォームは, 任意の代替案の非空部分集合にかんして, 単純ゲームと同様に, 特定の結託の集まりを選び効力 (effectiveness) を与える. 効力関数 (effectivity function) は, 結託や代替案集合に関して単調な特徴ゲームフォームであり, また社会 (グループ) 全体はつねに任意の非空部分集合にかんして効力をもつ. 単純ゲームはすべての非空の代替案部分集合において, 同じ結託の集合が効力を割り当てられている特殊な効力関数である.
- 単純ゲームと効力関数は, 優越関係を通じて共通のモデリングが可能であり, またそのコアは優越されない代替案の集合である. 安定性は, 任意の選好プロフィールに対してコアが非空であること. またそれは優越関係がサイクルをなさないことと同値である.

選択課題の実用的スキーマとしての論証ジレンマ: 権利システムとしての安定性

- 本論文では前診断的プロセスの例題を, 権利システム [8] として解釈することを試みた. 懸念集合 K は情報断片 $X \subseteq D$ に対して割当てられた検査の権利である. 選択手続きは権利へのアクセス, 行動集合 $B = \{X \text{ を検査する}, X \text{ を検査しない}\}$ の部分集合への写像である. 反証主義の例では, 観察データによらず, $K = X \cap C \neq \emptyset \Leftrightarrow$ 検査の権利 (の実行) とされた. また $P(Y)$ を集合 Y の非空部分集合族とすると, 実用的スキーマは効力関数 (特徴ゲームフォーム) $f : P(D) \rightarrow P(B)$ である.

PROLOG を用いた実験

- 以下は PROLOG を用いて, 本文で論じた論証ジレンマの場合を含む 3 人の社会でのすべての単純ゲームを生成し, 安定性を含むその性質を分析した結果の一部である. コマンドプロンプト `?-` からのゴール入力で用いられている PROLOG の述語を簡単に説明すると以下のようなものである.

```
gen_win/1 現在のモデル (代替案とエージェント集合) の下で可能な単純ゲームを生成する
verify_win/1 現在のワークスペース上の単純ゲームモデルの諸性質を分析し表示する
is_unanimity_based_preference/4 結託内部の合意に基づく社会的選好を計算する
r_0/5, r_0(K,R,_,[_ ,q-trans,_,_]) 生成した選好関係, その準推移性を満たすもの
```

- なお命題 4 の証明で用いた循環的依存性 (価値制限と共に準推移的な社会的選好の必要十分条件をなす) は, 厳密には固有単純ゲームについて述べられたものである. ゲームが固有 (proper) であるというのは勝利提携の補集合は勝利提携になれないという性質である. そこで以下では, 準推移的でない選好を生じる選好プロフィールをもつかどうかによって固有単純ゲームをそれぞれ生成してみた.

```
%Some proper games have no quasi-transitive orderings.
% N=[1,2,3] the society of taro-hanako-jiro case.
/*
?- gen_win(W),verify_win([yes,yes,S,We|_]),
  \Y+ (is_unanimity_based_preference(K,R,RN,Y),
  (Y\Y=true ;\Y+ r_0(K,R,_,[_ ,q-trans,_,_])),
  nl,write(W), tab(1), write(S;We),fail.
```

```
[[1], [1, 2], [1, 3], [1, 2, 3]] yes;yes([1])
[[2], [1, 2], [2, 3], [1, 2, 3]] yes;yes([2])
[[3], [1, 3], [2, 3], [1, 2, 3]] yes;yes([3])
[[1, 2], [1, 2, 3]] no([[2, 3], [1]]);yes([1, 2])
[[1, 3], [1, 2, 3]] no([[2, 3], [1]]);yes([1, 3])
[[2, 3], [1, 2, 3]] no([[1, 3], [2]]);yes([2, 3])
[[1, 2, 3]] no([[2, 3], [1]]);yes([1, 2, 3])
```

