# Sequential Voting Process and Decomposable Voting Rules

Lirong Xia[1] Jérôme Lang[2] Mingsheng Ying[1]

[1]Department of Computer Science
Tsinghua University
Beijing, China

[2]IRIT
Université Paul Sabatier
Toulouse, France

May 31, 2007, at Multiagent group, Stanford University

# Outline

# Outline

Social choice study the design and evaluation of collective decision making from the preferences of agents.

- *agents*: A set of agents $\mathcal{A}$.
- *preference*: Given a finite set of candidates $\mathcal{X}$, $\mathcal{D}$ is the preference structure (linear order, subset, etc).
- *collective decision*: A function from $\mathcal{D}^{\mathcal{A}}$ to specific range:
  - **Aggregation function**: $h : \mathcal{D}^{\mathcal{A}} \to \mathcal{D}$.
  - **Social choice function**: $r : \mathcal{D}^{\mathcal{A}} \to \mathcal{X}$.
  - **Correspondence**: $c : \mathcal{D}^{\mathcal{A}} \to 2^{\mathcal{X}} \setminus \emptyset$.

Social choice study the design and evaluation of collective decision making from the preferences of agents.

- *agents*: A set of agents $\mathcal{A}$.
- *preference*: Given a finite set of candidates $\mathcal{X}$, $\mathcal{D}$ is the preference structure (linear order, subset, etc).
- *collective decision*: A function from $\mathcal{D}^{\mathcal{A}}$ to specific range:
  - **Aggregation function**: $h : \mathcal{D}^{\mathcal{A}} \to \mathcal{D}$.
  - **Social choice function**: $r : \mathcal{D}^{\mathcal{A}} \to \mathcal{X}$.
  - **Correspondence**: $c : \mathcal{D}^{\mathcal{A}} \to 2^{\mathcal{X}} \setminus \emptyset$.

Social choice study the design and evaluation of collective decision making from the preferences of agents.

- *agents*: A set of agents $\mathcal{A}$.
- *preference*: Given a finite set of candidates $\mathcal{X}$, $\mathcal{D}$ is the preference structure (linear order, subset, etc).
- *collective decision*: A function from $\mathcal{D}^{\mathcal{A}}$ to specific range:
  - **Aggregation function**: $h : \mathcal{D}^{\mathcal{A}} \to \mathcal{D}$.
  - **Social choice function**: $r : \mathcal{D}^{\mathcal{A}} \to \mathcal{X}$.
  - **Correspondence**: $c : \mathcal{D}^{\mathcal{A}} \to 2^{\mathcal{X}} \backslash \emptyset$.

Social choice study the design and evaluation of collective decision making of agents (cont.)

- design: A lot of voting rules, *Majority, Borda, Kemeny, etc*.
- evaluation: Several common voting criteria, *neutrality, efficiency, consistency, etc*.

**Arrow's impossibility theorem**: Any aggregation function defined on all profiles and satisfying *efficiency* and *independence from irrelevant alternatives* is dictatorial.

Social choice study the design and evaluation of collective decision making of agents (cont.)

- design: A lot of voting rules, *Majority, Borda, Kemeny, etc*.
- evaluation: Several common voting criteria, *neutrality, efficiency, consistency, etc*.

**Arrow's impossibility theorem**: Any aggregation function defined on all profiles and satisfying *efficiency* and *independence from irrelevant alternatives* is dictatorial.

BUT, no computational considerations

- When the number of candidates increases, the preference structure $\mathcal{D}$ increases exponentially.
- In case of linear order, $|\mathcal{D}| = |\mathcal{X}|!$.
- Running time ...

# Outline

Example: choosing the menu

A group of people have to agree on a common menu composed of a staple, a main dish, and drink. The staple can be rice or noodle; the main dish can be chicken or beef; and the drink can be pure water or coke. How do they proceed?

Combinatorial domain

- Three issues:
  - $D_s = \{$ rice, noodle $\}$
  - $D_m = \{$ chicken, beef $\}$
  - $D_d = \{$ pure water, coke $\}$.
- $\mathcal{X} = D_s \times D_m \times D_d$.

How to model preference structure?

- Linear preference over $\mathcal{X}$?
- Conditional preferences might be a good idea.

How to model preference structure?

- Linear preference over $\mathcal{X}$?
- Conditional preferences might be a good idea.

How to model preference structure?

