MATH 456, 2023
Mathematical Modeling

Mark C. Wilson

April 20, 2023

What is a mathematical model?

- An abstraction of a real-world problem using mathematical objects.

What is a mathematical model?

- An abstraction of a real-world problem using mathematical objects.
- They are everywhere - even basic counting and arithmetic use modeling.

What is a mathematical model?

- An abstraction of a real-world problem using mathematical objects.
- They are everywhere - even basic counting and arithmetic use modeling.
- Physics has historically been a great source of models, but they are now used in all areas of science and in other disciplines.

Why use a mathematical model?

- Cheaper than building a physical one, or manipulating a real system.

Why use a mathematical model?

- Cheaper than building a physical one, or manipulating a real system.
- Ethical concerns about manipulating a real system.

Why use a mathematical model?

- Cheaper than building a physical one, or manipulating a real system.
- Ethical concerns about manipulating a real system.
- Parameters may mean there are infinitely many possible models, and models allow for optimization.

Why use a mathematical model?

- Cheaper than building a physical one, or manipulating a real system.
- Ethical concerns about manipulating a real system.
- Parameters may mean there are infinitely many possible models, and models allow for optimization.
- Generate experimental predictions that can be tested.

What makes a good mathematical model?

- Grounded in principles of application area (axioms, laws).

What makes a good mathematical model?

- Grounded in principles of application area (axioms, laws).
- Understandable and fairly simple (cf neural nets, Einstein).

What makes a good mathematical model?

- Grounded in principles of application area (axioms, laws).
- Understandable and fairly simple (cf neural nets, Einstein).
- Precise enough to make clear predictions.

What makes a good mathematical model?

- Grounded in principles of application area (axioms, laws).
- Understandable and fairly simple (cf neural nets, Einstein).
- Precise enough to make clear predictions.
- Robust: doesn't change behavior dramatically when parameter values change slightly.

Overview

- We will concentrate on social science applications related to collective decision-making.

Overview

- We will concentrate on social science applications related to collective decision-making.
- Voting (public goods) and consensus.

Overview

- We will concentrate on social science applications related to collective decision-making.
- Voting (public goods) and consensus.
- Resource allocation (private goods).

Overview

- We will concentrate on social science applications related to collective decision-making.
- Voting (public goods) and consensus.
- Resource allocation (private goods).
- Preferences and how they may change.

Overview

The intuitive concepts “randomness” and “chance” have been formalized in probability theory, which took several centuries to evolve to its present state. We need to be fluent in the basic language of probability.

Probability

- A discrete sample space is simply a finite set Ω, whose elements are called elementary events. A subset of Ω is called an event. We can also allow infinite sets such as $\mathbb{N}$, but not $\mathbb{R}$.

Probability

- A discrete sample space is simply a finite set Ω, whose elements are called elementary events. A subset of Ω is called an event. We can also allow infinite sets such as $\mathbb{N}$, but not $\mathbb{R}$.
- A probability measure on Ω is a function $P : \Omega \to \mathbb{R}$ such that $0 \leq P(x) \leq 1$ and $\sum_{x \in \Omega} P(x) = 1$. The pair (Ω, P) gives us a probability space.

Probability

- A discrete sample space is simply a finite set Ω, whose elements are called elementary events. A subset of Ω is called an event. We can also allow infinite sets such as $\mathbb{N}$, but not $\mathbb{R}$.
- A probability measure on Ω is a function $P : \Omega \to \mathbb{R}$ such that $0 \leq P(x) \leq 1$ and $\sum_{x \in \Omega} P(x) = 1$. The pair (Ω, P) gives us a probability space.
- We can then define, for each subset A of Ω, $P(A) = \sum_{x \in A} P(x)$, so that $0 \leq P(A) \leq 1$.

Probability

- A discrete sample space is simply a finite set Ω, whose elements are called elementary events. A subset of Ω is called an event. We can also allow infinite sets such as $\mathbb{N}$, but not $\mathbb{R}$.
- A probability measure on Ω is a function $P : \Omega \to \mathbb{R}$ such that $0 \leq P(x) \leq 1$ and $\sum_{x \in \Omega} P(x) = 1$. The pair (Ω, P) gives us a probability space.
- We can then define, for each subset A of Ω, $P(A) = \sum_{x \in A} P(x)$, so that $0 \leq P(A) \leq 1$.
- Two events A, B are independent if $P(A \cap B) = P(A)P(B)$.

Probability

- A discrete sample space is simply a finite set Ω, whose elements are called elementary events. A subset of Ω is called an event. We can also allow infinite sets such as $\mathbb{N}$, but not $\mathbb{R}$.
- A probability measure on Ω is a function $P : \Omega \to \mathbb{R}$ such that $0 \leq P(x) \leq 1$ and $\sum_{x\in\Omega} P(x) = 1$. The pair (Ω, P) gives us a probability space.
- We can then define, for each subset A of Ω, $P(A) = \sum_{x\in A} P(x)$, so that $0 \leq P(A) \leq 1$.
- Two events A, B are independent if $P(A \cap B) = P(A)P(B)$.
- If A and B are disjoint then $P(A \cup B) = P(A) + P(B)$; in general $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

Examples of probability spaces

- $\Omega = \{0, 1, 2, \ldots, n-1\}$ (for some fixed positive integer n), and P is the uniform measure that gives value $1/n$ to each element.

Examples of probability spaces

- $\Omega = \{0, 1, 2, \dots, n-1\}$ (for some fixed positive integer n), and P is the uniform measure that gives value $1/n$ to each element.
- Ω is the set of all permutations of n elements (for some fixed positive integer n), and P is the uniform measure that gives value $1/n!$ to each element.

Examples of probability spaces

- $\Omega = \{0, 1, 2, \ldots, n-1\}$ (for some fixed positive integer n), and P is the uniform measure that gives value $1/n$ to each element.
- Ω is the set of all permutations of n elements (for some fixed positive integer n), and P is the uniform measure that gives value $1/n!$ to each element.
- Ω is the set of all binary strings of length n, and P is the uniform measure that gives value 2^{-n} to each element.

Examples of probability spaces

- $\Omega = \{0, 1, 2, \ldots, n-1\}$ (for some fixed positive integer n), and P is the uniform measure that gives value $1/n$ to each element.
- Ω is the set of all permutations of n elements (for some fixed positive integer n), and P is the uniform measure that gives value $1/n!$ to each element.
- Ω is the set of all binary strings of length n, and P is the uniform measure that gives value 2^{-n} to each element.
- $\Omega = \mathbb{N}$ and P is the measure with $P(n) = 2^{-(n+1)}$ for each $n \in \Omega$.

Random variable

- A random variable on a probability space Ω is a function $X : \Omega \to \mathbb{R}$.

Random variable

- A random variable on a probability space Ω is a function $X : \Omega \to \mathbb{R}$.
- Given an interval $I \subseteq \mathbb{R}$, we write $P(X \in I)$ to mean $P(\{\omega \in \Omega : X(\omega) \in I\})$.

Random variable

- A random variable on a probability space Ω is a function $X : \Omega \to \mathbb{R}$.
- Given an interval $I \subseteq \mathbb{R}$, we write $P(X \in I)$ to mean $P(\{\omega \in \Omega : X(\omega) \in I\})$.
- Special cases:

Random variable

- A random variable on a probability space Ω is a function $X : \Omega \to \mathbb{R}$.
- Given an interval $I \subseteq \mathbb{R}$, we write $P(X \in I)$ to mean $P(\{\omega \in \Omega : X(\omega) \in I\})$.
- Special cases:
  - $I = \{a\}$ (here we write $P(X = a)$)

Random variable

- A random variable on a probability space Ω is a function $X : \Omega \to \mathbb{R}$.
- Given an interval $I \subseteq \mathbb{R}$, we write $P(X \in I)$ to mean $P(\{\omega \in \Omega : X(\omega) \in I\})$.
- Special cases:
    - $I = \{a\}$ (here we write $P(X = a)$)
    - $I = (-\infty, a]$ (here we write $P(X \leq a)$).

Random variable

- A random variable on a probability space Ω is a function $X : \Omega \to \mathbb{R}$.
- Given an interval $I \subseteq \mathbb{R}$, we write $P(X \in I)$ to mean $P(\{\omega \in \Omega : X(\omega) \in I\})$.
- Special cases:
  - $I = \{a\}$ (here we write $P(X = a)$)
  - $I = (-\infty, a]$ (here we write $P(X \leq a)$).
- The probability mass function of X is the function given by $f_X(a) = P(X = a)$. The cumulative distribution function is $F_X(a) = P(X \leq a)$.

Random variable

- A random variable on a probability space Ω is a function $X : \Omega \to \mathbb{R}$.
- Given an interval $I \subseteq \mathbb{R}$, we write $P(X \in I)$ to mean $P(\{\omega \in \Omega : X(\omega) \in I\})$.
- Special cases:
  - $I = \{a\}$ (here we write $P(X = a)$)
  - $I = (-\infty, a]$ (here we write $P(X \leq a)$).
- The probability mass function of X is the function given by $f_X(a) = P(X = a)$. The cumulative distribution function is $F_X(a) = P(X \leq a)$.
- The probability mass function or cumulative distribution function often tell us all we need to know about the behavior of the random variable, and the exact value of Ω and P is not needed.

Famous random variables

- Bernoulli (with parameter p): here $0 \leq p \leq 1$, Ω has two elements a, b, the probability distribution is $P(a) = p, P(b) = 1 - p$, and $X(a) = 0, X(b) = 1$. Hence $f_X(0) = p, f_X(1) = 1 - p$.

Famous random variables

- Bernoulli (with parameter p): here $0 \leq p \leq 1$, Ω has two elements a, b, the probability distribution is $P(a) = p, P(b) = 1 - p$, and $X(a) = 0, X(b) = 1$. Hence $f_X(0) = p, f_X(1) = 1 - p$.
- Binomial (with parameters n, p): here $0 \leq p \leq 1$, n is a positive integer and for each integer k with $0 \leq k \leq n$, $f_X(k) = p^k(1 - p)^{n-k}$.

Famous random variables

- Bernoulli (with parameter p): here $0 \leq p \leq 1$, Ω has two elements a, b, the probability distribution is $P(a) = p, P(b) = 1 - p$, and $X(a) = 0, X(b) = 1$. Hence $f_X(0) = p, f_X(1) = 1 - p$.
- Binomial (with parameters n, p): here $0 \leq p \leq 1$, n is a positive integer and for each integer k with $0 \leq k \leq n$, $f_X(k) = p^k(1 - p)^{n-k}$.
- Geometric (with parameter p): here $0 < p \leq 1$, and for each positive integer k, $f_X(k) = p(1 - p)^{k-1}$.

Famous random variables

- Bernoulli (with parameter p): here $0 \leq p \leq 1$, Ω has two elements a, b, the probability distribution is $P(a) = p, P(b) = 1 - p$, and $X(a) = 0, X(b) = 1$. Hence $f_X(0) = p, f_X(1) = 1 - p$.
- Binomial (with parameters n, p): here $0 \leq p \leq 1$, n is a positive integer and for each integer k with $0 \leq k \leq n$, $f_X(k) = p^k(1-p)^{n-k}$.
- Geometric (with parameter p): here $0 < p \leq 1$, and for each positive integer k, $f_X(k) = p(1-p)^{k-1}$.
- Poisson (with parameter λ): here $\lambda > 0$ and for each $k \in \mathbb{N}$,

$$f_X(k) = e^{-\lambda} \frac{\lambda^k}{k!}.$$

Moments

- The mean or expectation of X is

$$E[X] := \sum_{a} a f_X(a) = \sum_{\omega \in \Omega} X(\omega) P(\omega).$$

Moments

- The mean or expectation of X is

$$E[X] := \sum_{a} a f_X(a) = \sum_{\omega \in \Omega} X(\omega) P(\omega).$$

- Expectation is a linear operator:

$$E[X+Y] = E[X] + E[Y]$$
$$E[\lambda X] = \lambda E[X].$$

Moments

- The mean or expectation of X is

$$E[X] := \sum_{a} a f_X(a) = \sum_{\omega \in \Omega} X(\omega) P(\omega).$$

- Expectation is a linear operator:

$$E[X+Y] = E[X] + E[Y]$$
$$E[\lambda X] = \lambda E[X].$$

- The variance is defined by

$$V[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

and the standard deviation by $\sigma(X) = \sqrt{V[X]}$.

Conditional probability

- If $P(B) > 0$ then we define

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

and call it the conditional probability of A given B.

Conditional probability

- If $P(B) > 0$ then we define

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

and call it the conditional probability of A given B.

- If also $P(A) > 0$ then we have Bayes' rule

$$P(B \mid A) = \frac{P(A \mid B)P(B)}{P(A \mid B)P(B) + P(A \mid \overline{B})P(\overline{B})}$$

Joint distribution and independence

- The *joint mass function* of X and Y is

$$f_{X,Y}(a,b) = P(X = a \text{ and } Y = b) = P(A) \cap P(B)$$

where we define
$A = \{\omega \in \Omega : X(\omega) = a\}, B = \{\omega \in \Omega : Y(\omega) = b\}.$

Joint distribution and independence

- The *joint mass function* of X and Y is

$$f_{X,Y}(a,b) = P(X = a \text{ and } Y = b) = P(A) \cap P(B)$$

  where we define
  $A = \{\omega \in \Omega : X(\omega) = a\}, B = \{\omega \in \Omega : Y(\omega) = b\}$.
- Variables X and Y are independent if
  $f_{X,Y}(a,b) = f_X(a) f_Y(b)$ for all $a, b \in \mathbb{R}$.

Overview

The intuitive concepts of "bigger than" or "better than" require a model of ordered sets. These are simple kinds of (directed) graphs. Graphs allow us to discuss concepts such as "connected to", "related to", distance and other useful ideas. They have enormously many applications.

Partial orders

- A partial order on a set S is a binary relation (a subset R of $S \times S$ that is:

Partial orders

- A partial order on a set S is a binary relation (a subset R of $S \times S$ that is:
  - reflexive ($(a, a) \in R$ for all $a \in S$)

Partial orders

- A partial order on a set S is a binary relation (a subset R of $S \times S$ that is:
  - reflexive ($(a, a) \in R$ for all $a \in S$)
  - transitive (if $(a, b) \in R$ and $(b, c) \in R$ then $(a, c) \in R$).

Partial orders

- A partial order on a set S is a binary relation (a subset R of $S \times S$ that is:
  - reflexive ($(a, a) \in R$ for all $a \in S$)
  - transitive (if $(a, b) \in R$ and $(b, c) \in R$ then $(a, c) \in R$).
  - antisymmetric (if $(a, b) \in R$ and $(b, a) \in R$ then $a = b$).

Partial orders

- A partial order on a set S is a binary relation (a subset R of $S \times S$ that is:
  - reflexive ($(a, a) \in R$ for all $a \in S$)
  - transitive (if $(a, b) \in R$ and $(b, c) \in R$ then $(a, c) \in R$).
  - antisymmetric (if $(a, b) \in R$ and $(b, a) \in R$ then $a = b$).
- We usually write $a \preceq b$ or $b \succeq a$ for $(a, b) \in R$.

Partial orders

- A partial order on a set S is a binary relation (a subset R of $S \times S$ that is:
  - reflexive ($(a, a) \in R$ for all $a \in S$)
  - transitive (if $(a, b) \in R$ and $(b, c) \in R$ then $(a, c) \in R$).
  - antisymmetric (if $(a, b) \in R$ and $(b, a) \in R$ then $a = b$).
- We usually write $a \preceq b$ or $b \succeq a$ for $(a, b) \in R$.
- A strict partial order is like a partial order, but we remove reflexivity, and we write $a \prec b$. If we instead remove antisymmetry from a partial order, we get a preorder.

Partial orders

- A partial order on a set S is a binary relation (a subset R of $S \times S$ that is:
  - reflexive ($(a, a) \in R$ for all $a \in S$)
  - transitive (if $(a, b) \in R$ and $(b, c) \in R$ then $(a, c) \in R$).
  - antisymmetric (if $(a, b) \in R$ and $(b, a) \in R$ then $a = b$).
- We usually write $a \preceq b$ or $b \succeq a$ for $(a, b) \in R$.
- A strict partial order is like a partial order, but we remove reflexivity, and we write $a \prec b$. If we instead remove antisymmetry from a partial order, we get a preorder.
- An order is total if for every (a, b), it is the case that $(a, b) \in R$ or $(b, a) \in R$.

Graph basics

- A directed graph or digraph is a finite set V of nodes (or vertices) and a subset $E \subseteq V \times V$ of edges (or arcs). We write $G = (V, E)$.