固有ゲームで社会的
意思決定関数が準推
移性に違反しないの
は, 独裁 (上 3 パタン)
か寡頭制および全会
一致 (下 4 パタン)

```
No
?- gen_win(W),verify_win([yes,yes,S,We|_]),
  \Y+ \Y+ (is_unanimity_based_preference(K,R,RN,Y),
  \Y+ r_0(K,R,_,[_ ,q-trans,_,_]),nl,write(S), tab(1), write(We;W),fail.
```

```
[[1, 2], [1, 3], [2, 3], [1, 2, 3]] yes;no
[[1, 2], [1, 3], [1, 2, 3]] no([[2, 3], [1]]);yes([1])
[[1, 2], [2, 3], [1, 2, 3]] no([[1, 3], [2]]);yes([2])
[[1, 3], [2, 3], [1, 2, 3]] no([[1, 2], [3]]);yes([3])
```

多数決では準推
移性へ違反する
(強循環)

```
No
?-
*/
```

- さらに強ゲーム (つまり敗北提携の補集合は必ず勝利提携である) とそれ以外に分けて生成検査すると以下のものであった.

```
/*
?- gen_win(W),verify_win([yes,yes,yes|I]),nl,verify_win,
  verify_eff(S,EF),nl,write('core stability':S),nl,write(ef:EF),fail.
```

```
game:[[1, 2, 3], [1, 3], [1, 2], [1]]
is monotonic
is proper
is strong
is weak with veto players:[1]
is dictatorial
core stability:[true, fail]
```

```
eff:[true, true, fail, true, true, true]
ベ-シ 2/2
```

```
game:[[1, 2, 3], [2, 3], [1, 2], [2]]
is monotonic
is proper
is strong
is weak with veto players:[2]
is dictatorial
core stability:[true, fail]
eff:[true, true, fail, true, true]
```

```
game:[[1, 2, 3], [2, 3], [1, 3], [3]]
is monotonic
is proper
is strong
is weak with veto players:[3]
is dictatorial
core stability:[true, fail]
eff:[true, true, fail, true, true]
```

```
game:[[1, 2, 3], [2, 3], [1, 3], [1, 2]]
is monotonic
is proper
is strong
is not weak
is inessential but nondictatorial
core stability:[fail, fail]
eff:[true, true, fail, true, true]
```

単純多数決原理
は不安定

```
No
?-
*/

% proper but not strong games
/*
?- gen_win(W),verify_win([yes,yes,no(T)|I]),nl,verify_win,
nl,write('is not strong':T),
verify_eff(S,EF),nl,write('core stability':S),nl,write(eff:EF),fail.
```

```
game:[[1, 2, 3], [1, 3], [1, 2]]
is monotonic
is proper
is weak with veto players:[1]
is inessential but nondictatorial
is not strong: ([2, 3], [1])
core stability:[true, fail]
eff:[true, true, fail, true, fail]
```

花子と次郎の結託を無効化
したものは安定(しかし強安
定ではない)

```
game:[[1, 2, 3], [2, 3], [1, 2]]
is monotonic
is proper
is weak with veto players:[2]
```

```
is inessential but nondictatorial
is not strong: ([1, 3], [2])
core stability:[true, fail]
eff:[true, true, fail, true, fail, fail]
```

```
game:[[1, 2, 3], [1, 2]]
is monotonic
is proper
is weak with veto players:[1, 2]
is inessential but nondictatorial
is not strong: ([2, 3], [1])
core stability:[true, fail]
eff:[true, true, fail, true, fail, true]
```

```
game:[[1, 2, 3], [2, 3], [1, 3]]
is monotonic
is proper
is weak with veto players:[3]
is inessential but nondictatorial
is not strong: ([1, 2], [3])
core stability:[true, fail]
eff:[true, true, fail, true, fail, fail]
```

```
game:[[1, 2, 3], [1, 3]]
is monotonic
is proper
is weak with veto players:[1, 3]
is inessential but nondictatorial
is not strong: ([2, 3], [1])
core stability:[true, fail]
eff:[true, true, fail, true, fail, true]
```

```
game:[[1, 2, 3], [2, 3]]
is monotonic
is proper
is weak with veto players:[2, 3]
is inessential but nondictatorial
is not strong: ([1, 3], [2])
core stability:[true, fail]
eff:[true, true, fail, true, fail, true]
```

```
game:[[1, 2, 3]]
is monotonic
is proper
is weak with veto players:[1, 2, 3]
is inessential but nondictatorial
is not strong: ([2, 3], [1])
core stability:[true, fail]
eff:[true, true, fail, true, fail, true]
```

```
No
?-
*/
```