- Linear preference over $\mathcal{X}$? BAD IDEA
- Conditional preferences might be a good idea.

How to model preference structure?

- Linear preference over $\mathcal{X}$? BAD IDEA
- Conditional preferences might be a good idea.

- "I prefer rice to noodle, and when the staple is rice, I would like chicken, after having the chicken, I want a cup of water."
  - $\mathbf{x}_s = \text{rice} \succ \mathbf{x}_s = \text{noodle}$
  - $\mathbf{x}_s = \text{rice} : \mathbf{x}_m = \text{chicken} \succ \mathbf{x}_m = \text{beef}$
  - $\mathbf{x}_m = \text{chicken} : \mathbf{x}_d = \text{pure water} \succ \mathbf{x}_d = \text{coke}$

- “I prefer rice to noodle, and when the staple is rice, I would like chicken, after having the chicken, I want a cup of water."
  - $\mathbf{x}_s = \text{rice} \succ \mathbf{x}_s = \text{noodle}$
  - $\mathbf{x}_s = \text{rice} : \mathbf{x}_m = \text{chicken} \succ \mathbf{x}_m = \text{beef}$
  - $\mathbf{x}_m = \text{chicken} : \mathbf{x}_d = \text{pure water} \succ \mathbf{x}_d = \text{coke}$

- “I prefer rice to noodle, and when the staple is rice, I would like chicken, after having the chicken, I want a cup of water."
  - $\mathbf{x}_s = \text{rice} \succ \mathbf{x}_s = \text{noodle}$
  - $\mathbf{x}_s = \text{rice} : \mathbf{x}_m = \text{chicken} \succ \mathbf{x}_m = \text{beef}$
  - $\mathbf{x}_m = \text{chicken} : \mathbf{x}_d = \text{pure water} \succ \mathbf{x}_d = \text{coke}$

- “I prefer rice to noodle, and when the staple is rice, I would like chicken, after having the chicken, I want a cup of water."
  - $\mathbf{x}_s = \text{rice} \succ \mathbf{x}_s = \text{noodle}$
  - $\mathbf{x}_s = \text{rice} : \mathbf{x}_m = \text{chicken} \succ \mathbf{x}_m = \text{beef}$
  - $\mathbf{x}_m = \text{chicken} : \mathbf{x}_d = \text{pure water} \succ \mathbf{x}_d = \text{coke}$

CP-net $\mathcal{N}$ [C. Boutilier *el al* 04]

- A set of variables $\mathcal{I} = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$, taking values in $D_1, \ldots, D_p$
- A directed acyclic graph $G = (\mathcal{I}, E)$.
- A CPT for each $\mathbf{x}_i$ indicating the conditional preference on $D_i$.

If for all $i$, $|D_i|$ is limited by a constant, then $\mathcal{D}$ is in linear of $|\mathcal{X}|$.

CP-net $\mathcal{N}$ [C. Boutilier *el al* 04]

- A set of variables $\mathcal{I} = \{\mathbf{x}_1, \dots, \mathbf{x}_p\}$, taking values in $D_1, \dots, D_p$
- A directed acyclic graph $G = (\mathcal{I}, E)$.
- A CPT for each $\mathbf{x}_i$ indicating the conditional preference on $D_i$.

If for all $i$, $|D_i|$ is limited by a constant, then $\mathcal{D}$ is in linear of $|\mathcal{X}|$.

## Linear order extending $\mathcal{N}$

- A linear order $V$ on $\mathcal{X} = D_1 \times \ldots \times D_p$ is said to extend $\mathcal{N}$, if for all $\mathbf{x}_i$, the projection of $V$ on $\mathbf{x}_i$ is the same as the entry in $\mathcal{N}$.
- $V$ is compatible with a linear order $\mathcal{O}$ on $\mathcal{I}$ if the graph of its CP-net is compatible with $\mathcal{O}$.
- Notice $V$ is not unique.

## Linear order extending $\mathcal{N}$

- A linear order $V$ on $\mathcal{X} = D_1 \times \ldots \times D_p$ is said to extend $\mathcal{N}$, if for all $\mathbf{x}_i$, the projection of $V$ on $\mathbf{x}_i$ is the same as the entry in $\mathcal{N}$.
- $V$ is compatible with a linear order $\mathcal{O}$ on $\mathcal{I}$ if the graph of its CP-net is compatible with $\mathcal{O}$.
- Notice $V$ is not unique.