Graph basics

- A directed graph or digraph is a finite set V of nodes (or vertices) and a subset $E \subseteq V \times V$ of edges (or arcs). We write $G = (V, E)$.
- A walk from node v to node w is a sequence of edges $(v, v_1), (v_1, v_2), \ldots, (v_{n-1}, w)$. A cycle is a walk where $v = w$. A path is a walk where all vertices are different.

Graph basics

- A directed graph or digraph is a finite set V of nodes (or vertices) and a subset $E \subseteq V \times V$ of edges (or arcs). We write $G = (V, E)$.
- A walk from node v to node w is a sequence of edges $(v, v_1), (v_1, v_2), \ldots, (v_{n-1}, w)$. A cycle is a walk where $v = w$. A path is a walk where all vertices are different.
- An acyclic digraph is one with no cycles. It can be represented as a partial order on V, where $v \geq w$ means (v, w) is an edge.

Graph basics

- A directed graph or digraph is a finite set V of nodes (or vertices) and a subset $E \subseteq V \times V$ of edges (or arcs). We write $G = (V, E)$.
- A walk from node v to node w is a sequence of edges $(v, v_1), (v_1, v_2), \ldots, (v_{n-1}, w)$. A cycle is a walk where $v = w$. A path is a walk where all vertices are different.
- An acyclic digraph is one with no cycles. It can be represented as a partial order on V, where $v \geq w$ means (v, w) is an edge.
- A graph is similar to a digraph but the edges are unordered (technically they are subsets of size 2 of V).

Graph basics

- A directed graph or digraph is a finite set V of nodes (or vertices) and a subset $E \subseteq V \times V$ of edges (or arcs). We write $G = (V, E)$.
- A walk from node v to node w is a sequence of edges $(v, v_1), (v_1, v_2), \dots, (v_{n-1}, w)$. A cycle is a walk where $v = w$. A path is a walk where all vertices are different.
- An acyclic digraph is one with no cycles. It can be represented as a partial order on V, where $v \geq w$ means (v, w) is an edge.
- A graph is similar to a digraph but the edges are unordered (technically they are subsets of size 2 of V).
- Everything above can be done also for weighted digraphs, where each edge has a real number called its weight.

Adjacency

- If we order the vertices $v_1, \ldots, v_n$, we can represent G by its (weighted) adjacency matrix, the $n \times n$ matrix $M = (m_{ij})$ for which $m_{ij} = 1$ if $(i, j) \in E$ and $m_{ij} = 0$ otherwise.

Adjacency

- If we order the vertices $v_1, \ldots, v_n$, we can represent G by its (weighted) adjacency matrix, the $n \times n$ matrix $M = (m_{ij})$ for which $m_{ij} = 1$ if $(i,j) \in E$ and $m_{ij} = 0$ otherwise.
- Thus M counts the "1-step" walks in G between each pair of nodes. Interestingly, the power M^k counts the k-step walks!

Preferences

- There are many ways to express preferences, which are simply orderings of a finite set A of alternatives such that $|A| = m$.

Preferences

- There are many ways to express preferences, which are simply orderings of a finite set A of alternatives such that $|A| = m$.
- We concentrate on:

Preferences

- There are many ways to express preferences, which are simply orderings of a finite set A of alternatives such that $|A| = m$.
- We concentrate on:
  - utility functions;

Preferences

- There are many ways to express preferences, which are simply orderings of a finite set A of alternatives such that $|A| = m$.
- We concentrate on:
  - utility functions;
  - strict ordinal preferences;

Preferences

- There are many ways to express preferences, which are simply orderings of a finite set A of alternatives such that $|A| = m$.
- We concentrate on:
  - utility functions;
  - strict ordinal preferences;
  - weak ordinal preferences;

Preferences

- There are many ways to express preferences, which are simply orderings of a finite set A of alternatives such that $|A| = m$.
- We concentrate on:
  - utility functions;
  - strict ordinal preferences;
  - weak ordinal preferences;
  - dichotomous (binary) preferences.

Preferences

- There are many ways to express preferences, which are simply orderings of a finite set A of alternatives such that $|A| = m$.
- We concentrate on:
  - utility functions;
  - strict ordinal preferences;
  - weak ordinal preferences;
  - dichotomous (binary) preferences.
- We may deal later with incomplete preferences but for now we assume that all elements are ranked.

Dichotomous preferences

- There is a subset of A that is approved. Usually we require this to be nonempty and often not to equal the whole set A.

Dichotomous preferences

- There is a subset of A that is approved. Usually we require this to be nonempty and often not to equal the whole set A.
- There is no further information obtainable about, for example, how approved items rank with respect to each other.

Dichotomous preferences

- There is a subset of A that is approved. Usually we require this to be nonempty and often not to equal the whole set A.
- There is no further information obtainable about, for example, how approved items rank with respect to each other.
- However all elements that are approved are ranked higher than all that are not approved.

Dichotomous preferences

- There is a subset of A that is approved. Usually we require this to be nonempty and often not to equal the whole set A.
- There is no further information obtainable about, for example, how approved items rank with respect to each other.
- However all elements that are approved are ranked higher than all that are not approved.
- Such preferences are perhaps easier to elicit but we still face the problem of "where to draw the line".

Dichotomous preferences

- There is a subset of A that is approved. Usually we require this to be nonempty and often not to equal the whole set A.
- There is no further information obtainable about, for example, how approved items rank with respect to each other.
- However all elements that are approved are ranked higher than all that are not approved.
- Such preferences are perhaps easier to elicit but we still face the problem of "where to draw the line".
- The number of such preferences is $2^m - 2$.

Weak ordinal preferences

- A weak ordinal preference is a total preorder on A.

Weak ordinal preferences

- A weak ordinal preference is a total preorder on A.
- In other words, the agent lists the elements of A in descending order of preference, and indifference is allowed (two consecutive elements may be tied).

Weak ordinal preferences

- A weak ordinal preference is a total preorder on A.
- In other words, the agent lists the elements of A in descending order of preference, and indifference is allowed (two consecutive elements may be tied).
- The number of ways to do this is the mth ordered Bell number.

Weak ordinal preferences

- A weak ordinal preference is a total preorder on A.
- In other words, the agent lists the elements of A in descending order of preference, and indifference is allowed (two consecutive elements may be tied).
- The number of ways to do this is the mth ordered Bell number.
- These numbers grow as $c^m m!$ for some constant $c > 1$.

Weak ordinal preferences

- A weak ordinal preference is a total preorder on A.
- In other words, the agent lists the elements of A in descending order of preference, and indifference is allowed (two consecutive elements may be tied).
- The number of ways to do this is the mth ordered Bell number.
- These numbers grow as $c^m m!$ for some constant $c > 1$.
- We use the symbol $\preceq$, so $a \preceq_i b$ means i prefers b to a.

Strict ordinal preferences

- A strict ordinal preference is a total linear order on A.

Strict ordinal preferences

- A strict ordinal preference is a total linear order on A.
- In other words, the elements are listed in order, and indifference is not allowed.

Strict ordinal preferences

- A strict ordinal preference is a total linear order on A.
- In other words, the elements are listed in order, and indifference is not allowed.
- The number of ways to do this is $m!$.

Strict ordinal preferences

- A strict ordinal preference is a total linear order on A.
- In other words, the elements are listed in order, and indifference is not allowed.
- The number of ways to do this is $m!$.
- This is the type of preference we will use most often. We use the symbol $\prec$.

Cardinal preferences

- A utility function is a function $u : A \to \mathbb{R}_+$. Intuitively, it measures how much the agent derives "utility" (enjoyment, payoff, ...) from each alternative.

Cardinal preferences

- A utility function is a function $u : A \to \mathbb{R}_+$. Intuitively, it measures how much the agent derives "utility" (enjoyment, payoff, ...) from each alternative.
- It induces a weak ordinal preference: $a \preceq b$ if and only if $u(a) \leq u(b)$.

Cardinal preferences

- A utility function is a function $u : A \to \mathbb{R}_+$. Intuitively, it measures how much the agent derives "utility" (enjoyment, payoff, ...) from each alternative.
- It induces a weak ordinal preference: $a \preceq b$ if and only if $u(a) \leq u(b)$.
- This allows us to compare across agents but requires us to elicit much more information. It is a controversial approach in some applications.

Cardinal preferences

- A utility function is a function $u : A \to \mathbb{R}_+$. Intuitively, it measures how much the agent derives "utility" (enjoyment, payoff, ...) from each alternative.
- It induces a weak ordinal preference: $a \preceq b$ if and only if $u(a) \leq u(b)$.
- This allows us to compare across agents but requires us to elicit much more information. It is a controversial approach in some applications.
- Preferences of this type are also called cardinal preferences.

Examples

- Let $A = \{a, b, c\}$. There are

Examples

- Let $A = \{a, b, c\}$. There are
  - 6 strict preference orders, namely $a \prec b \prec c, a \prec c \prec b, b \prec a \prec c, b \prec c \prec a, c \prec a \prec b, c \prec b \prec a$;

Examples

- Let $A = \{a, b, c\}$. There are
  - 6 strict preference orders, namely $a \prec b \prec c, a \prec c \prec b, b \prec a \prec c, b \prec c \prec a, c \prec a \prec b, c \prec b \prec a$;
  - 13 weak preference orders, namely those above plus $a \sim b \sim c, a \sim b \prec c, a \sim c \prec b, b \sim c \prec a, a \prec b \sim c, b \prec a \sim c, c \prec a \sim b$.

Examples

- Let $A = \{a, b, c\}$. There are
  - 6 strict preference orders, namely $a \prec b \prec c, a \prec c \prec b, b \prec a \prec c, b \prec c \prec a, c \prec a \prec b, c \prec b \prec a$;
  - 13 weak preference orders, namely those above plus $a \sim b \sim c, a \sim b \prec c, a \sim c \prec b, b \sim c \prec a, a \prec b \sim c, b \prec a \sim c, c \prec a \sim b$.
  - 6 dichotomous preferences, namely $a \mid bc, b \mid ac, c \mid ab, ab \mid c, ac \mid b, bc \mid a$.

Profiles

- Let S be the set of all possible preferences (in a particular model) over elements of A.

Profiles

- Let S be the set of all possible preferences (in a particular model) over elements of A.
- A profile is a function $V \to S$. The set of all profiles is denoted by $P(V, A)$.

Profiles

- Let S be the set of all possible preferences (in a particular model) over elements of A.
- A profile is a function $V \to S$. The set of all profiles is denoted by $P(V, A)$.
- A preference distribution is a multiset on S with weight n. In other words, we choose a natural number multiplicity for each element of S, and ensure the sum of multiplicities is n.

Profiles

- Let S be the set of all possible preferences (in a particular model) over elements of A.
- A profile is a function $V \to S$. The set of all profiles is denoted by $P(V, A)$.
- A preference distribution is a multiset on S with weight n. In other words, we choose a natural number multiplicity for each element of S, and ensure the sum of multiplicities is n.
- In other words, we count the number of agents with each preference.

Profiles

- Let S be the set of all possible preferences (in a particular model) over elements of A.
- A profile is a function $V \to S$. The set of all profiles is denoted by $P(V, A)$.
- A preference distribution is a multiset on S with weight n. In other words, we choose a natural number multiplicity for each element of S, and ensure the sum of multiplicities is n.
- In other words, we count the number of agents with each preference.
- The number of preference distributions for on S is $\binom{n+|S|-1}{|S|}$ (why?)

Overview

Situations in which a group must make a decision on a single outcome that affects all of them are very common. More generally, we are trying to combine individual preferences in order to obtain a reasonably satisfying outcome for the whole society. We will not study strategic behavior here – we focus on the method of aggregating preferences to obtain an outcome. We typically want our method to satisfy some “reasonable” properties. If we choose enough such properties, we can determine the rule uniquely. If we choose too many, we have an impossibility (nonexistence) result. It is interesting how few axioms are needed to obtain impossibility results.

Social choice

Social choice theory deals with problems of collective decision-making, and solutions to them. Such problems include:

- choosing an alternative;
- allocating resources;
- reaching consensus;
- forming coalitions;
- aggregating judgments and beliefs.

In order to model these, we must make many choices. Then we must analyse them!

Basic setup

- A set A of alternatives, of size m.

Basic setup

- A set A of alternatives, of size m.
- A set V of agents, of size n.

Basic setup

- A set A of alternatives, of size m.
- A set V of agents, of size n.
- Each agent has preferences over elements of A, and $P(V, A)$ is the set of all preference profiles.

Basic setup

- A set A of alternatives, of size m.
- A set V of agents, of size n.
- Each agent has preferences over elements of A, and $P(V, A)$ is the set of all preference profiles.
- We consider the set X of outcomes, which depends on the problem. For example, a typical element of X may be:

Basic setup

- A set A of alternatives, of size m.
- A set V of agents, of size n.
- Each agent has preferences over elements of A, and $P(V, A)$ is the set of all preference profiles.
- We consider the set X of outcomes, which depends on the problem. For example, a typical element of X may be:
  - a single alternative (voting)

Basic setup

- A set A of alternatives, of size m.
- A set V of agents, of size n.
- Each agent has preferences over elements of A, and $P(V, A)$ is the set of all preference profiles.
- We consider the set X of outcomes, which depends on the problem. For example, a typical element of X may be:
  - a single alternative (voting)
  - a subset of alternatives (committee selection)

Basic setup

- A set A of alternatives, of size m.
- A set V of agents, of size n.
- Each agent has preferences over elements of A, and $P(V, A)$ is the set of all preference profiles.
- We consider the set X of outcomes, which depends on the problem. For example, a typical element of X may be:
  - a single alternative (voting)
  - a subset of alternatives (committee selection)
  - a ranking of the alternatives

Basic setup

- A set A of alternatives, of size m.
- A set V of agents, of size n.
- Each agent has preferences over elements of A, and $P(V, A)$ is the set of all preference profiles.
- We consider the set X of outcomes, which depends on the problem. For example, a typical element of X may be:
  - a single alternative (voting)
  - a subset of alternatives (committee selection)
  - a ranking of the alternatives
  - an allocation of alternatives to agents (resource allocation).

The goal

- We seek an aggregation method, which is just a mapping from $P(V, A) \to X$.

The goal

- We seek an aggregation method, which is just a mapping from $P(V, A) \to X$.
- We will compare various methods in terms of two main principles: fairness and efficiency.

The goal

- We seek an aggregation method, which is just a mapping from $P(V, A) \to X$.
- We will compare various methods in terms of two main principles: fairness and efficiency.
- These concepts must be defined properly in each application. However there are a few key ideas that come up in many cases.

The goal

- We seek an aggregation method, which is just a mapping from $P(V, A) \to X$.
- We will compare various methods in terms of two main principles: fairness and efficiency.
- These concepts must be defined properly in each application. However there are a few key ideas that come up in many cases.
    - One fairness criterion is anonymity: the outcome should not depend on the identities of the agents, and should be symmetric.

The goal

- We seek an aggregation method, which is just a mapping from $P(V, A) \to X$.
- We will compare various methods in terms of two main principles: fairness and efficiency.
- These concepts must be defined properly in each application. However there are a few key ideas that come up in many cases.
  - One fairness criterion is anonymity: the outcome should not depend on the identities of the agents, and should be symmetric.
  - A common efficiency criterion is Pareto optimality (below).

The goal

- We seek an aggregation method, which is just a mapping from $P(V, A) \to X$.
- We will compare various methods in terms of two main principles: fairness and efficiency.
- These concepts must be defined properly in each application. However there are a few key ideas that come up in many cases.
  - One fairness criterion is anonymity: the outcome should not depend on the identities of the agents, and should be symmetric.
  - A common efficiency criterion is Pareto optimality (below).
- One potentially tricky issue (which we return to later) is that eliciting agents' preferences over outcomes may be much harder than eliciting preferences over alternatives, so that we sometimes need to estimate them.

Comparing outcomes

- Assuming that individuals have preferences over outcomes and that we know them, how can we compare two outcomes from the point of view of the group?

Comparing outcomes

- Assuming that individuals have preferences over outcomes and that we know them, how can we compare two outcomes from the point of view of the group?
- An outcome a is Pareto preferred to outcome b if every individual (usually called agent) prefers a to b.

Comparing outcomes

- Assuming that individuals have preferences over outcomes and that we know them, how can we compare two outcomes from the point of view of the group?
- An outcome a is Pareto preferred to outcome b if every individual (usually called agent) prefers a to b.
- An outcome b for which no outcome is Pareto preferred to it is called Pareto optimal.

Comparing outcomes

- Assuming that individuals have preferences over outcomes and that we know them, how can we compare two outcomes from the point of view of the group?
- An outcome a is Pareto preferred to outcome b if every individual (usually called agent) prefers a to b.
- An outcome b for which no outcome is Pareto preferred to it is called Pareto optimal.
- When we have cardinal utilities, we can consider the social welfare, an aggregation of all the individual utilities. The most common methods are:

Comparing outcomes

- Assuming that individuals have preferences over outcomes and that we know them, how can we compare two outcomes from the point of view of the group?
- An outcome a is Pareto preferred to outcome b if every individual (usually called agent) prefers a to b.
- An outcome b for which no outcome is Pareto preferred to it is called Pareto optimal.
- When we have cardinal utilities, we can consider the social welfare, an aggregation of all the individual utilities. The most common methods are:
    - utilitarian: sum of individual utilities;

Comparing outcomes

- Assuming that individuals have preferences over outcomes and that we know them, how can we compare two outcomes from the point of view of the group?
- An outcome a is Pareto preferred to outcome b if every individual (usually called agent) prefers a to b.
- An outcome b for which no outcome is Pareto preferred to it is called Pareto optimal.
- When we have cardinal utilities, we can consider the social welfare, an aggregation of all the individual utilities. The most common methods are:
    - utilitarian: sum of individual utilities;
    - egalitarian: minimum of all utilities.

Comparing outcomes

- Assuming that individuals have preferences over outcomes and that we know them, how can we compare two outcomes from the point of view of the group?
- An outcome a is Pareto preferred to outcome b if every individual (usually called agent) prefers a to b.
- An outcome b for which no outcome is Pareto preferred to it is called Pareto optimal.
- When we have cardinal utilities, we can consider the social welfare, an aggregation of all the individual utilities. The most common methods are:
  - utilitarian: sum of individual utilities;
  - egalitarian: minimum of all utilities.
- Both assume that it makes sense to compare utilities across agents, which can be very controversial.

Basic setup for voting

- A set A of alternatives, of size m.

Basic setup for voting

- A set A of alternatives, of size m.
- A set V of agents, of size n.

Basic setup for voting

- A set A of alternatives, of size m.
- A set V of agents, of size n.
- Each agent has preferences over elements of A, and $P(V, A)$ is the set of all preference profiles.

Basic setup for voting

- A set A of alternatives, of size m.
- A set V of agents, of size n.
- Each agent has preferences over elements of A, and $P(V, A)$ is the set of all preference profiles.
- We first consider the case where each preference is a complete linear ordering of the elements of A.

Basic setup for voting

- A set A of alternatives, of size m.
- A set V of agents, of size n.
- Each agent has preferences over elements of A, and $P(V, A)$ is the set of all preference profiles.
- We first consider the case where each preference is a complete linear ordering of the elements of A.
- We want an aggregation method that always outputs a single alternative.

Basic setup for voting

- A set A of alternatives, of size m.
- A set V of agents, of size n.
- Each agent has preferences over elements of A, and $P(V, A)$ is the set of all preference profiles.
- We first consider the case where each preference is a complete linear ordering of the elements of A.
- We want an aggregation method that always outputs a single alternative.
- We first consider doing this in a deterministic way.

Formal definitions

- A voting rule is a mapping $P(V, A) \to 2^A \setminus \{\emptyset\}$. Informally, it chooses a subset of the candidates as winners.

Formal definitions

- A voting rule is a mapping $P(V, A) \to 2^A \setminus \{\emptyset\}$. Informally, it chooses a subset of the candidates as winners.
- A resolute voting rule is a mapping $P(V, A) \to A$. Informally, it chooses a single winner in every case.

Formal definitions

- A voting rule is a mapping $P(V, A) \to 2^A \setminus \{\emptyset\}$. Informally, it chooses a subset of the candidates as winners.
- A resolute voting rule is a mapping $P(V, A) \to A$. Informally, it chooses a single winner in every case.
- Some people call voting rules social choice correspondences and resolute voting rules social choice functions.

Common axioms

Let F be a social choice correspondence.

- Anonymity: symmetry between voters

Note: for a social choice function, Pareto optimality and unanimity are the same.

Common axioms

Let F be a social choice correspondence.

- Anonymity: symmetry between voters
- Neutrality: symmetry between candidates

Note: for a social choice function, Pareto optimality and unanimity are the same.

Common axioms

Let F be a social choice correspondence.

- Anonymity: symmetry between voters
- Neutrality: symmetry between candidates
- Unanimity: if all voters rank x first, then x is the unique winner

Note: for a social choice function, Pareto optimality and unanimity are the same.

Common axioms

Let F be a social choice correspondence.

- Anonymity: symmetry between voters
- Neutrality: symmetry between candidates
- Unanimity: if all voters rank x first, then x is the unique winner
- Nonimposition: for each alternative, there is some distribution of voter preferences that makes it a winner

Note: for a social choice function, Pareto optimality and unanimity are the same.

Some strange social choice correspondences

- Omninominator rule: every candidate ranked first by some voter is a winner.

Some strange social choice correspondences

- Omninominator rule: every candidate ranked first by some voter is a winner.
- Dictatorship of voter i: the sole winner is the candidate at the top of i's preference order.

Some strange social choice correspondences

- Omninominator rule: every candidate ranked first by some voter is a winner.
- Dictatorship of voter i: the sole winner is the candidate at the top of i's preference order.
- Imposed rule of candidate j: candidate j is the sole winner.

Some strange social choice correspondences

- Omninominator rule: every candidate ranked first by some voter is a winner.
- Dictatorship of voter i: the sole winner is the candidate at the top of i's preference order.
- Imposed rule of candidate j: candidate j is the sole winner.
- A weird rule never used: candidate with the fewest first-place votes wins.

Some strange social choice correspondences

- Omninominator rule: every candidate ranked first by some voter is a winner.
- Dictatorship of voter i: the sole winner is the candidate at the top of i's preference order.
- Imposed rule of candidate j: candidate j is the sole winner.
- A weird rule never used: candidate with the fewest first-place votes wins.
- Which of the above axioms do these rules satisfy?

Complete symmetry is a problem for voting

- Suppose we have complete symmetry in the profile.

Complete symmetry is a problem for voting

- Suppose we have complete symmetry in the profile.
- For example, we may have 2 voters and 2 candidates a, b.

Complete symmetry is a problem for voting

- Suppose we have complete symmetry in the profile.
- For example, we may have 2 voters and 2 candidates a, b.
- If voter 1 has preference $a \succ b$ and voter 2 has preference $b \succ a$, then what should we do?

Complete symmetry is a problem for voting

- Suppose we have complete symmetry in the profile.
- For example, we may have 2 voters and 2 candidates a, b.
- If voter 1 has preference $a \succ b$ and voter 2 has preference $b \succ a$, then what should we do?
- If we require anonymity and neutrality (complete fairness to alternatives), then it is impossible to make a deterministic choice.

Complete symmetry is a problem for voting

- Suppose we have complete symmetry in the profile.
- For example, we may have 2 voters and 2 candidates a, b.
- If voter 1 has preference $a \succ b$ and voter 2 has preference $b \succ a$, then what should we do?
- If we require anonymity and neutrality (complete fairness to alternatives), then it is impossible to make a deterministic choice.
- We need a tiebreaking method. The most common are:

Complete symmetry is a problem for voting

- Suppose we have complete symmetry in the profile.
- For example, we may have 2 voters and 2 candidates a, b.
- If voter 1 has preference $a \succ b$ and voter 2 has preference $b \succ a$, then what should we do?
- If we require anonymity and neutrality (complete fairness to alternatives), then it is impossible to make a deterministic choice.
- We need a tiebreaking method. The most common are:
  - choose at random

Complete symmetry is a problem for voting

- Suppose we have complete symmetry in the profile.
- For example, we may have 2 voters and 2 candidates a, b.
- If voter 1 has preference $a \succ b$ and voter 2 has preference $b \succ a$, then what should we do?
- If we require anonymity and neutrality (complete fairness to alternatives), then it is impossible to make a deterministic choice.
- We need a tiebreaking method. The most common are:
  - choose at random
  - give extra influence to one voter

Complete symmetry is a problem for voting

- Suppose we have complete symmetry in the profile.
- For example, we may have 2 voters and 2 candidates a, b.
- If voter 1 has preference $a \succ b$ and voter 2 has preference $b \succ a$, then what should we do?
- If we require anonymity and neutrality (complete fairness to alternatives), then it is impossible to make a deterministic choice.
- We need a tiebreaking method. The most common are:
  - choose at random
  - give extra influence to one voter
  - give extra influence to one alternative.

The family of positional scoring rules

- If there are m candidates, fix a weight vector $w_1, w_2, \ldots, w_m$ such that $w_1 \geq w_2 \geq \cdots \geq w_m$ and $w_1 > w_m$.

The family of positional scoring rules

- If there are m candidates, fix a weight vector $w_1, w_2, \ldots, w_m$ such that $w_1 \geq w_2 \geq \cdots \geq w_m$ and $w_1 > w_m$.
- Each voter gives w_1 points to his most preferred candidate, w_2 to the next, ..., w_m to the least preferred.

The family of positional scoring rules

- If there are m candidates, fix a weight vector $w_1, w_2, \ldots, w_m$ such that $w_1 \geq w_2 \geq \cdots \geq w_m$ and $w_1 > w_m$.
- Each voter gives w_1 points to his most preferred candidate, w_2 to the next, ..., w_m to the least preferred.
- The candidate with highest total score wins.

The family of positional scoring rules

- If there are m candidates, fix a weight vector $w_1, w_2, \ldots, w_m$ such that $w_1 \geq w_2 \geq \cdots \geq w_m$ and $w_1 > w_m$.
- Each voter gives w_1 points to his most preferred candidate, w_2 to the next, ..., w_m to the least preferred.
- The candidate with highest total score wins.
- Usually we normalize (via an affine transformation) so that $w_1 = 1, w_m = 0$.

The family of positional scoring rules

- If there are m candidates, fix a weight vector $w_1, w_2, \ldots, w_m$ such that $w_1 \geq w_2 \geq \cdots \geq w_m$ and $w_1 > w_m$.
- Each voter gives w_1 points to his most preferred candidate, w_2 to the next, ..., w_m to the least preferred.
- The candidate with highest total score wins.
- Usually we normalize (via an affine transformation) so that $w_1 = 1, w_m = 0$.
- Note that some familiar rules fall into this class:

The family of positional scoring rules

- If there are m candidates, fix a weight vector $w_1, w_2, \ldots, w_m$ such that $w_1 \geq w_2 \geq \cdots \geq w_m$ and $w_1 > w_m$.
- Each voter gives w_1 points to his most preferred candidate, w_2 to the next, ..., w_m to the least preferred.
- The candidate with highest total score wins.
- Usually we normalize (via an affine transformation) so that $w_1 = 1, w_m = 0$.
- Note that some familiar rules fall into this class:
  - plurality: $w = (1, 0, \ldots, 0)$

The family of positional scoring rules

- If there are m candidates, fix a weight vector $w_1, w_2, \ldots, w_m$ such that $w_1 \geq w_2 \geq \cdots \geq w_m$ and $w_1 > w_m$.
- Each voter gives w_1 points to his most preferred candidate, w_2 to the next, ..., w_m to the least preferred.
- The candidate with highest total score wins.
- Usually we normalize (via an affine transformation) so that $w_1 = 1, w_m = 0$.
- Note that some familiar rules fall into this class:
  - plurality: $w = (1, 0, \ldots, 0)$
  - antiplurality $w = (1, 1, \ldots, 1, 0)$

The family of positional scoring rules

- If there are m candidates, fix a weight vector $w_1, w_2, \ldots, w_m$ such that $w_1 \geq w_2 \geq \cdots \geq w_m$ and $w_1 > w_m$.
- Each voter gives w_1 points to his most preferred candidate, w_2 to the next, ..., w_m to the least preferred.
- The candidate with highest total score wins.
- Usually we normalize (via an affine transformation) so that $w_1 = 1, w_m = 0$.
- Note that some familiar rules fall into this class:
  - plurality: $w = (1, 0, \ldots, 0)$
  - antiplurality $w = (1, 1, \ldots, 1, 0)$
  - Borda: $w = (m-1, m-2, \ldots, 1, 0)$.

Geometrically defined rules

- If all voters agree (a consensus) on the top choice, or on the whole ranking, then it is easy to see who the winner should be.

Geometrically defined rules

- If all voters agree (a consensus) on the top choice, or on the whole ranking, then it is easy to see who the winner should be.
- A general idea is to use a notion of distance on the set of profiles to find the nearest consensus, and then choose the winner.

Geometrically defined rules

- If all voters agree (a consensus) on the top choice, or on the whole ranking, then it is easy to see who the winner should be.
- A general idea is to use a notion of distance on the set of profiles to find the nearest consensus, and then choose the winner.
- We have a lot of flexibility in the choice of distance.

Kemeny's rule

- This rule first chooses a ranking of candidates ("the Kemeny consensus") minimizing the total distance to each element of the profile, then chooses the top element of this ranking.

Kemeny's rule

- This rule first chooses a ranking of candidates ("the Kemeny consensus") minimizing the total distance to each element of the profile, then chooses the top element of this ranking.
- Distance is measured in terms of inversions: the distance between two permutations is the number of swaps needed to convert one into the other (the number of swaps used by bubblesort).

Kemeny's rule

- This rule first chooses a ranking of candidates ("the Kemeny consensus") minimizing the total distance to each element of the profile, then chooses the top element of this ranking.
- Distance is measured in terms of inversions: the distance between two permutations is the number of swaps needed to convert one into the other (the number of swaps used by bubblesort).
- It turns out that if we instead find the closest ranking where every voter agrees on the top alternative, we get the Borda rule.

Slater's rule

- Start with the majority graph (a tournament), and invert (flip direction of) as few arcs as possible until we reach an acyclic tournament. The winner is the winner of that tournament.

Slater's rule

- Start with the majority graph (a tournament), and invert (flip direction of) as few arcs as possible until we reach an acyclic tournament. The winner is the winner of that tournament.
- Kemeny's rule can be interpreted in a similar way: start with the weighted majority graph and minimize the sum of weights of inverted arcs.

Condorcet principle and paradox

- Suppose candidate a is elected as sole winner by some voting rule.

Condorcet principle and paradox

- Suppose candidate a is elected as sole winner by some voting rule.
- If a majority of voters prefer some other candidate b, many voters will be dissatisfied.

Condorcet principle and paradox

- Suppose candidate a is elected as sole winner by some voting rule.
- If a majority of voters prefer some other candidate b, many voters will be dissatisfied.
- If a majority of voters prefer b to every other candidate, then if anyone other than b is the winner, a majority will be dissatisfied.

Condorcet principle and paradox

- Suppose candidate a is elected as sole winner by some voting rule.
- If a majority of voters prefer some other candidate b, many voters will be dissatisfied.
- If a majority of voters prefer b to every other candidate, then if anyone other than b is the winner, a majority will be dissatisfied.
- Condorcet's principle: if a Condorcet winner exists, it should be the sole winner — no rule violating this should be used. The rules that satisfy the principle are Condorcet consistent.

Condorcet principle and paradox

- Suppose candidate a is elected as sole winner by some voting rule.
- If a majority of voters prefer some other candidate b, many voters will be dissatisfied.
- If a majority of voters prefer b to every other candidate, then if anyone other than b is the winner, a majority will be dissatisfied.
- Condorcet's principle: if a Condorcet winner exists, it should be the sole winner — no rule violating this should be used. The rules that satisfy the principle are Condorcet consistent.
- Note that a CW need not exist: if we have 3 candidates a, b, c and voters abc, cab, bca then no candidate is preferred by a majority to the others. This is the Condorcet paradox.

The majority digraph

- Create a node for each alternative and an arc from a to b if a strict majority of voters prefer a to b.

The majority digraph

- Create a node for each alternative and an arc from a to b if a strict majority of voters prefer a to b.
- This digraph is a tournament (its underlying graph is complete) if n is odd.

The majority digraph

- Create a node for each alternative and an arc from a to b if a strict majority of voters prefer a to b.
- This digraph is a tournament (its underlying graph is complete) if n is odd.
- A weighted version is also useful. If a defeats b by a margin of w in a pairwise election then there is an arc from a to b with weight w.

The majority digraph

- Create a node for each alternative and an arc from a to b if a strict majority of voters prefer a to b.
- This digraph is a tournament (its underlying graph is complete) if n is odd.
- A weighted version is also useful. If a defeats b by a margin of w in a pairwise election then there is an arc from a to b with weight w.
- Each procedure for specifying the (unique) winner of a tournament, which always chooses a source node if there is one, yields a (resolute) voting rule satisfying Condorcet's principle.

Copeland's rule

- Based on the idea of a round-robin tournament in chess or football.

Copeland's rule

- Based on the idea of a round-robin tournament in chess or football.
- The Copeland score of a is the number of arcs going out of a (the outdegree) in the majority digraph.