## Linear order extending $\mathcal{N}$

- A linear order $V$ on $\mathcal{X} = D_1 \times \ldots \times D_p$ is said to extend $\mathcal{N}$, if for all $\mathbf{x}_i$, the projection of $V$ on $\mathbf{x}_i$ is the same as the entry in $\mathcal{N}$.
- $V$ is compatible with a linear order $\mathcal{O}$ on $\mathcal{I}$ if the graph of its CP-net is compatible with $\mathcal{O}$.
- Notice $V$ is not unique.

# Outline

1. Introduction
   - Basic settings
   - Compact representation
   - Sequential voting process
2. Local rules vs. Global rule
3. Nearly neutral
4. Order independent sequential composition

Voting on combinatorial domain with preferences modeled by CP-nets [J.Lang 07]

- A set of issues $\mathcal{I} = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$ on $\mathcal{X} = D_1 \times \ldots \times D_p$.
- A linear order $\mathcal{O}$ over $\mathcal{I}$, for example $\mathcal{O} = \mathbf{x}_1 > \ldots > \mathbf{x}_p$.
- $p$ local voting rules $r_1, \ldots, r_p$.
- Input: A profile $P = (V_1, \ldots, V_N)$ s.t. $V_j$ is compatible with $\mathcal{O}$.

Output: $(d_1, \ldots, d_p) \in \mathcal{X}$ through a $p$-step process

1. Select $d_1$ by $r_1$ from $P|_{\mathbf{x}_1}$.
2. Select $d_2$ by $r_2$ from $P|_{\mathbf{x}_2|\mathbf{x}_1=d_1}$.

   $\vdots$

p. Select $d_p$ by $r_p$ from $P|_{\mathbf{x}_p|\mathbf{x}_1=d_1,\ldots,\mathbf{x}_{p-1}=d_{p-1}}$.

Such a rule is defined to be the *sequential composition* of $r_1, \ldots, r_p$, denoted by $Seq(r_1, \ldots, r_p)$. It is said to be *decomposable*.

Output: $(d_1, \ldots, d_p) \in \mathcal{X}$ through a $p$-step process

1. Select $d_1$ by $r_1$ from $P|_{\mathbf{x}_1}$.
2. Select $d_2$ by $r_2$ from $P|_{\mathbf{x}_2|\mathbf{x}_1=d_1}$.

⋮

p. Select $d_p$ by $r_p$ from $P|_{\mathbf{x}_p|\mathbf{x}_1=d_1,\ldots,\mathbf{x}_{p-1}=d_{p-1}}$.

Such a rule is defined to be the *sequential composition* of $r_1, \ldots, r_p$, denoted by $Seq(r_1, \ldots, r_p)$. It is said to be *decomposable*.

Output: $(d_1, \ldots, d_p) \in \mathcal{X}$ through a $p$-step process

1. Select $d_1$ by $r_1$ from $P|_{\mathbf{x}_1}$.
2. Select $d_2$ by $r_2$ from $P|_{\mathbf{x}_2|\mathbf{x}_1=d_1}$.

   ⋮

p. Select $d_p$ by $r_p$ from $P|_{\mathbf{x}_p|\mathbf{x}_1=d_1,\ldots,\mathbf{x}_{p-1}=d_{p-1}}$.

Such a rule is defined to be the *sequential composition* of $r_1, \ldots, r_p$, denoted by $Seq(r_1, \ldots, r_p)$. It is said to be *decomposable*.

Output: $(d_1, \ldots, d_p) \in \mathcal{X}$ through a $p$-step process

1. Select $d_1$ by $r_1$ from $P|_{\mathbf{x}_1}$.
2. Select $d_2$ by $r_2$ from $P|_{\mathbf{x}_2|\mathbf{x}_1=d_1}$.

   ⋮

p. Select $d_p$ by $r_p$ from $P|_{\mathbf{x}_p|\mathbf{x}_1=d_1,\ldots,\mathbf{x}_{p-1}=d_{p-1}}$.

Such a rule is defined to be the *sequential composition* of $r_1, \ldots, r_p$, denoted by $Seq(r_1, \ldots, r_p)$. It is said to be *decomposable*.