Copeland's rule

- Based on the idea of a round-robin tournament in chess or football.
- The Copeland score of a is the number of arcs going out of a (the outdegree) in the majority digraph.
- If there is a tie in the majority tournament, award some fixed α to each node, with $0 \leq \alpha \leq 1$.

Copeland's rule

- Based on the idea of a round-robin tournament in chess or football.
- The Copeland score of a is the number of arcs going out of a (the outdegree) in the majority digraph.
- If there is a tie in the majority tournament, award some fixed α to each node, with $0 \leq \alpha \leq 1$.
- The winner is the candidate with highest total score.

Copeland's rule

- Based on the idea of a round-robin tournament in chess or football.
- The Copeland score of a is the number of arcs going out of a (the outdegree) in the majority digraph.
- If there is a tie in the majority tournament, award some fixed α to each node, with $0 \leq \alpha \leq 1$.
- The winner is the candidate with highest total score.
- This rule often elects more than one winner.

Copeland's rule

- Based on the idea of a round-robin tournament in chess or football.
- The Copeland score of a is the number of arcs going out of a (the outdegree) in the majority digraph.
- If there is a tie in the majority tournament, award some fixed α to each node, with $0 \leq \alpha \leq 1$.
- The winner is the candidate with highest total score.
- This rule often elects more than one winner.
- This rule is clearly Condorcet consistent.

Plurality with runoff

- There are two rounds of voting.

Plurality with runoff

- There are two rounds of voting.
- In the first round, plurality is used and the two highest-scoring alternatives advance to the next round.

Plurality with runoff

- There are two rounds of voting.
- In the first round, plurality is used and the two highest-scoring alternatives advance to the next round.
- In the second round, plurality is used on the new set of candidates.

Plurality with runoff

- There are two rounds of voting.
- In the first round, plurality is used and the two highest-scoring alternatives advance to the next round.
- In the second round, plurality is used on the new set of candidates.
- This is used to elect the French President.

Plurality with runoff

- There are two rounds of voting.
- In the first round, plurality is used and the two highest-scoring alternatives advance to the next round.
- In the second round, plurality is used on the new set of candidates.
- This is used to elect the French President.
- It has some counter-intuitive properties. For example, adding support to a candidate may make it go from winning to losing (not positively responsive); abstaining from voting for a candidate may turn it from a loser into a winner.

Instant runoff

- Each voter submits a full preference order of candidates.

Instant runoff

- Each voter submits a full preference order of candidates.
- At each stage, the plurality loser is eliminated and scores recalculated.

Instant runoff

- Each voter submits a full preference order of candidates.
- At each stage, the plurality loser is eliminated and scores recalculated.
- If at any stage a candidate scores more than half, it is the winner.

Example: runoff methods

- Suppose we have 3 candidates a, b, c and preferences as follows: 25 abc, 46 cab, 24 bca.

Example: runoff methods

- Suppose we have 3 candidates a, b, c and preferences as follows: 25 abc, 46 cab, 24 bca.
- Plurality with runoff: b is eliminated in round 1. Voters can vote again. If they do not change preferences, then c beats a in round 2.

Example: runoff methods

- Suppose we have 3 candidates a, b, c and preferences as follows: 25 abc, 46 cab, 24 bca.
- Plurality with runoff: b is eliminated in round 1. Voters can vote again. If they do not change preferences, then c beats a in round 2.
- Instant runoff: b is eliminated, and then c beats a as above.

Example: runoff methods

- Suppose we have 3 candidates a, b, c and preferences as follows: 25 abc, 46 cab, 24 bca.
- Plurality with runoff: b is eliminated in round 1. Voters can vote again. If they do not change preferences, then c beats a in round 2.
- Instant runoff: b is eliminated, and then c beats a as above.
- Suppose that 2 of the abc voters fail to vote. Then b wins, because a is eliminated in round 1. Note that these voters are better off not voting at all (the no-show paradox).

The assignment problem

- Suppose that $|V| = |A|$. An assignment (sometimes called pure assignment or discrete assignment) is a bijection (one-to-one and onto function) $V \to A$.

The assignment problem

- Suppose that $|V| = |A|$. An assignment (sometimes called pure assignment or discrete assignment) is a bijection (one-to-one and onto function) $V \to A$.
- In other words, we allocate a unique element of A to each element of V.

The assignment problem

- Suppose that $|V| = |A|$. An assignment (sometimes called pure assignment or discrete assignment) is a bijection (one-to-one and onto function) $V \to A$.
- In other words, we allocate a unique element of A to each element of V.
- This is sometimes called the house allocation problem.

The assignment problem

- Suppose that $|V| = |A|$. An assignment (sometimes called pure assignment or discrete assignment) is a bijection (one-to-one and onto function) $V \to A$.
- In other words, we allocate a unique element of A to each element of V.
- This is sometimes called the house allocation problem.
- It was first studied formally in 1979 but seems very basic!

Practical applications of this and related problems

- Assigning dormitory rooms to students.

Practical applications of this and related problems

- Assigning dormitory rooms to students.
- Assigning papers for student presentation (this course!)

Practical applications of this and related problems

- Assigning dormitory rooms to students.
- Assigning papers for student presentation (this course!)
- Assigning time slots for student presentations (this course!)

Practical applications of this and related problems

- Assigning dormitory rooms to students.
- Assigning papers for student presentation (this course!)
- Assigning time slots for student presentations (this course!)
- Assigning players to sports teams.

Practical applications of this and related problems

- Assigning dormitory rooms to students.
- Assigning papers for student presentation (this course!)
- Assigning time slots for student presentations (this course!)
- Assigning players to sports teams.
- Kidney donation.

Practical applications of this and related problems

- Assigning dormitory rooms to students.
- Assigning papers for student presentation (this course!)
- Assigning time slots for student presentations (this course!)
- Assigning players to sports teams.
- Kidney donation.
- Assigning conference papers to reviewers.

Preferences over outcomes

- The set of assignments is very large, so eliciting preferences over them will be difficult.

Preferences over outcomes

- The set of assignments is very large, so eliciting preferences over them will be difficult.
- We usually assume that each agent cares only about their own item, and not about what other agents receive.

Preferences over outcomes

- The set of assignments is very large, so eliciting preferences over them will be difficult.
- We usually assume that each agent cares only about their own item, and not about what other agents receive.
- This way, we do not need any extra information.

Serial dictatorship

- This algorithm proceeds as follows:

Serial dictatorship

- This algorithm proceeds as follows:
  - Fix a strict total ordering of the agents.

Serial dictatorship

- This algorithm proceeds as follows:
  - Fix a strict total ordering of the agents.
  - Assign agents to objects in turn, each being assigned to the remaining object highest on the agent's list.

Serial dictatorship

- This algorithm proceeds as follows:
  - Fix a strict total ordering of the agents.
  - Assign agents to objects in turn, each being assigned to the remaining object highest on the agent's list.
  - Remove each item from A after assignment.

Serial dictatorship

- This algorithm proceeds as follows:
  - Fix a strict total ordering of the agents.
  - Assign agents to objects in turn, each being assigned to the remaining object highest on the agent's list.
  - Remove each item from A after assignment.
  - Terminate when there are no more items left.

Top Trading Cycles

- This algorithm proceeds as follows:

Top Trading Cycles

- This algorithm proceeds as follows:
  - Fix some initial assignment of items to agents.

Top Trading Cycles

- This algorithm proceeds as follows:
  - Fix some initial assignment of items to agents.
  - Each agent points to (the agent holding) their top item.

Top Trading Cycles

- This algorithm proceeds as follows:
  - Fix some initial assignment of items to agents.
  - Each agent points to (the agent holding) their top item.
  - The resulting digraph has cycles; reallocate all objects along cycles.

Top Trading Cycles

- This algorithm proceeds as follows:
  - Fix some initial assignment of items to agents.
  - Each agent points to (the agent holding) their top item.
  - The resulting digraph has cycles; reallocate all objects along cycles.
  - Remove each (agent, item) pair from all the cycles.

Top Trading Cycles

- This algorithm proceeds as follows:
  - Fix some initial assignment of items to agents.
  - Each agent points to (the agent holding) their top item.
  - The resulting digraph has cycles; reallocate all objects along cycles.
  - Remove each (agent, item) pair from all the cycles.
  - Repeat the process with the remaining agents and their 2nd choice items, etc.

Top Trading Cycles

- This algorithm proceeds as follows:
  - Fix some initial assignment of items to agents.
  - Each agent points to (the agent holding) their top item.
  - The resulting digraph has cycles; reallocate all objects along cycles.
  - Remove each (agent, item) pair from all the cycles.
  - Repeat the process with the remaining agents and their 2nd choice items, etc.
  - Terminate when there are no more items left.

Boston method

- This algorithm proceeds as follows:

Boston method

- This algorithm proceeds as follows:
  - Fix an order of agents (to be used for tiebreaking).

Boston method

- This algorithm proceeds as follows:
  - Fix an order of agents (to be used for tiebreaking).
  - Each agent bids for their top item.

Boston method

- This algorithm proceeds as follows:
  - Fix an order of agents (to be used for tiebreaking).
  - Each agent bids for their top item.
  - Break ties according to the given order above, and allocate as many objects as possible.

Boston method

- This algorithm proceeds as follows:
  - Fix an order of agents (to be used for tiebreaking).
  - Each agent bids for their top item.
  - Break ties according to the given order above, and allocate as many objects as possible.
  - Remove each (agent, item) pair.

Boston method

- This algorithm proceeds as follows:
    - Fix an order of agents (to be used for tiebreaking).
    - Each agent bids for their top item.
    - Break ties according to the given order above, and allocate as many objects as possible.
    - Remove each (agent, item) pair.
    - Repeat the process with the remaining agents and their 2nd choice items, etc.

Boston method

- This algorithm proceeds as follows:
  - Fix an order of agents (to be used for tiebreaking).
  - Each agent bids for their top item.
  - Break ties according to the given order above, and allocate as many objects as possible.
  - Remove each (agent, item) pair.
  - Repeat the process with the remaining agents and their 2nd choice items, etc.
  - Terminate when there are no more items left.

Axiomatic properties

- SD is Pareto efficient and strategyproof (there is no incentive to lie about one's preferences). This is pretty obvious.

Axiomatic properties

- SD is Pareto efficient and strategyproof (there is no incentive to lie about one's preferences). This is pretty obvious.
- TTC is Pareto efficient and strategyproof. This requires more work to prove.

Axiomatic properties

- SD is Pareto efficient and strategyproof (there is no incentive to lie about one's preferences). This is pretty obvious.
- TTC is Pareto efficient and strategyproof. This requires more work to prove.
- Boston is Pareto efficient but not strategyproof.

Fairness

- Suppose that every agent has the same preference order.

Fairness

- Suppose that every agent has the same preference order.
- No deterministic assignment method can treat agents equally in that situation.

Fairness

- Suppose that every agent has the same preference order.
- No deterministic assignment method can treat agents equally in that situation.
- If we insist on treating agents symmetrically (anonymity) then randomization is inevitable.

Fairness

- Suppose that every agent has the same preference order.
- No deterministic assignment method can treat agents equally in that situation.
- If we insist on treating agents symmetrically (anonymity) then randomization is inevitable.
- A common way is to choose the order of agents (in SD) or the initial endowment (in TTC) uniformly at random.

Fairness

- Suppose that every agent has the same preference order.
- No deterministic assignment method can treat agents equally in that situation.
- If we insist on treating agents symmetrically (anonymity) then randomization is inevitable.
- A common way is to choose the order of agents (in SD) or the initial endowment (in TTC) uniformly at random.
- This still creates some unfairness.

Random assignments

- A random assignment is a probability distribution on assignments.

Random assignments

- A random assignment is a probability distribution on assignments.
- The expectation is called a fractional assignment. It is represented by a doubly stochastic matrix.

Random assignments

- A random assignment is a probability distribution on assignments.
- The expectation is called a fractional assignment. It is represented by a doubly stochastic matrix.
- We can "pretend" that items are infinitely divisible and interpret each entry as a fraction of the item that we receive, or fraction of time we get to use it.

Random assignments

- A random assignment is a probability distribution on assignments.
- The expectation is called a fractional assignment. It is represented by a doubly stochastic matrix.
- We can "pretend" that items are infinitely divisible and interpret each entry as a fraction of the item that we receive, or fraction of time we get to use it.
- Interestingly, the randomized versions of SD and TTC algorithms always give the same fractional assignment on each input (proved in 1998).

Preferences over random assignments

- It is not obvious what it means for a random assignment to be preferred to another one.

Preferences over random assignments

- It is not obvious what it means for a random assignment to be preferred to another one.
- A standard way is to consider the expected utility derived by the agent. However this requires cardinal preferences.

Preferences over random assignments

- It is not obvious what it means for a random assignment to be preferred to another one.
- A standard way is to consider the expected utility derived by the agent. However this requires cardinal preferences.
- Another way (using only ordinal preferences) is to consider stochastic dominance.

Preferences over random assignments

- It is not obvious what it means for a random assignment to be preferred to another one.
- A standard way is to consider the expected utility derived by the agent. However this requires cardinal preferences.
- Another way (using only ordinal preferences) is to consider stochastic dominance.
- A probability distribution P stochastically dominates another P' according to agent i if for each k,

$$\sum_{k \prec_i j} P(j) \geq \sum_{k \prec_i j} P'(j)$$

and there is some k for which the inequality is strict.

Preferences over random assignments

- It is not obvious what it means for a random assignment to be preferred to another one.
- A standard way is to consider the expected utility derived by the agent. However this requires cardinal preferences.
- Another way (using only ordinal preferences) is to consider stochastic dominance.
- A probability distribution P stochastically dominates another P' according to agent i if for each k,

$$\sum_{k \prec_i j} P(j) \geq \sum_{k \prec_i j} P'(j)$$

and there is some k for which the inequality is strict.
- Alternatively, the expected return from gambling on P is higher than that from gambling on P', *no matter what utilities the agent may have as long as they are consistent with* $\prec_i$.

PS algorithm

- Think of each item as being infinitely divisible, so that the probability of an agent getting item i can be thought of as the fraction of item i that the agent receives.

PS algorithm

- Think of each item as being infinitely divisible, so that the probability of an agent getting item i can be thought of as the fraction of item i that the agent receives.
- Each agent "eats" items at unit speed, choosing from the most preferred item that has not yet been consumed.

PS algorithm

- Think of each item as being infinitely divisible, so that the probability of an agent getting item i can be thought of as the fraction of item i that the agent receives.
- Each agent "eats" items at unit speed, choosing from the most preferred item that has not yet been consumed.
- Note that several agents may be eating a given item at the same time.

PS algorithm

- Think of each item as being infinitely divisible, so that the probability of an agent getting item i can be thought of as the fraction of item i that the agent receives.
- Each agent "eats" items at unit speed, choosing from the most preferred item that has not yet been consumed.
- Note that several agents may be eating a given item at the same time.
- On termination this yields a random assignment.

Example

- Suppose we have items a, b, c, d and agents $1, 2, 3, 4$ with strict ordinal preferences:

Example

- Suppose we have items a, b, c, d and agents $1, 2, 3, 4$ with strict ordinal preferences:
  - Two agents have $a \prec b \prec c \prec d$;

Example

- Suppose we have items a, b, c, d and agents $1, 2, 3, 4$ with strict ordinal preferences:
  - Two agents have $a \prec b \prec c \prec d$;
  - Two agents have $b \prec a \prec d \prec c$.

Example

- Suppose we have items a, b, c, d and agents $1, 2, 3, 4$ with strict ordinal preferences:
  - Two agents have $a \prec b \prec c \prec d$;
  - Two agents have $b \prec a \prec d \prec c$.
- RSD and PS respectively give the following fractional assignments.

$$\begin{bmatrix} \frac{5}{12} & \frac{1}{12} & \frac{5}{12} & \frac{1}{12} \\ \frac{5}{12} & \frac{1}{12} & \frac{5}{12} & \frac{1}{12} \\ \frac{1}{12} & \frac{5}{12} & \frac{1}{12} & \frac{5}{12} \\ \frac{1}{12} & \frac{5}{12} & \frac{1}{12} & \frac{5}{12} \end{bmatrix}, \qquad \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

Example

- Suppose we have items a, b, c, d and agents $1, 2, 3, 4$ with strict ordinal preferences:
  - Two agents have $a \prec b \prec c \prec d$;
  - Two agents have $b \prec a \prec d \prec c$.
- RSD and PS respectively give the following fractional assignments.

$$\begin{bmatrix} \frac{5}{12} & \frac{1}{12} & \frac{5}{12} & \frac{1}{12} \\ \frac{5}{12} & \frac{1}{12} & \frac{5}{12} & \frac{1}{12} \\ \frac{1}{12} & \frac{5}{12} & \frac{1}{12} & \frac{5}{12} \\ \frac{1}{12} & \frac{5}{12} & \frac{1}{12} & \frac{5}{12} \end{bmatrix}, \qquad \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}$$

- Note that each agent prefers (in terms of stochastic dominance) the outcome under PS to that under RSD.

Overview

In electoral systems, we seek fair allocations of district seats based on population (e.g. in US Congress), or allocation of seats to parties based on votes (e.g. in many countries' parliaments). The mathematics is almost exactly the same, and there are two names for many concepts. The ideas can be applied to other situations.

Apportionment

- The basic setup: S items (seats) to be divided among n agents, with agent i having weight P_i, $\sum_i P_i = P$.

Apportionment

- The basic setup: S items (seats) to be divided among n agents, with agent i having weight P_i, $\sum_i P_i = P$.
- Preferences are simple: every agent prefers more seats to fewer, and seats are identical.

Apportionment

- The basic setup: S items (seats) to be divided among n agents, with agent i having weight P_i, $\sum_i P_i = P$.
- Preferences are simple: every agent prefers more seats to fewer, and seats are identical.
- Proportionality is desired, and the difficulty comes from the fact that allocations must be discrete.

Apportionment

- The basic setup: S items (seats) to be divided among n agents, with agent i having weight P_i, $\sum_i P_i = P$.
- Preferences are simple: every agent prefers more seats to fewer, and seats are identical.
- Proportionality is desired, and the difficulty comes from the fact that allocations must be discrete.
- Let $S_i := Ss_i$ be the allocation of items to agent i, $P_i := p_i P$, so $\sum_i s_i = 1 = \sum_i p_i$.

Historical notes

- US Constitution (1787): "Representatives and direct taxes shall be apportioned among the several States which may be included within this Union, according to their respective numbers, ... The number of Representatives shall not exceed one for every thirty thousand, but each State shall have at least one Representative ...".

Historical notes

- US Constitution (1787): "Representatives and direct taxes shall be apportioned among the several States which may be included within this Union, according to their respective numbers, ... The number of Representatives shall not exceed one for every thirty thousand, but each State shall have at least one Representative ...".
- Hamilton's method vetoed by George Washington 1792, Jefferson's used.

Historical notes

- US Constitution (1787): “Representatives and direct taxes shall be apportioned among the several States which may be included within this Union, according to their respective numbers, ... The number of Representatives shall not exceed one for every thirty thousand, but each State shall have at least one Representative ...”.
- Hamilton’s method vetoed by George Washington 1792, Jefferson’s used.
- 1842 Webster adopted.

Historical notes

- US Constitution (1787): "Representatives and direct taxes shall be apportioned among the several States which may be included within this Union, according to their respective numbers, ... The number of Representatives shall not exceed one for every thirty thousand, but each State shall have at least one Representative ...".
- Hamilton's method vetoed by George Washington 1792, Jefferson's used.
- 1842 Webster adopted.
- 1852 Hamilton adopted.

Historical notes

- US Constitution (1787): "Representatives and direct taxes shall be apportioned among the several States which may be included within this Union, according to their respective numbers, ... The number of Representatives shall not exceed one for every thirty thousand, but each State shall have at least one Representative ...".
- Hamilton's method vetoed by George Washington 1792, Jefferson's used.
- 1842 Webster adopted.
- 1852 Hamilton adopted.
- 1872 apportionment done wrongly; changed result of 1876 presidential election.

Historical notes

- US Constitution (1787): "Representatives and direct taxes shall be apportioned among the several States which may be included within this Union, according to their respective numbers, ... The number of Representatives shall not exceed one for every thirty thousand, but each State shall have at least one Representative ...".
- Hamilton's method vetoed by George Washington 1792, Jefferson's used.
- 1842 Webster adopted.
- 1852 Hamilton adopted.
- 1872 apportionment done wrongly; changed result of 1876 presidential election.
- 1901 Webster adopted.

Historical notes

- US Constitution (1787): "Representatives and direct taxes shall be apportioned among the several States which may be included within this Union, according to their respective numbers, ... The number of Representatives shall not exceed one for every thirty thousand, but each State shall have at least one Representative ...".
- Hamilton's method vetoed by George Washington 1792, Jefferson's used.
- 1842 Webster adopted.
- 1852 Hamilton adopted.
- 1872 apportionment done wrongly; changed result of 1876 presidential election.
- 1901 Webster adopted.
- 1941 Huntington-Hill adopted, still used; some experts still lobbying for Webster.

Axioms

- Lower quota: $S_i \geq \lfloor Sp_i \rfloor$ (each agent gets at least its lower quota).

Axioms

- Lower quota: $S_i \geq \lfloor Sp_i \rfloor$ (each agent gets at least its lower quota).
- Upper quota: $S_i \leq \lceil Sp_i \rceil$ (no agent gets more than its upper quota).

Axioms

- Lower quota: $S_i \geq \lfloor Sp_i \rfloor$ (each agent gets at least its lower quota).
- Upper quota: $S_i \leq \lceil Sp_i \rceil$ (no agent gets more than its upper quota).
- House monotonicity: if S increases to S' and $p_i = p_i'$ for all i then $S_i' \geq S_i$ for all i.

Axioms

- Lower quota: $S_i \geq \lfloor Sp_i \rfloor$ (each agent gets at least its lower quota).
- Upper quota: $S_i \leq \lceil Sp_i \rceil$ (no agent gets more than its upper quota).
- House monotonicity: if S increases to S' and $p_i = p_i'$ for all i then $S_i' \geq S_i$ for all i.
- Population monotonicity: if S is fixed and p_i increases while p_j decreases, i should not lose a seat to j.

Axioms

- Lower quota: $S_i \geq \lfloor Sp_i \rfloor$ (each agent gets at least its lower quota).
- Upper quota: $S_i \leq \lceil Sp_i \rceil$ (no agent gets more than its upper quota).
- House monotonicity: if S increases to S' and $p_i = p'_i$ for all i then $S'_i \geq S_i$ for all i.
- Population monotonicity: if S is fixed and p_i increases while p_j decreases, i should not lose a seat to j.
- New agents: if n increases, S is fixed and the old P_i do not increase, no s_i should increase.

Axioms

- Lower quota: $S_i \geq \lfloor Sp_i \rfloor$ (each agent gets at least its lower quota).
- Upper quota: $S_i \leq \lceil Sp_i \rceil$ (no agent gets more than its upper quota).
- House monotonicity: if S increases to S' and $p_i = p_i'$ for all i then $S_i' \geq S_i$ for all i.
- Population monotonicity: if S is fixed and p_i increases while p_j decreases, i should not lose a seat to j.
- New agents: if n increases, S is fixed and the old P_i do not increase, no s_i should increase.
- Balinski & Young (1982) show that there is no apportionment algorithm that always satisfies the first 4 axioms. In fact anything satisfying the first two fails the fourth and fifth.

Axioms

- Lower quota: $S_i \geq \lfloor Sp_i \rfloor$ (each agent gets at least its lower quota).
- Upper quota: $S_i \leq \lceil Sp_i \rceil$ (no agent gets more than its upper quota).
- House monotonicity: if S increases to S' and $p_i = p_i'$ for all i then $S_i' \geq S_i$ for all i.
- Population monotonicity: if S is fixed and p_i increases while p_j decreases, i should not lose a seat to j.
- New agents: if n increases, S is fixed and the old P_i do not increase, no s_i should increase.
- Balinski & Young (1982) show that there is no apportionment algorithm that always satisfies the first 4 axioms. In fact anything satisfying the first two fails the fourth and fifth.
- There are many other axioms involving anonymity, lack of bias toward agents of large (or small) weight, etc.

Largest remainder algorithms

- The largest remainders (or Hamilton) method first allocates the lower quota $\lfloor Sp_i \rfloor$ to agent i. The remaining seats are allocated in decreasing order of the fractional part $Sp_i - \lfloor Sp_i \rfloor$.

Largest remainder algorithms

- The largest remainders (or Hamilton) method first allocates the lower quota $\lfloor Sp_i \rfloor$ to agent i. The remaining seats are allocated in decreasing order of the fractional part $Sp_i - \lfloor Sp_i \rfloor$.
- By construction this always satisfies the quota axioms.

Largest remainder algorithms

- The largest remainders (or Hamilton) method first allocates the lower quota $\lfloor Sp_i \rfloor$ to agent i. The remaining seats are allocated in decreasing order of the fractional part $Sp_i - \lfloor Sp_i \rfloor$.
- By construction this always satisfies the quota axioms.
- There are other largest remainder methods, which use different formulae in place of the lower quota, such as the Droop quota.

Largest remainder algorithms

- The largest remainders (or Hamilton) method first allocates the lower quota $\lfloor Sp_i \rfloor$ to agent i. The remaining seats are allocated in decreasing order of the fractional part $Sp_i - \lfloor Sp_i \rfloor$.
- By construction this always satisfies the quota axioms.
- There are other largest remainder methods, which use different formulae in place of the lower quota, such as the Droop quota.
- These methods fail the last 3 axioms above.

Divisor methods

- A strictly increasing sequence $d(s), s \geq 0$ is fixed. For each $s \leq S$ we compute all quotients $q_{is} := p_i/d(s)$. Seats are allocated in decreasing order of value of q_{is}.

Divisor methods

- A strictly increasing sequence $d(s), s \geq 0$ is fixed. For each $s \leq S$ we compute all quotients $q_{is} := p_i/d(s)$. Seats are allocated in decreasing order of value of q_{is}.
- The main examples are as follows (but any sequence can be used):

Divisor methods

- A strictly increasing sequence $d(s), s \geq 0$ is fixed. For each $s \leq S$ we compute all quotients $q_{is} := p_i/d(s)$. Seats are allocated in decreasing order of value of q_{is}.
- The main examples are as follows (but any sequence can be used):
    - (Jefferson-D'Hondt): $d(s) = s + 1$;

Divisor methods

- A strictly increasing sequence $d(s), s \geq 0$ is fixed. For each $s \leq S$ we compute all quotients $q_{is} := p_i/d(s)$. Seats are allocated in decreasing order of value of q_{is}.
- The main examples are as follows (but any sequence can be used):
  - (Jefferson-D'Hondt): $d(s) = s + 1$;
  - (Webster-St Lagüe): $d(s) = s + 1/2$;

Divisor methods

- A strictly increasing sequence $d(s), s \geq 0$ is fixed. For each $s \leq S$ we compute all quotients $q_{is} := p_i/d(s)$. Seats are allocated in decreasing order of value of q_{is}.
- The main examples are as follows (but any sequence can be used):
  - (Jefferson-D'Hondt): $d(s) = s + 1$;
  - (Webster-St Lagüe): $d(s) = s + 1/2$;
  - (Dean): $d(s) = s(s + 1)/(2s + 1)$;

Divisor methods

- A strictly increasing sequence $d(s), s \geq 0$ is fixed. For each $s \leq S$ we compute all quotients $q_{is} := p_i/d(s)$. Seats are allocated in decreasing order of value of q_{is}.
- The main examples are as follows (but any sequence can be used):
  - (Jefferson-D'Hondt): $d(s) = s + 1$;
  - (Webster-St Lagüe): $d(s) = s + 1/2$;
  - (Dean): $d(s) = s(s + 1)/(2s + 1)$;
  - (Huntington-Hill): $d(s) = \sqrt{s(s + 1)}$.

Divisor methods

- A strictly increasing sequence $d(s), s \geq 0$ is fixed. For each $s \leq S$ we compute all quotients $q_{is} := p_i/d(s)$. Seats are allocated in decreasing order of value of q_{is}.
- The main examples are as follows (but any sequence can be used):
  - (Jefferson-D'Hondt): $d(s) = s + 1$;
  - (Webster-St Lagüe): $d(s) = s + 1/2$;
  - (Dean): $d(s) = s(s+1)/(2s+1)$;
  - (Huntington-Hill): $d(s) = \sqrt{s(s+1)}$.
  - (Adams): $d(s) = s$.

Divisor methods

- A strictly increasing sequence $d(s), s \geq 0$ is fixed. For each $s \leq S$ we compute all quotients $q_{is} := p_i/d(s)$. Seats are allocated in decreasing order of value of q_{is}.
- The main examples are as follows (but any sequence can be used):
  - (Jefferson-D'Hondt): $d(s) = s + 1$;
  - (Webster-St Lagüe): $d(s) = s + 1/2$;
  - (Dean): $d(s) = s(s + 1)/(2s + 1)$;
  - (Huntington-Hill): $d(s) = \sqrt{s(s + 1)}$.
  - (Adams): $d(s) = s$.
- These can all be interpreted in terms of a specific way of rounding the fractional allocations Sp_i. For example, Jefferson always rounds down; Webster to the nearest integer, Dean using harmonic mean, Huntington using geometric mean, Adams up.

Divisor methods - alternative interpretation

- Start with the standard “price per seat” $D := P/S$, and adjust D until the rounded-off seat numbers add to the correct total.

Divisor methods - alternative interpretation

- Start with the standard “price per seat” $D := P/S$, and adjust D until the rounded-off seat numbers add to the correct total.
- There will be no change in seat numbers allocated unless P_i/D crosses the rounding boundary. This explains the divisor sequences above.

Divisor methods - alternative interpretation

- Start with the standard "price per seat" $D := P/S$, and adjust D until the rounded-off seat numbers add to the correct total.
- There will be no change in seat numbers allocated unless P_i/D crosses the rounding boundary. This explains the divisor sequences above.
- Divisor methods all violate at least one quota axiom, but satisfy all the other axioms. They are the only methods satisfying population monotonicity.

Examples

- Suppose agents a, b, c have weights $5, 3, 1$ and $N = 4$. Hamilton's method gives seat allocation $2, 1, 1$. If we change to $N = 5$, it then gives $3, 2, 0$ (Alabama paradox).

Examples

- Suppose agents a, b, c have weights $5, 3, 1$ and $N = 4$. Hamilton's method gives seat allocation $2, 1, 1$. If we change to $N = 5$, it then gives $3, 2, 0$ (Alabama paradox).
- Suppose agents a, b, c have weights $55, 29, 16$ and $N = 10$. The allocation under Hamilton is $5, 3, 2$. If weights change to $58, 28, 17$ then allocation changes to $6, 3, 1$.

Examples

- Suppose agents a, b, c have weights $5, 3, 1$ and $N = 4$. Hamilton's method gives seat allocation $2, 1, 1$. If we change to $N = 5$, it then gives $3, 2, 0$ (Alabama paradox).
- Suppose agents a, b, c have weights $55, 29, 16$ and $N = 10$. The allocation under Hamilton is $5, 3, 2$. If weights change to $58, 28, 17$ then allocation changes to $6, 3, 1$.
- Given divisor sequence $1 = d(0) < d(1) < \dots$, choose $k > d(N)/(N-1)$, have k agents with weight 1 and one agent with weight W, where $d(N) < W \leq k(N-1)$. Then the big agent wins all seats but its quota is at most $N-1$, so upper quota is violated.

Examples

- Suppose agents a, b, c have weights $5, 3, 1$ and $N = 4$. Hamilton's method gives seat allocation $2, 1, 1$. If we change to $N = 5$, it then gives $3, 2, 0$ (Alabama paradox).
- Suppose agents a, b, c have weights $55, 29, 16$ and $N = 10$. The allocation under Hamilton is $5, 3, 2$. If weights change to $58, 28, 17$ then allocation changes to $6, 3, 1$.
- Given divisor sequence $1 = d(0) < d(1) < \dots$, choose $k > d(N)/(N-1)$, have k agents with weight 1 and one agent with weight W, where $d(N) < W \leq k(N-1)$. Then the big agent wins all seats but its quota is at most $N-1$, so upper quota is violated.
- Lower quota can also be violated by divisor methods. In the above example if we have $W > N + 1 - k$ but $W < d(1)/d(0)$ then each small agent gets at least one seat and the big agent does not make its quota.