Example: R=Rice, N=Noodle, C=Chicken, B=Beef, W=water, K=Coke. $r_1, r_2, r_2$ are all Majority rules. $\mathcal{O} = \mathbf{x}_s > \mathbf{x}_m > \mathbf{x}_d$.

| issue | voter1 | voter2 | voter3 |
|---|---|---|---|
| $\mathbf{x}_s$ | $R \succ N$ | $R \succ N$ | $N \succ R$ |
| $\mathbf{x}_m$ | $R : C \succ B$ | $R : B \succ C$ | $R : C \succ B$ |
| $\mathbf{x}_m$ | $N : C \succ B$ | $N : C \succ B$ | $N : C \succ B$ |
| $\mathbf{x}_d$ | $C : W \succ K$ | $C : W \succ K$ | $C : K \succ W$ |
| $\mathbf{x}_d$ | $B : K \succ W$ | $B : W \succ K$ | $B : K \succ W$ |

$\mathbf{x}_s$: $r_1(R \succ N, R \succ N, N \succ R) = R$,
$\mathbf{x}_m$: $r_2(C \succ B, B \succ C, C \succ B) = C$,
$\mathbf{x}_d$: $r_3(W \succ K, W \succ K, K \succ W) = W$,
$Seq(r_1, r_2, r_3)(V_1, V_2, V_3) = (R, C, W)$.

Example: R=Rice, N=Noodle, C=Chicken, B=Beef, W=water, K=Coke. $r_1, r_2, r_2$ are all Majority rules. $\mathcal{O} = \mathbf{x}_s > \mathbf{x}_m > \mathbf{x}_d$.

| issue | voter1 | voter2 | voter3 |
|---|---|---|---|
| $\mathbf{x}_s$ | $R \succ N$ | $R \succ N$ | $N \succ R$ |
| $\mathbf{x}_m$ | $R : C \succ B$ | $R : B \succ C$ | $R : C \succ B$ |
| $\mathbf{x}_m$ | $N : C \succ B$ | $N : C \succ B$ | $N : C \succ B$ |
| $\mathbf{x}_d$ | $C : W \succ K$ | $C : W \succ K$ | $C : K \succ W$ |
| $\mathbf{x}_d$ | $B : K \succ W$ | $B : W \succ K$ | $B : K \succ W$ |

$\mathbf{x}_s$: $r_1(R \succ N, R \succ N, N \succ R) = R$,

$\mathbf{x}_m$: $r_2(C \succ B, B \succ C, C \succ B) = C$,

$\mathbf{x}_d$: $r_3(W \succ K, W \succ K, K \succ W) = W$,

$Seq(r_1, r_2, r_3)(V_1, V_2, V_3) = (R, C, W)$.

Example: R=Rice, N=Noodle, C=Chicken, B=Beef, W=water, K=Coke. $r_1, r_2, r_2$ are all Majority rules. $\mathcal{O} = \mathbf{x}_s > \mathbf{x}_m > \mathbf{x}_d$.

| issue | voter1 | voter2 | voter3 |
|---|---|---|---|
| $\mathbf{x}_s$ | $R \succ N$ | $R \succ N$ | $N \succ R$ |
| $\mathbf{x}_m$ | $R : C \succ B$ | $R : B \succ C$ | $R : C \succ B$ |
| $\mathbf{x}_m$ | $N : C \succ B$ | $N : C \succ B$ | $N : C \succ B$ |
| $\mathbf{x}_d$ | $C : W \succ K$ | $C : W \succ K$ | $C : K \succ W$ |
| $\mathbf{x}_d$ | $B : K \succ W$ | $B : W \succ K$ | $B : K \succ W$ |

$\mathbf{x}_s$: $r_1(R \succ N, R \succ N, N \succ R) = R$,
$\mathbf{x}_m$: $r_2(C \succ B, B \succ C, C \succ B) = C$,
$\mathbf{x}_d$: $r_3(W \succ K, W \succ K, K \succ W) = W$,
$Seq(r_1, r_2, r_3)(V_1, V_2, V_3) = (R, C, W)$.