Welfare approach: proportionality

- An obvious idea is to try to minimize some global measure of disproportionality. For example, given the weights p_i and the seat fractions s_i given by an allocation algorithm, we could try to minimize, for example:

Welfare approach: proportionality

- An obvious idea is to try to minimize some global measure of disproportionality. For example, given the weights p_i and the seat fractions s_i given by an allocation algorithm, we could try to minimize, for example:
  - $\sum_i |s_i - p_i|$ (the ℓ^1-norm), or another norm such as ℓ^2, ℓ^∞;

Welfare approach: proportionality

- An obvious idea is to try to minimize some global measure of disproportionality. For example, given the weights p_i and the seat fractions s_i given by an allocation algorithm, we could try to minimize, for example:
  - $\sum_i |s_i - p_i|$ (the ℓ^1-norm), or another norm such as ℓ^2, ℓ^∞;
  - $\sum_i (s_i - p_i)^2 / p_i$;

Welfare approach: proportionality

- An obvious idea is to try to minimize some global measure of disproportionality. For example, given the weights p_i and the seat fractions s_i given by an allocation algorithm, we could try to minimize, for example:
  - $\sum_i |s_i - p_i|$ (the ℓ^1-norm), or another norm such as ℓ^2, ℓ^∞;
  - $\sum_i (s_i - p_i)^2 / p_i$;
  - $\sum_i (s_i - p_i)^2 / s_i$;

Welfare approach: proportionality

- An obvious idea is to try to minimize some global measure of disproportionality. For example, given the weights p_i and the seat fractions s_i given by an allocation algorithm, we could try to minimize, for example:
  - $\sum_i |s_i - p_i|$ (the ℓ^1-norm), or another norm such as ℓ^2, ℓ^∞;
  - $\sum_i (s_i - p_i)^2 / p_i$;
  - $\sum_i (s_i - p_i)^2 / s_i$;
  - $\max_i s_i / p_i$;

Welfare approach: proportionality

- An obvious idea is to try to minimize some global measure of disproportionality. For example, given the weights p_i and the seat fractions s_i given by an allocation algorithm, we could try to minimize, for example:
  - $\sum_i |s_i - p_i|$ (the ℓ^1-norm), or another norm such as ℓ^2, ℓ^∞;
  - $\sum_i (s_i - p_i)^2 / p_i$;
  - $\sum_i (s_i - p_i)^2 / s_i$;
  - $\max_i s_i / p_i$;
  - $\max_i p_i / s_i$;

Welfare approach: proportionality

- An obvious idea is to try to minimize some global measure of disproportionality. For example, given the weights p_i and the seat fractions s_i given by an allocation algorithm, we could try to minimize, for example:
  - $\sum_i |s_i - p_i|$ (the ℓ^1-norm), or another norm such as ℓ^2, ℓ^∞;
  - $\sum_i (s_i - p_i)^2/p_i$;
  - $\sum_i (s_i - p_i)^2/s_i$;
  - $\max_i s_i/p_i$;
  - $\max_i p_i/s_i$;
- These measures turn out to be minimized by the Hamilton, Webster, Huntington, Jefferson and Adams methods respectively.

Geometric interpretation

- Consider the vectors x, y whose ith components are $p_i S$ (ideal seat quota) and S_i (actual number of seats).

Geometric interpretation

- Consider the vectors x, y whose ith components are $p_i S$ (ideal seat quota) and S_i (actual number of seats).
- Hamilton's method minimizes the distance from y to x.

Geometric interpretation

- Consider the vectors x, y whose ith components are $p_i S$ (ideal seat quota) and S_i (actual number of seats).
- Hamilton's method minimizes the distance from y to x.
- Webster's method minimizes the distance from y to the line through x and the origin.

Geometric interpretation

- Consider the vectors x, y whose ith components are $p_i S$ (ideal seat quota) and S_i (actual number of seats).
- Hamilton's method minimizes the distance from y to x.
- Webster's method minimizes the distance from y to the line through x and the origin.
- Koppel & Diskin (2008) listed several axioms for measures of disproportionality and found that the cosine measure (defined as $1 - c$ where c is the cosine of the angle between x and y) satisfies them all, while the measures above did not. This is equivalent to the distance between the normalized vectors $x/||x||_2$ and $y/||y||_2$.

Geometric interpretation

- Consider the vectors x, y whose ith components are $p_i S$ (ideal seat quota) and S_i (actual number of seats).
- Hamilton's method minimizes the distance from y to x.
- Webster's method minimizes the distance from y to the line through x and the origin.
- Koppel & Diskin (2008) listed several axioms for measures of disproportionality and found that the cosine measure (defined as $1 - c$ where c is the cosine of the angle between x and y) satisfies them all, while the measures above did not. This is equivalent to the distance between the normalized vectors $x/||x||_2$ and $y/||y||_2$.
- Is there a nice algorithm that minimises the cosine measure?

Bias

- A basic fairness idea is freedom from bias: an apportionment method should not systematically favour large (or small) agents.

Bias

- A basic fairness idea is freedom from bias: an apportionment method should not systematically favour large (or small) agents.
- Jefferson is biased in favour of large agents. Hamilton and Webster are not. Adams is biased in favour of small agents.

Bias

- A basic fairness idea is freedom from bias: an apportionment method should not systematically favour large (or small) agents.
- Jefferson is biased in favour of large agents. Hamilton and Webster are not. Adams is biased in favour of small agents.
- For fixed n Huntington and Dean have a bias in favour of small agents because their rounding cutoff is less than halfway to the next integer. As $n \to \infty$ this bias tends to zero. The bias of Jefferson and Adams does not disappear in this way.

Overview

Unlike strategic games, here we focus on the idea that a group can achieve a good result (e.g. a monetary payoff) by working together, and we want to allocate the gains to the members in a way that respects their different contributions.

Coalitional games: standard form

- A cooperative game (in characteristic function form) is given by a set N of players and a function $v : 2^N \to \mathbb{R}$ such that $v(\emptyset) = 0$. We will write $n = |N|$.

Coalitional games: standard form

- A cooperative game (in characteristic function form) is given by a set N of players and a function $v : 2^N \to \mathbb{R}$ such that $v(\emptyset) = 0$. We will write $n = |N|$.
- The game is a TU-game (Transferable Utility) if every possible division of the value $v(C)$ between members of C is possible.

Small concrete running example

- Three players A, B, C want to buy icecream. They can contribute $3, 4, 5$ dollars respectively.

Small concrete running example

- Three players A, B, C want to buy icecream. They can contribute $3, 4, 5$ dollars respectively.
- The players care only about icecream and not money. They enjoy icecream equally and their enjoyment is proportional to the amount they receive.

Small concrete running example

- Three players A, B, C want to buy icecream. They can contribute $3, 4, 5$ dollars respectively.
- The players care only about icecream and not money. They enjoy icecream equally and their enjoyment is proportional to the amount they receive.
- They can buy a 500g tub of icecream for \$7, a 750g tub for \$9, or a 1000g tub for \$11.

Small concrete running example

- Three players A, B, C want to buy icecream. They can contribute $3, 4, 5$ dollars respectively.
- The players care only about icecream and not money. They enjoy icecream equally and their enjoyment is proportional to the amount they receive.
- They can buy a 500g tub of icecream for \$7, a 750g tub for \$9, or a 1000g tub for \$11.
- For example, the payoff to $\{A, C\}$ is 500, while the payoff to $\{B, C\}$ is 750.

Coalitional games: examples

- Unanimity games: fix a subset $S \subseteq N$. Then $v(T) = 1$ if and only if $T \supseteq S$ and $v(T) = 0$ otherwise.

Coalitional games: examples

- Unanimity games: fix a subset $S \subseteq N$. Then $v(T) = 1$ if and only if $T \supseteq S$ and $v(T) = 0$ otherwise.
- Weighted voting in kleptocracy: each of n players has a weight $w_i \in \mathbb{N}$. Fix $q \in (0, 1]$. A coalition T gets payoff $\$10^9$ if $\sum_{i \in T} x_i \geq q$ and 0 otherwise.

Coalitional games: examples

- Unanimity games: fix a subset $S \subseteq N$. Then $v(T) = 1$ if and only if $T \supseteq S$ and $v(T) = 0$ otherwise.
- Weighted voting in kleptocracy: each of n players has a weight $w_i \in \mathbb{N}$. Fix $q \in (0, 1]$. A coalition T gets payoff $\$10^9$ if $\sum_{i \in T} x_i \geq q$ and 0 otherwise.
- Cost sharing: each agent requires some amount of infrastructure (e.g. airport runways, electricity transmission lines) and all must contribute to building it.

Coalitional games: examples

- Unanimity games: fix a subset $S \subseteq N$. Then $v(T) = 1$ if and only if $T \supseteq S$ and $v(T) = 0$ otherwise.
- Weighted voting in kleptocracy: each of n players has a weight $w_i \in \mathbb{N}$. Fix $q \in (0, 1]$. A coalition T gets payoff $\$10^9$ if $\sum_{i \in T} x_i \geq q$ and 0 otherwise.
- Cost sharing: each agent requires some amount of infrastructure (e.g. airport runways, electricity transmission lines) and all must contribute to building it.
- Simple games: the payoff to each coalition is either 0 or 1.

Coalitional games: examples

- Unanimity games: fix a subset $S \subseteq N$. Then $v(T) = 1$ if and only if $T \supseteq S$ and $v(T) = 0$ otherwise.
- Weighted voting in kleptocracy: each of n players has a weight $w_i \in \mathbb{N}$. Fix $q \in (0, 1]$. A coalition T gets payoff $\$10^9$ if $\sum_{i \in T} x_i \geq q$ and 0 otherwise.
- Cost sharing: each agent requires some amount of infrastructure (e.g. airport runways, electricity transmission lines) and all must contribute to building it.
- Simple games: the payoff to each coalition is either 0 or 1.
- Machine learning: for example the payoff is some kind of accuracy score and the players are the features (predictor variables).

Properties of coalitional games

- Monotonicity: if $S \subseteq T \subseteq N$ then $v(S) \leq v(T) \leq v(N)$.
  "Larger coalitions are more valuable".

Properties of coalitional games

- Monotonicity: if $S \subseteq T \subseteq N$ then $v(S) \leq v(T) \leq v(N)$. "Larger coalitions are more valuable".
- Additivity: $v(S \cup T) = v(S) + v(T)$ if $S \cap T = \emptyset$. "Forming coalitions is pointless".

Properties of coalitional games

- Monotonicity: if $S \subseteq T \subseteq N$ then $v(S) \leq v(T) \leq v(N)$. "Larger coalitions are more valuable".
- Additivity: $v(S \cup T) = v(S) + v(T)$ if $S \cap T = \emptyset$. "Forming coalitions is pointless".
- Superadditivity: $v(S \cup T) \geq v(S) + v(T)$ if $S \cap T = \emptyset$. "Forming coalitions increases value".

Properties of coalitional games

- Monotonicity: if $S \subseteq T \subseteq N$ then $v(S) \leq v(T) \leq v(N)$. "Larger coalitions are more valuable".
- Additivity: $v(S \cup T) = v(S) + v(T)$ if $S \cap T = \emptyset$. "Forming coalitions is pointless".
- Superadditivity: $v(S \cup T) \geq v(S) + v(T)$ if $S \cap T = \emptyset$. "Forming coalitions increases value".
- Convexity: $v(S \cup T) + v(S \cap T) \geq v(S) + v(T)$. "Increasing returns to scale".

Shapley value

- An allocation is a function mapping each subset of N to a payoff vector (where component i gives the amount allocated to player i).

Shapley value

- An allocation is a function mapping each subset of N to a payoff vector (where component i gives the amount allocated to player i).
- A basic principle is that a player's allocation should depend on its marginal contribution to the coalition.

Shapley value

- An allocation is a function mapping each subset of N to a payoff vector (where component i gives the amount allocated to player i).
- A basic principle is that a player's allocation should depend on its marginal contribution to the coalition.
- Shapley (1953) devised the following allocation. Suppose that players join a coalition one at a time. Pay each of them the marginal value that they contribute to the coalition. This depends strongly on the order of players, so average over all permutations of N.

Shapley value

- An allocation is a function mapping each subset of N to a payoff vector (where component i gives the amount allocated to player i).
- A basic principle is that a player's allocation should depend on its marginal contribution to the coalition.
- Shapley (1953) devised the following allocation. Suppose that players join a coalition one at a time. Pay each of them the marginal value that they contribute to the coalition. This depends strongly on the order of players, so average over all permutations of N.
- The Shapley value gives player i the above payoff.

Formulae for Shapley value

$$\begin{aligned}\phi_i(v) &= \sum_{k=0}^{n-1} \frac{1}{n\binom{n-1}{k}} \sum_{|S|=k} \left[v(S \cup \{i\}) - v(S)\right] \\ &= \sum_{k=1}^{n} \frac{1}{k\binom{n}{k}} \sum_{|S|=k} \left[v(S) - v(S \setminus \{i\})\right].\end{aligned}$$

Power indices

- If we restrict the Shapley value to the class of simple games, we obtain the Shapley-Shubik power index. The idea is that this measures the power of each player.

Power indices

- If we restrict the Shapley value to the class of simple games, we obtain the Shapley-Shubik power index. The idea is that this measures the power of each player.
- Another power index is the Penrose-Banzhaf index, given by

$$\beta_i = 2^{-n-1} \sum_{S \subseteq N} \left[v(S \cup \{i\}) - v(S) \right].$$

Power indices

- If we restrict the Shapley value to the class of simple games, we obtain the Shapley-Shubik power index. The idea is that this measures the power of each player.
- Another power index is the Penrose-Banzhaf index, given by

$$\beta_i = 2^{-n-1} \sum_{S \subseteq N} \left[v(S \cup \{i\}) - v(S) \right].$$

- Here the idea is that every coalition occurs with equal probability. For example, in yes-no voting, every possible configuration of yes/no votes is equally likely to occur.

Properties of allocations and power indices

- Anonymity: $\psi_{\pi i}(\pi v) = \psi_i(v)$. Players' identities are irrelevant, only their role in coalitions matters.

Properties of allocations and power indices

- Anonymity: $\psi_{\pi i}(\pi v) = \psi_i(v)$. Players' identities are irrelevant, only their role in coalitions matters.
- Additivity: $\psi_i(v + w) = \psi_i(v) + \psi_i(w)$.

Properties of allocations and power indices

- Anonymity: $\psi_{\pi i}(\pi v) = \psi_i(v)$. Players' identities are irrelevant, only their role in coalitions matters.
- Additivity: $\psi_i(v + w) = \psi_i(v) + \psi_i(w)$.
- Null Player Property: $\psi_i(v) = 0$ if i is a null player (it contributes zero value to every coalition it joins).

Properties of allocations and power indices

- Anonymity: $\psi_{\pi i}(\pi v) = \psi_i(v)$. Players' identities are irrelevant, only their role in coalitions matters.
- Additivity: $\psi_i(v + w) = \psi_i(v) + \psi_i(w)$.
- Null Player Property: $\psi_i(v) = 0$ if i is a null player (it contributes zero value to every coalition it joins).
- Efficiency: $\sum_i \psi_i(v) = v(N)$.

Axiomatic results

- An allocation satisfies Anonymity, Additivity, Null Player Property and Efficiency if and only if it is the Shapley value.

Axiomatic results

- An allocation satisfies Anonymity, Additivity, Null Player Property and Efficiency if and only if it is the Shapley value.
- Anonymity and Null Player Property determine the allocation for unanimity games up to a constant factor, Efficiency determines the factor, and Additivity then gives the result for all games.

Axiomatic results

- An allocation satisfies Anonymity, Additivity, Null Player Property and Efficiency if and only if it is the Shapley value.
- Anonymity and Null Player Property determine the allocation for unanimity games up to a constant factor, Efficiency determines the factor, and Additivity then gives the result for all games.
- The Penrose-Banzhaf allocation satisfies Anonymity, Additivity, Null Player Property.

The core

- The allocation given by, say, the Shapley value may fail to be stable. That is, some players may have incentive to form smaller coalitions and divide the gains themselves.

The core

- The allocation given by, say, the Shapley value may fail to be stable. That is, some players may have incentive to form smaller coalitions and divide the gains themselves.
- The core is the set of all efficient allocations ψ such that for all $S \subseteq N$, $\sum_i \psi_i(v) \geq v(S)$.

The core

- The allocation given by, say, the Shapley value may fail to be stable. That is, some players may have incentive to form smaller coalitions and divide the gains themselves.
- The core is the set of all efficient allocations ψ such that for all $S \subseteq N$, $\sum_i \psi_i(v) \geq v(S)$.
- The core can be empty, but for convex games it always contains the Shapley allocation.

The core

- The allocation given by, say, the Shapley value may fail to be stable. That is, some players may have incentive to form smaller coalitions and divide the gains themselves.
- The core is the set of all efficient allocations ψ such that for all $S \subseteq N$, $\sum_i \psi_i(v) \geq v(S)$.
- The core can be empty, but for convex games it always contains the Shapley allocation.
- Note that an allocation in the core must maximize the sum of payoffs of players over all allocations.

The core

- The allocation given by, say, the Shapley value may fail to be stable. That is, some players may have incentive to form smaller coalitions and divide the gains themselves.
- The core is the set of all efficient allocations ψ such that for all $S \subseteq N$, $\sum_i \psi_i(v) \geq v(S)$.
- The core can be empty, but for convex games it always contains the Shapley allocation.
- Note that an allocation in the core must maximize the sum of payoffs of players over all allocations.
- The elements of the core can be described as feasible solutions to a linear programming problem.

Overview

In many situations we deal with agents who have conflicting preferences over outcomes. Each seeks to obtain a more preferred outcome, but must deal with the actions and preferences of the other players. This is a huge subject and we only consider a small part. We assume that players have *common knowledge* — each player knows the preferences and payoffs of all players, all players know that all players know, ...

Simultaneous games

- There is a finite set of players.

Simultaneous games

- There is a finite set of players.
- Each player has a set of pure strategies or actions from which to choose. They can choose these with probabilities, giving mixed strategies.

Simultaneous games

- There is a finite set of players.
- Each player has a set of pure strategies or actions from which to choose. They can choose these with probabilities, giving mixed strategies.
- Each choice of strategy by all players is a strategy profile which yields an outcome. Each strategy profile yields a payoff (utility) to player. Larger payoffs (to itself) are preferred by each player.

Simultaneous games

- There is a finite set of players.
- Each player has a set of pure strategies or actions from which to choose. They can choose these with probabilities, giving mixed strategies.
- Each choice of strategy by all players is a strategy profile which yields an outcome. Each strategy profile yields a payoff (utility) to player. Larger payoffs (to itself) are preferred by each player.
- Players move simultaneously (each player must move before gaining information about the moves of other players).

Simultaneous games

- There is a finite set of players.
- Each player has a set of pure strategies or actions from which to choose. They can choose these with probabilities, giving mixed strategies.
- Each choice of strategy by all players is a strategy profile which yields an outcome. Each strategy profile yields a payoff (utility) to player. Larger payoffs (to itself) are preferred by each player.
- Players move simultaneously (each player must move before gaining information about the moves of other players).
- We use plurality voting as a running example: there are finitely many candidates and each player can vote for exactly one of them.

Example — Prisoners' Dilemma

- This is a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.

Example — Prisoners' Dilemma

- This is a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_1 to o_4. Player 1 prefers o_2 to o_1 to o_4 to o_3, while player 2 prefers o_3 to o_1 to o_4 to o_2.

Example — Prisoners' Dilemma

- This is a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_1 to o_4. Player 1 prefers o_2 to o_1 to o_4 to o_3, while player 2 prefers o_3 to o_1 to o_4 to o_2.
- If player 1 plays a, player 2 "should" play b to get the better outcome, while if player 1 plays b, player 2 should also play b.

Example — Prisoners' Dilemma

- This is a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_1 to o_4. Player 1 prefers o_2 to o_1 to o_4 to o_3, while player 2 prefers o_3 to o_1 to o_4 to o_2.
- If player 1 plays a, player 2 "should" play b to get the better outcome, while if player 1 plays b, player 2 should also play b.
- However Player 1 has the same computation. Thus both players play (b, b), which is worse for them both than (a, a).

Example — BoS

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.

Example — BoS

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_1 and o_4 to o_2 and o_3. Player 1 prefers o_1 to o_4, while player 2 prefers o_4 to o_1.

Example — BoS

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_1 and o_4 to o_2 and o_3. Player 1 prefers o_1 to o_4, while player 2 prefers o_4 to o_1.
- The outcomes relate to the strategies as before:

Example — BoS

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_1 and o_4 to o_2 and o_3. Player 1 prefers o_1 to o_4, while player 2 prefers o_4 to o_1.
- The outcomes relate to the strategies as before:
- If player 1 plays a, player 2 should play a to get the better outcome, while if player 1 plays b, player 2 should also play b.

Example — BoS

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_1 and o_4 to o_2 and o_3. Player 1 prefers o_1 to o_4, while player 2 prefers o_4 to o_1.
- The outcomes relate to the strategies as before:
- If player 1 plays a, player 2 should play a to get the better outcome, while if player 1 plays b, player 2 should also play b.
- However Player 1 has the same computation.

Example — BoS

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_1 and o_4 to o_2 and o_3. Player 1 prefers o_1 to o_4, while player 2 prefers o_4 to o_1.
- The outcomes relate to the strategies as before:
- If player 1 plays a, player 2 should play a to get the better outcome, while if player 1 plays b, player 2 should also play b.
- However Player 1 has the same computation.
- This can clearly occur as a simultaneous voting game. For example o_1 is "a wins" and o_4 is "b wins", while o_2, o_3 are "a and b win".

Example — chicken

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.

Example — chicken

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_2 and o_3 to o_1 and o_4, and each player is indifferent between o_1 and o_4.

Example — chicken

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_2 and o_3 to o_1 and o_4, and each player is indifferent between o_1 and o_4.
- The outcomes relate to the strategies as before:

Example — chicken

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_2 and o_3 to o_1 and o_4, and each player is indifferent between o_1 and o_4.
- The outcomes relate to the strategies as before:
- If player 1 plays a, player 2 should play b to get the better outcome, while if player 1 plays b, player 2 should play a.

Example — chicken

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_2 and o_3 to o_1 and o_4, and each player is indifferent between o_1 and o_4.
- The outcomes relate to the strategies as before:
- If player 1 plays a, player 2 should play b to get the better outcome, while if player 1 plays b, player 2 should play a.
- However Player 1 has the same computation. What should they do?

Example — stag hunt

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.

Example — stag hunt

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_1 to all other outcomes. Player 1 prefers o_2 to o_4 to o_3 and Player 2 prefers o_3 to o_4 to o_2.

Example — stag hunt

- This a simultaneous game with 2 players, each of which has (the same) two strategies a and b. There are 4 possible outcomes $o_1 = (a, a), o_2 = (b, a), o_3 = (a, b), o_4 = (b, b)$.
- Each player prefers o_1 to all other outcomes. Player 1 prefers o_2 to o_4 to o_3 and Player 2 prefers o_3 to o_4 to o_2.
- Note the similarities and differences to Prisoners' Dilemma.

Strategic domination

- For each strategy profile S and player i, let S_{-i} denote the profile with i deleted (i.e. the strategies of all other players are specified).

Strategic domination

- For each strategy profile S and player i, let S_{-i} denote the profile with i deleted (i.e. the strategies of all other players are specified).
- The strategy s weakly dominates the strategy t if for all S_{-i}, the payoff to i obtained by playing s is at least as great as the payoff obtained by playing t, *and* there is at least one S_{-i} for which the payoff is strictly greater. If the payoff is always strictly greater, s strongly dominates t.

Strategic domination

- For each strategy profile S and player i, let S_{-i} denote the profile with i deleted (i.e. the strategies of all other players are specified).
- The strategy s weakly dominates the strategy t if for all S_{-i}, the payoff to i obtained by playing s is at least as great as the payoff obtained by playing t, *and* there is at least one S_{-i} for which the payoff is strictly greater. If the payoff is always strictly greater, s strongly dominates t.
- This gives a partial order on strategies.

Strategic domination

- For each strategy profile S and player i, let S_{-i} denote the profile with i deleted (i.e. the strategies of all other players are specified).
- The strategy s weakly dominates the strategy t if for all S_{-i}, the payoff to i obtained by playing s is at least as great as the payoff obtained by playing t, *and* there is at least one S_{-i} for which the payoff is strictly greater. If the payoff is always strictly greater, s strongly dominates t.
- This gives a partial order on strategies.
- A dominant strategy is one that weakly dominates all other strategies (a maximum element of the partial order). It need not exist (does for Prisoners' Dilemma, not for BoS or chicken).

Strategic domination

- For each strategy profile S and player i, let S_{-i} denote the profile with i deleted (i.e. the strategies of all other players are specified).
- The strategy s weakly dominates the strategy t if for all S_{-i}, the payoff to i obtained by playing s is at least as great as the payoff obtained by playing t, *and* there is at least one S_{-i} for which the payoff is strictly greater. If the payoff is always strictly greater, s strongly dominates t.
- This gives a partial order on strategies.
- A dominant strategy is one that weakly dominates all other strategies (a maximum element of the partial order). It need not exist (does for Prisoners' Dilemma, not for BoS or chicken).
- Choosing a weakly dominated strategy is not something we expect from a rational player.

Best reply

- A best reply for player i given profile S is a strategy s that maximizes the payoff to i, given that other players are playing S_{-i}.

Best reply

- A best reply for player i given profile S is a strategy s that maximizes the payoff to i, given that other players are playing S_{-i}.
- Example: suppose all voters except i vote sincerely. The Gibbard-Satterthwaite Theorem (1973) says that for almost every voting rule and some distribution of preferences, voting sincerely is not a best reply for some player.

Best reply

- A best reply for player i given profile S is a strategy s that maximizes the payoff to i, given that other players are playing S_{-i}.
- Example: suppose all voters except i vote sincerely. The Gibbard-Satterthwaite Theorem (1973) says that for almost every voting rule and some distribution of preferences, voting sincerely is not a best reply for some player.
- Example: suppose we use plurality voting with 3 candidates and 10 voters, and 9 of the voters vote for a. Then every vote by the other player yields the same outcome (a wins), and so every move is a best reply.

Best reply

- A best reply for player i given profile S is a strategy s that maximizes the payoff to i, given that other players are playing S_{-i}.
- Example: suppose all voters except i vote sincerely. The Gibbard-Satterthwaite Theorem (1973) says that for almost every voting rule and some distribution of preferences, voting sincerely is not a best reply for some player.
- Example: suppose we use plurality voting with 3 candidates and 10 voters, and 9 of the voters vote for a. Then every vote by the other player yields the same outcome (a wins), and so every move is a best reply.
- A dominant strategy is a strategy that is a best reply for player i for every S_{-i}.

Best reply

- A best reply for player i given profile S is a strategy s that maximizes the payoff to i, given that other players are playing S_{-i}.
- Example: suppose all voters except i vote sincerely. The Gibbard-Satterthwaite Theorem (1973) says that for almost every voting rule and some distribution of preferences, voting sincerely is not a best reply for some player.
- Example: suppose we use plurality voting with 3 candidates and 10 voters, and 9 of the voters vote for a. Then every vote by the other player yields the same outcome (a wins), and so every move is a best reply.
- A dominant strategy is a strategy that is a best reply for player i for every S_{-i}.
- A weakly dominated strategy is never the unique best reply, but it may be a best reply sometimes.

Solution concept – dominance solvability

- We can remove weakly dominated strategies from a game, assuming that all players are rational.

Solution concept – dominance solvability

- We can remove weakly dominated strategies from a game, assuming that all players are rational.
- The reduced game may now have dominated strategies that were not dominated before, so we iterate. Note that we assume that rationality of all players is common knowledge to all players, as are all the preferences.

Solution concept – dominance solvability

- We can remove weakly dominated strategies from a game, assuming that all players are rational.
- The reduced game may now have dominated strategies that were not dominated before, so we iterate. Note that we assume that rationality of all players is common knowledge to all players, as are all the preferences.
- This process ends after finitely many steps. If we remove only strongly dominated strategies, then it turns out not to matter in what order we remove them. Games that reduce in this way to a single action by each player are called dominance solvable and there is an obvious prediction for what rational players will do, and hence for the outcome of the game.

Example – dominance solvability

- Consider a plurality voting game where 2 players have preference order abc, 1 has order bac, and 1 has order bca.

Example – dominance solvability

- Consider a plurality voting game where 2 players have preference order abc, 1 has order bac, and 1 has order bca.
- Abstention is weakly dominated, and voting for the least preferred candidate is also weakly dominated.

Example – dominance solvability

- Consider a plurality voting game where 2 players have preference order abc, 1 has order bac, and 1 has order bca.
- Abstention is weakly dominated, and voting for the least preferred candidate is also weakly dominated.
- Thus the first 3 voters can eliminate c as an action, and choose between a and b only. The 4th eliminates voting for a.

Example – dominance solvability

- Consider a plurality voting game where 2 players have preference order abc, 1 has order bac, and 1 has order bca.
- Abstention is weakly dominated, and voting for the least preferred candidate is also weakly dominated.
- Thus the first 3 voters can eliminate c as an action, and choose between a and b only. The 4th eliminates voting for a.
- The 4th voter can now reason that c cannot win in any situation, and so voting for c is dominated by voting for b.

Example – dominance solvability

- Consider a plurality voting game where 2 players have preference order abc, 1 has order bac, and 1 has order bca.
- Abstention is weakly dominated, and voting for the least preferred candidate is also weakly dominated.
- Thus the first 3 voters can eliminate c as an action, and choose between a and b only. The 4th eliminates voting for a.
- The 4th voter can now reason that c cannot win in any situation, and so voting for c is dominated by voting for b.
- Assuming that the first two voters prefer the winner set $\{a\}$ to $\{a, b\}$ to $\{b\}$ (and the analogous assumption for the third voter), voting for the second ranked candidate is dominated by voting for the first.

Example – dominance solvability

- Consider a plurality voting game where 2 players have preference order abc, 1 has order bac, and 1 has order bca.
- Abstention is weakly dominated, and voting for the least preferred candidate is also weakly dominated.
- Thus the first 3 voters can eliminate c as an action, and choose between a and b only. The 4th eliminates voting for a.
- The 4th voter can now reason that c cannot win in any situation, and so voting for c is dominated by voting for b.
- Assuming that the first two voters prefer the winner set $\{a\}$ to $\{a, b\}$ to $\{b\}$ (and the analogous assumption for the third voter), voting for the second ranked candidate is dominated by voting for the first.
- Thus the outcome will be $\{a, b\}$. In this case, it is the same as the sincere outcome.

Example – chair's paradox

- Consider a plurality voting game where players 1,2, 3 have preferences abc, bca, cab respectively.

Example – chair's paradox

- Consider a plurality voting game where players 1,2, 3 have preferences abc, bca, cab respectively.
- The tiebreaking rule is that ties are resolved according to the preferences of player 1.

Example – chair's paradox

- Consider a plurality voting game where players 1,2, 3 have preferences abc, bca, cab respectively.
- The tiebreaking rule is that ties are resolved according to the preferences of player 1.
- The sincere outcome is that a wins on tiebreak.

Example – chair's paradox

- Consider a plurality voting game where players 1,2, 3 have preferences abc, bca, cab respectively.
- The tiebreaking rule is that ties are resolved according to the preferences of player 1.
- The sincere outcome is that a wins on tiebreak.
- For player 1, voting b or c is weakly dominated by voting a. Then for player 3, voting c weakly dominates both a and b. After these reductions, player 2 votes c and c wins.

Example – chair's paradox

- Consider a plurality voting game where players 1,2, 3 have preferences abc, bca, cab respectively.
- The tiebreaking rule is that ties are resolved according to the preferences of player 1.
- The sincere outcome is that a wins on tiebreak.
- For player 1, voting b or c is weakly dominated by voting a. Then for player 3, voting c weakly dominates both a and b. After these reductions, player 2 votes c and c wins.
- Note that c is the worst alternative for player 1, yet she seems to have more power.

Example – chair's paradox

- Consider a plurality voting game where players 1,2, 3 have preferences abc, bca, cab respectively.
- The tiebreaking rule is that ties are resolved according to the preferences of player 1.
- The sincere outcome is that a wins on tiebreak.
- For player 1, voting b or c is weakly dominated by voting a. Then for player 3, voting c weakly dominates both a and b. After these reductions, player 2 votes c and c wins.
- Note that c is the worst alternative for player 1, yet she seems to have more power.
- However if player 1 gives up the right to cast the tiebreaking vote, she may do better. For example, if player 1 announces that she will not vote for a then b will be elected.

Solution concept – Nash equilibrium

- Suppose that every player plays best reply to the strategies of the other players. Then unilaterally deviating is a (weakly) dominated strategy for each player.

Solution concept – Nash equilibrium

- Suppose that every player plays best reply to the strategies of the other players. Then unilaterally deviating is a (weakly) dominated strategy for each player.
- Such a strategy profile is called a Nash equilibrium. It is pure if every player plays a pure strategy (no randomization).

Solution concept – Nash equilibrium

- Suppose that every player plays best reply to the strategies of the other players. Then unilaterally deviating is a (weakly) dominated strategy for each player.
- Such a strategy profile is called a Nash equilibrium. It is pure if every player plays a pure strategy (no randomization).
- Prisoner's Dilemma has a unique NE, namely (b, b). BoS has two pure NE, namely (a, a) and (b, b). There is also another, mixed, NE if we assign utilities to each player consistent with their preferences. How do we find it?

Mixed Nash equilibrium computation

- In BoS, suppose player 1 plays a with probability p while player 2 plays a with probability q. Assume $0 < p, q < 1$.