Example: R=Rice, N=Noodle, C=Chicken, B=Beef, W=water, K=Coke. $r_1, r_2, r_2$ are all Majority rules. $\mathcal{O} = \mathbf{x}_s > \mathbf{x}_m > \mathbf{x}_d$.

| issue | voter1 | voter2 | voter3 |
|---|---|---|---|
| $\mathbf{x}_s$ | $R \succ N$ | $R \succ N$ | $N \succ R$ |
| $\mathbf{x}_m$ | $R : C \succ B$ | $R : B \succ C$ | $R : C \succ B$ |
| $\mathbf{x}_m$ | $N : C \succ B$ | $N : C \succ B$ | $N : C \succ B$ |
| $\mathbf{x}_d$ | $C : W \succ K$ | $C : W \succ K$ | $C : K \succ W$ |
| $\mathbf{x}_d$ | $B : K \succ W$ | $B : W \succ K$ | $B : K \succ W$ |

$\mathbf{x}_s$: $r_1(R \succ N, R \succ N, N \succ R) = R$,
$\mathbf{x}_m$: $r_2(C \succ B, B \succ C, C \succ B) = C$,
$\mathbf{x}_d$: $r_3(W \succ K, W \succ K, K \succ W) = W$,
$Seq(r_1, r_2, r_3)(V_1, V_2, V_3) = (R, C, W)$.

Example: R=Rice, N=Noodle, C=Chicken, B=Beef, W=water, K=Coke. $r_1, r_2, r_2$ are all Majority rules. $\mathcal{O} = \mathbf{x}_s > \mathbf{x}_m > \mathbf{x}_d$.

| issue | voter1 | voter2 | voter3 |
|---|---|---|---|
| $\mathbf{x}_s$ | $R \succ N$ | $R \succ N$ | $N \succ R$ |
| $\mathbf{x}_m$ | $R : C \succ B$ | $R : B \succ C$ | $R : C \succ B$ |
| $\mathbf{x}_m$ | $N : C \succ B$ | $N : C \succ B$ | $N : C \succ B$ |
| $\mathbf{x}_d$ | $C : W \succ K$ | $C : W \succ K$ | $C : K \succ W$ |
| $\mathbf{x}_d$ | $B : K \succ W$ | $B : W \succ K$ | $B : K \succ W$ |

$\mathbf{x}_s$: $r_1(R \succ N, R \succ N, N \succ R) = R$,
$\mathbf{x}_m$: $r_2(C \succ B, B \succ C, C \succ B) = C$,
$\mathbf{x}_d$: $r_3(W \succ K, W \succ K, K \succ W) = W$,
$Seq(r_1, r_2, r_3)(V_1, V_2, V_3) = (R, C, W)$.

A special case: seat-by-seat voting

- Each vote corresponds to a CP-net without edge.
- $d_i$ is selected from $P|_{\mathbf{x}_i}$ independently.
- Shortcoming: too demanding.

A special case: seat-by-seat voting

- Each vote corresponds to a CP-net without edge.
- $d_i$ is selected from $P|_{\mathbf{x}_i}$ independently.
- Shortcoming: too demanding.

A special case: seat-by-seat voting

- Each vote corresponds to a CP-net without edge.
- $d_i$ is selected from $P|_{\mathbf{x}_i}$ independently.
- Shortcoming: too demanding.

## Outline

Question

What can we learn about $r_1, \ldots, r_p$ from $Seq(r_1, \ldots, r_p)$? And vice versa.

# Outline

Common voting criteria for a rule $r$, $P$ is a profile

- **Neutrality**: The voting rule is indifferent with permutation between candidates.
- **Monotonicity**: Raising the winner $w$ of a profile $P$ simultaneously in all votes, the new profile will also selects $w$.
- **Strong Monotonicity**: Rising a set $Y$ of candidates simultaneously in all votes, the winner of new profile will be in $\{w\} \cup Y$.
- **Consistency**: If $r(P_1) = r(P_2)$, then $r(P_1 \cup P_2) = r(P_1)$.
- **Efficiency**: There is no candidate that is preferred to $r(P)$ in all votes of $P$.

# Outline

Criteria that can inherit from $Seq(r_1, \ldots, r_p)$ to local rules

## Theorem

*Let* `Prop` $\in$ *{anonymity, homogeneity, neutrality, strong monotonicity, consistency, participation, efficiency}. If* $Seq(r_1, \ldots, r_p)$ *satisfies* `Prop` *then for any* $1 \leq i \leq p$, $r_i$ *also satisfies* `Prop`*.*

## Theorem

*If* $Seq(r_1, \ldots, r_p)$ *satisfies monotonicity, then* $r_p$ *satisfies monotonicity.*

## Outline

Criteria that can be lifted from local rules to their sequential composition , positive results.

### Theorem

*Let* $\texttt{Prop} \in$ *{anonymity, homogeneity, consistency, strong monotonicity}. If for any* $1 \leq i \leq p$, $r_i$ *also satisfies* $\texttt{Prop}$ *then* $Seq(r_1, \ldots, r_p)$ *satisfies* $\texttt{Prop}$.

### Theorem

*If* $r_p$ *satisfies monotonicity, then* $Seq(r_1, \ldots, r_p)$ *satisfies monotonicity.*

## Negative results

### Theorem

*Let* $\texttt{Prop} \in$ *{neutrality, participation, efficiency}. There exists* $r_i$, $i \leq p$*, satisfying* $\texttt{Prop}$ *but* $Seq(r_1, \ldots, r_p)$ *does not.*

## Summary

| Criteria | Global to local | Local to global |
|---|---|---|
| *Anonymity* | Y | Y |
| *Homogeneity* | Y | Y |
| *Neutrality* | Y | N |
| *Monotonicity* | Only $r_p$ | Only $r_p$ |
| *Consistency* | Y | Y |
| *Participation* | Y | N |
| *Consensus* | Y | N |
| *Strong monotonicity* | Y | Y |

# Outline

J.-P. Benoît and L.A. Kornhauser proved an impossibility theorem about neutrality and efficiency of seat-by-seat rules.

## Theorem

*If at least one of the following two conditions holds*

1. $|p \geq 3|$, *or*
2. $p = 2$, *and* $|D_1| \geq 3$ *or* $|D_2| \geq 3$.

*then the only* **efficient** *seat-by-seat voting rule is the dictatorship.*

## Theorem

*Suppose* $p \geq 3$, $r_i$ *is efficient for all* $i \leq p$, *then the only* **neutral** *seat-by-seat rule is a dictatorship.*

Two binary issues, we proved

### Theorem

*The sequential majority rule over two binary issues satisfies* **neutrality** *and* **efficiency***.*

We proved a stronger form of their theorem

## Theorem

*If at least one of the following two conditions holds*

1. $|p \geq 3|$, *or*
2. $p = 2$, *and* $|D_1| \geq 3$ *or* $|D_2| \geq 3$.

*then the only* **efficient** *or* **neutral** *seat-by-seat voting rule or correspondence is the dictatorship or always chooses every candidate.*

Improvements

- Removed the local efficiency condition.
- Proved two issues case for neutrality.
- Extend to seat-by-seat correspondences.

Therefore the existence of neutrality or efficiency of sequential rule (correspondence) is completely solved.

## Outline

Slightly relax the domain constraint of properties.

For example, neutrality means for all $P$ compatible with $\mathcal{O}$, and for all permutation $M$, if $M(P)$ is compatible with $\mathcal{O}$, then

$$M(Seq(r_1, \ldots, r_p)(P)) = Seq(r_1, \ldots, r_p)M(P).$$

Replace the universal quantifier by a weaker one.

- by for all profile in a subset $U$ of all “legal" profiles.
- Such $U$ should be “large" enough.

Replace the universal quantifier by a weaker one.

- by for all profile in a subset $U$ of all “legal" profiles.
- Such $U$ should be “large" enough.

Notation

- $Legal(\mathcal{O})$: All single votes compatible with the order $\mathcal{O}$.
- $Legal(\mathcal{O})^i$: The set of all $i$-voter profiles compatible with $\mathcal{O}$.
- $Seq(r_1, \ldots, r_p)$ is defined over $Legal(\mathcal{O}) \cup Legal(\mathcal{O})^2 \cup \ldots$.

## Definition

We say $U = U_1 \cup U_2 \cup \ldots$ is *nearly representative* for $Legal(\mathcal{O})$, if

- $\forall i, U_i \subseteq Legal(\mathcal{O})^i$.
- $\lim_{i \to \infty} \frac{|U_i|}{|Legal(\mathcal{O})|^i} = 1$.

We say $U = U_1 \cup U_2 \cup \ldots$ is *nearly representative* for $Legal(\mathcal{O})$, if

- $\forall i, U_i \subseteq Legal(\mathcal{O})^i$.
- $\lim_{i\to\infty} \frac{|U_i|}{|Legal(\mathcal{O})|^i} = 1$.

## Definition

We say $U = U_1 \cup U_2 \cup \ldots$ is *nearly representative* for $Legal(\mathcal{O})$, if

- $\forall i, U_i \subseteq Legal(\mathcal{O})^i$.
- $\lim_{i\to\infty} \dfrac{|U_i|}{|Legal(\mathcal{O})|^i} = 1$.