Mixed Nash equilibrium computation

- In BoS, suppose player 1 plays a with probability p while player 2 plays a with probability q. Assume $0 < p, q < 1$.
- Suppose that player 1 gets utility u_i from outcome i while player 2 gets v_i. Note that $u_1 > u_4 > u_2, u_3$ and $v_4 > v_1 > v_2, v_3$.

Mixed Nash equilibrium computation

- In BoS, suppose player 1 plays a with probability p while player 2 plays a with probability q. Assume $0 < p, q < 1$.
- Suppose that player 1 gets utility u_i from outcome i while player 2 gets v_i. Note that $u_1 > u_4 > u_2, u_3$ and $v_4 > v_1 > v_2, v_3$.
- The expected payoff to player 1 is $pqu_1 + p(1-q)u_2 + (1-p)qu_3 + (1-p)(1-q)u_4$, and to player 2 it is $pqv_1 + (1-q)pv_2 + (1-p)qv_3 + (1-p)(1-q)v_4$.

Mixed Nash equilibrium computation

- In BoS, suppose player 1 plays a with probability p while player 2 plays a with probability q. Assume $0 < p, q < 1$.
- Suppose that player 1 gets utility u_i from outcome i while player 2 gets v_i. Note that $u_1 > u_4 > u_2, u_3$ and $v_4 > v_1 > v_2, v_3$.
- The expected payoff to player 1 is $pqu_1 + p(1-q)u_2 + (1-p)qu_3 + (1-p)(1-q)u_4$, and to player 2 it is $pqv_1 + (1-q)pv_2 + (1-p)qv_3 + (1-p)(1-q)v_4$.
- Choose p to maximize player 1's payoff given q. This gives an equation determining q. Then maximize the second player's payoff, which determines p. Result: $p = (v_4 - v_2)/[(v_4 - v_2) + (v_1 - v_3)]$, and similarly for q.

Mixed Nash equilibrium computation

- In BoS, suppose player 1 plays a with probability p while player 2 plays a with probability q. Assume $0 < p, q < 1$.
- Suppose that player 1 gets utility u_i from outcome i while player 2 gets v_i. Note that $u_1 > u_4 > u_2, u_3$ and $v_4 > v_1 > v_2, v_3$.
- The expected payoff to player 1 is $pqu_1 + p(1-q)u_2 + (1-p)qu_3 + (1-p)(1-q)u_4$, and to player 2 it is $pqv_1 + (1-q)pv_2 + (1-p)qv_3 + (1-p)(1-q)v_4$.
- Choose p to maximize player 1's payoff given q. This gives an equation determining q. Then maximize the second player's payoff, which determines p. Result: $p = (v_4 - v_2)/[(v_4 - v_2) + (v_1 - v_3)]$, and similarly for q.
- Unfortunately this procedure leads to simultaneous nonlinear algebraic equations in general.

Basic facts about Nash equilibria

- NE always exist in finite games, but pure NE need not exist (e.g. rock-paper-scissors). Proved by Nash (1950) using a "fixed point theorem".

Basic facts about Nash equilibria

- NE always exist in finite games, but pure NE need not exist (e.g. rock-paper-scissors). Proved by Nash (1950) using a "fixed point theorem".
- There can be many NE, some of which Pareto-dominate others. Voting games give a good example. Suppose all voters have the same preference order and all vote for their least preferred candidate. Under plurality, for example, this is a NE.

Basic facts about Nash equilibria

- NE always exist in finite games, but pure NE need not exist (e.g. rock-paper-scissors). Proved by Nash (1950) using a "fixed point theorem".
- There can be many NE, some of which Pareto-dominate others. Voting games give a good example. Suppose all voters have the same preference order and all vote for their least preferred candidate. Under plurality, for example, this is a NE.
- Finding one NE of a game is a hard computational problem in general.

Mixed Nash equilibrium computation - special case

- For small games the following procedure can be useful. We show by example using BoS.

Mixed Nash equilibrium computation - special case

- For small games the following procedure can be useful. We show by example using BoS.
- Suppose both players play a mixed strategy that is not pure. Then given player 1's strategy, in order for player 2's to be a best reply, both pure strategies must yield the same expected payoff.

Mixed Nash equilibrium computation - special case

- For small games the following procedure can be useful. We show by example using BoS.
- Suppose both players play a mixed strategy that is not pure. Then given player 1's strategy, in order for player 2's to be a best reply, both pure strategies must yield the same expected payoff.
- Thus we have: $pv_3 + (1-p)v_4 = pv_1 + (1-p)v_2$, which yields the same result as before.

Mixed Nash equilibrium computation - special case

- For small games the following procedure can be useful. We show by example using BoS.
- Suppose both players play a mixed strategy that is not pure. Then given player 1's strategy, in order for player 2's to be a best reply, both pure strategies must yield the same expected payoff.
- Thus we have: $pv_3 + (1-p)v_4 = pv_1 + (1-p)v_2$, which yields the same result as before.
- The same reasoning holds for q.

Welfare aspects of equilibria

- Some NE are "better" than others, even if all are Pareto optimal.

Welfare aspects of equilibria

- Some NE are “better” than others, even if all are Pareto optimal.
- For example, in BoS the mixed NE has smaller utilitarian welfare than the pure equilibria.

Welfare aspects of equilibria

- Some NE are "better" than others, even if all are Pareto optimal.
- For example, in BoS the mixed NE has smaller utilitarian welfare than the pure equilibria.
- However the mixed NE is better when we consider the egalitarian welfare.

Welfare aspects of equilibria

- Some NE are "better" than others, even if all are Pareto optimal.
- For example, in BoS the mixed NE has smaller utilitarian welfare than the pure equilibria.
- However the mixed NE is better when we consider the egalitarian welfare.
- For either measure of welfare, we can compute the ratio of the best possible outcome to the best/worst outcome obtained in a NE. This is called the price of stability/price of anarchy.

Correlated equilibria

- Suppose that instead of randomizing independently, the players make random choices that are correlated.

Correlated equilibria

- Suppose that instead of randomizing independently, the players make random choices that are correlated.
- For example, in BoS, toss a coin and play (a, a) if heads, (b, b) if tails.

Correlated equilibria

- Suppose that instead of randomizing independently, the players make random choices that are correlated.
- For example, in BoS, toss a coin and play (a, a) if heads, (b, b) if tails.
- The expected payoff here is better for both than in any of the Nash equilibria.

Correlated equilibria

- Suppose that instead of randomizing independently, the players make random choices that are correlated.
- For example, in BoS, toss a coin and play (a, a) if heads, (b, b) if tails.
- The expected payoff here is better for both than in any of the Nash equilibria.
- Neither player has incentive to deviate from this unilaterally. Binding agreements are not needed.

Correlated equilibria

- Suppose that instead of randomizing independently, the players make random choices that are correlated.
- For example, in BoS, toss a coin and play (a, a) if heads, (b, b) if tails.
- The expected payoff here is better for both than in any of the Nash equilibria.
- Neither player has incentive to deviate from this unilaterally. Binding agreements are not needed.
- The coordination process can be done by an outside agent who sends private signals to the players, or by a public signal as in the above example.

Correlated equilibria

- Suppose that instead of randomizing independently, the players make random choices that are correlated.
- For example, in BoS, toss a coin and play (a, a) if heads, (b, b) if tails.
- The expected payoff here is better for both than in any of the Nash equilibria.
- Neither player has incentive to deviate from this unilaterally. Binding agreements are not needed.
- The coordination process can be done by an outside agent who sends private signals to the players, or by a public signal as in the above example.
- Correlated equilibria are easy to compute, and welfare functions can be optimized over them easily (the set of CE is convex which allows a lot of nice mathematical tools to be used).

Overview

We cover two topics related to network models: importance of nodes and influence of nodes on other nodes.

Centrality

- There are many ways to measure “importance” of a node v in a network G.

Centrality

- There are many ways to measure "importance" of a node v in a network G.
- One is to imagine information flowing between nodes via shortest paths.

Centrality

- There are many ways to measure "importance" of a node v in a network G.
- One is to imagine information flowing between nodes via shortest paths.
- Assuming v can inspect information flowing past it, and information flows randomly, the more such paths containing v, the more importance v has.

Centrality

- There are many ways to measure "importance" of a node v in a network G.
- One is to imagine information flowing between nodes via shortest paths.
- Assuming v can inspect information flowing past it, and information flows randomly, the more such paths containing v, the more importance v has.
- The betweenness centrality of v is

$$\sum_{v \neq s \neq t \neq v} \frac{\text{number of shortest paths joining } s \text{ and } t \text{ that contain } v}{\text{number of shortest paths joining } s \text{ and } t}$$

Centrality

- There are many ways to measure "importance" of a node v in a network G.
- One is to imagine information flowing between nodes via shortest paths.
- Assuming v can inspect information flowing past it, and information flows randomly, the more such paths containing v, the more importance v has.
- The betweenness centrality of v is

$$\sum_{v \neq s \neq t \neq v} \frac{\text{number of shortest paths joining } s \text{ and } t \text{ that contain } v}{\text{number of shortest paths joining } s \text{ and } t}$$

- This measure is often normalized to lie between 0 and 1, by dividing, for example, by the maximum possible number of (s, t) pairs.

Indegree-based centrality

- In a directed network representing citations, it makes sense to consider nodes as more important if they have more direct citations. This leads to the indegree centrality (i.e. the number of incoming edges, possibly normalized by the total number of edges).

Indegree-based centrality

- In a directed network representing citations, it makes sense to consider nodes as more important if they have more direct citations. This leads to the indegree centrality (i.e. the number of incoming edges, possibly normalized by the total number of edges).
- However 2nd, 3rd, etc, level citations should also count, but maybe less. One idea is to weight the level i citations by a factor α^i, for some fixed small positive α.

Indegree-based centrality

- In a directed network representing citations, it makes sense to consider nodes as more important if they have more direct citations. This leads to the indegree centrality (i.e. the number of incoming edges, possibly normalized by the total number of edges).
- However 2nd, 3rd, etc, level citations should also count, but maybe less. One idea is to weight the level i citations by a factor α^i, for some fixed small positive α.
- If A is the adjacency matrix, the total importance of v is

$$\sum_j \alpha^k (A^k)_{jv}$$

Indegree-based centrality

- In a directed network representing citations, it makes sense to consider nodes as more important if they have more direct citations. This leads to the indegree centrality (i.e. the number of incoming edges, possibly normalized by the total number of edges).
- However 2nd, 3rd, etc, level citations should also count, but maybe less. One idea is to weight the level i citations by a factor α^i, for some fixed small positive α.
- If A is the adjacency matrix, the total importance of v is

$$\sum_j \alpha^k (A^k)_{jv}$$

- The vector of all *Katz centrality* scores can therefore be written after some algebra as

$$\left((I - \alpha A^T)^{-1} - I \right) \mathbf{1}$$

Eigenvector centrality

- The idea that a work should be important if it is cited by important nodes leads to a circular-looking description

$$x_v = \frac{1}{\lambda} \sum_j A_{jv} x_j$$

where λ is a normalizing factor.

Eigenvector centrality

- The idea that a work should be important if it is cited by important nodes leads to a circular-looking description

  $$x_v = \frac{1}{\lambda} \sum_j A_{jv} x_j$$

  where λ is a normalizing factor.
- This can be written as $A^T \mathbf{x} = \lambda \mathbf{x}$, an eigenvalue problem.

Eigenvector centrality

- The idea that a work should be important if it is cited by important nodes leads to a circular-looking description

$$x_v = \frac{1}{\lambda} \sum_j A_{jv} x_j$$

where λ is a normalizing factor.

- This can be written as $A^T \mathbf{x} = \lambda \mathbf{x}$, an eigenvalue problem.
- We look for the largest real eigenvalue λ; it has an eigenvector with all positive values (this is guaranteed to exist by "Perron-Frobenius theory").

Eigenvector centrality

- The idea that a work should be important if it is cited by important nodes leads to a circular-looking description

$$x_v = \frac{1}{\lambda} \sum_j A_{jv} x_j$$

where λ is a normalizing factor.

- This can be written as $A^T \mathbf{x} = \lambda \mathbf{x}$, an eigenvalue problem.
- We look for the largest real eigenvalue λ; it has an eigenvector with all positive values (this is guaranteed to exist by "Perron-Frobenius theory").
- This gives the eigenvector centrality which we may want to normalize, as usual.

Diffusion in networks – De Groot model

- Very commonly we want to understand influence of nodes on other nodes (e.g. spread of ideas, spread of behaviors).

Diffusion in networks – De Groot model

- Very commonly we want to understand influence of nodes on other nodes (e.g. spread of ideas, spread of behaviors).
- There are many discrete time models. One for spread of opinions is the DeGroot model.

Diffusion in networks – De Groot model

- Very commonly we want to understand influence of nodes on other nodes (e.g. spread of ideas, spread of behaviors).
- There are many discrete time models. One for spread of opinions is the DeGroot model.
- Here each node has an opinion, which is a number in the range $[0, 1]$. Each node v also gets influenced by each node w by some amount p_{vw}, and averages the opinions of its neighbors. These numbers can be encoded in a weighted adjacency matrix.

Diffusion in networks – De Groot model

- Very commonly we want to understand influence of nodes on other nodes (e.g. spread of ideas, spread of behaviors).
- There are many discrete time models. One for spread of opinions is the DeGroot model.
- Here each node has an opinion, which is a number in the range $[0,1]$. Each node v also gets influenced by each node w by some amount p_{vw}, and averages the opinions of its neighbors. These numbers can be encoded in a weighted adjacency matrix.
- This yields a Markov chain, which in most cases converges to a steady-state distribution given by an eigenvalue problem, where all opinions are the same (consensus).

Diffusion in networks – De Groot model

- Very commonly we want to understand influence of nodes on other nodes (e.g. spread of ideas, spread of behaviors).
- There are many discrete time models. One for spread of opinions is the DeGroot model.
- Here each node has an opinion, which is a number in the range $[0, 1]$. Each node v also gets influenced by each node w by some amount p_{vw}, and averages the opinions of its neighbors. These numbers can be encoded in a weighted adjacency matrix.
- This yields a Markov chain, which in most cases converges to a steady-state distribution given by an eigenvalue problem, where all opinions are the same (consensus).
- A more general model (Friedkin-Johnsen) also has a probability of staying with the initial opinion.

De Groot model example

- Suppose the transition matrix is $T = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ and the initial belief vector $(1/2, 0, 2/3)^T$.

De Groot model example

- Suppose the transition matrix is $T = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ and the initial belief vector $(1/2, 0, 2/3)^T$.
- Then $p(t+1) = Tp(t)$ so $p(t) = T^t p(0)$.

De Groot model example

- Suppose the transition matrix is $T = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ and the initial belief vector $(1/2, 0, 2/3)^T$.
- Then $p(t+1) = Tp(t)$ so $p(t) = T^t p(0)$.
- The steady state distribution is obtained by solving for the left eigenvector $s = sT$. Perron-Frobenius theory again tells us there will be an all-positive solution.

De Groot model example

- Suppose the transition matrix is $T = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ and the initial belief vector $(1/2, 0, 2/3)^T$.
- Then $p(t+1) = Tp(t)$ so $p(t) = T^t p(0)$.
- The steady state distribution is obtained by solving for the left eigenvector $s = sT$. Perron-Frobenius theory again tells us there will be an all-positive solution.
- We obtain $s = (2/5, 2/5, 1/5)$ and so the unanimous steady state belief is $sp(0) = 1/3$.

De Groot model example

- Suppose the transition matrix is $T = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ and the initial belief vector $(1/2, 0, 2/3)^T$.
- Then $p(t+1) = Tp(t)$ so $p(t) = T^t p(0)$.
- The steady state distribution is obtained by solving for the left eigenvector $s = sT$. Perron-Frobenius theory again tells us there will be an all-positive solution.
- We obtain $s = (2/5, 2/5, 1/5)$ and so the unanimous steady state belief is $sp(0) = 1/3$.
- Note that the influence of the first two nodes on the final belief is twice that of the third node.

Diffusion in networks - threshold models

- A very different type of model is a threshold model. Typically each node is in one of a finite number of states, and most commonly there are only two states (for example, "riot" or "don't riot") .

Diffusion in networks - threshold models

- A very different type of model is a threshold model. Typically each node is in one of a finite number of states, and most commonly there are only two states (for example, "riot" or "don't riot") .
- A node changes its state to copy the majority of its neighbors (or some other fixed fraction θ).

Diffusion in networks - threshold models

- A very different type of model is a threshold model. Typically each node is in one of a finite number of states, and most commonly there are only two states (for example, "riot" or "don't riot") .
- A node changes its state to copy the majority of its neighbors (or some other fixed fraction θ).
- This can lead to complicated behavior that depends a lot on the structure of the network — consensus or lack of it, no convergence, etc. In many cases we do get convergence to the consensus.