$Seq(r_1, \ldots, r_p)$ is *nearly neutral*, if there exists $U$ nearly representative for $Legal(\mathcal{O})$ s.t. for all profile $P \in U$, and for all permutation $M$, if $M(P) \in U$, then

$$M(Seq(r_1, \ldots, r_p)(P)) = Seq(r_1, \ldots, r_p)M(P).$$

**Theorem**

*If local rules are neutral, then their sequential composition is nearly neutral.*

- Take $U$={All profiles that cover all CP-nets compatible with $\mathcal{O}$}.

# Outline

- In the sequential voting process, an voting order $\mathcal{O}$ is presumed. What if the order is not fixed?
- For example, this week we vote by the order Staple>Main dish> Drink, next week we may vote by the order Drink>Staple>Main dish.

Our solution: make union of all orders

Define it to be the sequential composition of $\{r_1, \ldots, r_p\}$ w.r.t. the order that $P$ is compatible with.

$$Seq^{OI}(r_1, r_2)(P) = \begin{cases} Seq(r_1, r_2)(P) & \text{If P is compatible with } \mathbf{x}_1 > \mathbf{x}_2 \\ Seq(r_2, r_1)(P) & \text{If P is compatible with } \mathbf{x}_2 > \mathbf{x}_1 \end{cases}$$

# Outline

Order-independent sequential composition $Seq^{OI}(r_1, \ldots, r_p)$

- The domain is $\bigcup_{\mathcal{O}, i \in \mathbb{N}} Legal(\mathcal{O})^i$
- For any $P \in Legal(\mathcal{O})^i$, $\mathcal{O} = \mathbf{x}_{i_1} > \ldots > \mathbf{x}_{i_p}$

$$Seq^{OI}(r_1, \ldots, r_p)(P) = Seq(r_{i_1}, \ldots, r_{i_p})(P).$$

It is well-defined.

Order-independent sequential composition $\mathit{Seq}^{OI}(r_1, \ldots, r_p)$

- The domain is $\bigcup_{\mathcal{O}, i \in \mathbb{N}} \mathit{Legal}(\mathcal{O})^i$
- For any $P \in \mathit{Legal}(\mathcal{O})^i$, $\mathcal{O} = \mathbf{x}_{i_1} > \ldots > \mathbf{x}_{i_p}$

$$\mathit{Seq}^{OI}(r_1, \ldots, r_p)(P) = \mathit{Seq}(r_{i_1}, \ldots, r_{i_p})(P).$$

It is well-defined.

# Outline

## Local vs. Global

| Criteria | Global to local | Local to global |
|---|---|---|
| *Anonymity* | Y | Y |
| *Homogeneity* | Y | Y |
| *Neutrality* | Y | N |
| *Monotonicity* | Y | Y |
| *Consistency* | Y | Y |
| *Participation* | Y | N |
| *Efficiency* | Y | N |

“Nearly" property.

**Theorem**

*If local rules are neutral, and $|D_i| = |D_j| \Rightarrow r_i = r_j$, then their order-independent sequential composition is nearly neutral.*

Summary: For sequential voting process, we did

- Local vs. global in terms of common voting criteria
- Extend the impossibility theorems of J.-P. Benoît and L.A. Kornhauser.
- “nearly" neutrality for sequential voting rule.
- Order-independent sequential composition and parallel works.

Future work

- Determination problem.
- Computational issues.
- Other forms of compact representation (TCP-net).

## Further readings


- C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole. CP-nets: a tool for representing and reasoning with conditional *ceteris paribus* statements. JAIR. 21:135–191, 2004.
- J.-P. Benoit and L.A. Kornhauser. Only a dictatorship is efficient or neutral. Technical report, 2006.
- J. Lang. Voting and aggregation on combinatorial domains with structured preferences. In *IJCAI'07*, pages 1366–1371, 2007.
- L. Xia, J. Lang, and M. Ying. Sequential voting and multiple election paradoxes. In *TARK'07*, 2007.
- L. Xia, J. Lang, and M. Ying. Strongly Decomposable Voting Rules on Multiattribute Domains. In *AAAI'07*, 2007.
- L. Xia. On the neutrality and efficiency of decomposable voting correspondences. *Working draft*.