Decision Making

SIPTA Summer School 2014, Montpellier

Matthias C. M. Troffaes

Department of Mathematical Sciences
Durham University

23 July 2014

Outline

Preparation: R code

1. start R
2. visit with browser: `https://raw.githubusercontent.com/mcmtroffaes/improb-redux/master/improb-redux.r`
3. select and copy all R code from browser: CTRL-A, CTRL-C
4. go to R console
5. paste code into R console: CTRL-V, ENTER
6. keep browser window open, so you can rinse & repeat steps 3–5 every time you start a new R session

Outline

What is Decision Making: Offshore Wind Example

ECOLOGIST
SETTING THE ENVIRONMENTAL AGENDA SINCE 1970

CLIMATE CHANGE | FOOD + FARMING | ENERGY | HEALTH | POLITICS + ECONOMICS | SCIENCE + TECHNOLOGY | NATURAL WORLD | SOCIET

NEWS | GREEN LIVING | CAMPAIGNING | BLOGS & COMMENTS | INTERVIEWS | MAGAZINE | VIDEO | ARCHIVE | EXTRAS | DIRECTOR

Home | Blogs and Comments | Comment

SEARCH Go

Offshore wind is too expensive!

Jim Platts

7th December 2013

Tweet 28 | +1 0 | Like 35

Prices paid in the UK to solar and wind generators will change to favour offshore wind at the expense of the others. Jim Platt warns that the policy is doomed to failure - offshore wind is just too expensive, and likely to remain so.

> “Building things in the middle of the sea is very, very expensive, and that isn't going to change.”

In 2012 the global wind industry manufactured and installed more than 20,000 turbines generating 45GW of energy. The leading firm alone, Danish company Vestas, installed more than 6GW of this via 2,500 turbines.

Offshore wind on the rocks? Photo: Chris Lishman / Shutterstock.com.

More articles about

wind | renewables | energy | oceans |

Related Articles

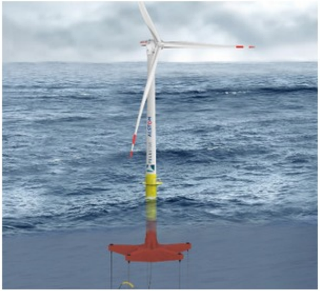
- Brazil and FIFA have failed to protect their World Cup mascot

What is Decision Making: Offshore Wind Example

Floating turbines could cut offshore wind energy costs: study

11 July 2014

Print Email



Floating turbines could cut the cost of offshore wind power to below £85/MWh by the mid-2020s, according to an engineering design study by The Glosten Associates for the Energy Technologies Institute (ETI).

The new study revealed that the company's PelaStar tension leg floating platform (TLP) could deliver further reductions as the technology matures and is designed to provide high capacity factors in wind speeds exceeding 10m per second in water between 60m and 1,200m deep.

The UK is said to have over a third of Europe's potential offshore wind resource, which is enough to meet the power demand of the country nearly three times over.

The FEED study has shown that Glosten's PelaStar TLP design could play a major role in reducing UK offshore wind energy costs.

The company said that the TLP technology is suitable for water depths from as low as 55m up to several hundred metres.

What is Decision Making: Offshore Wind Example

Operations & Maintenance of Offshore Wind

30% of cost of offshore wind is operations & maintenance
= **huge** chunk of money

Types of Maintenance

- preventive (prevent future failures)
- corrective (fix after failure)

What is Decision Making: Offshore Wind Example

Decisions

criterion: minimize cost

- when to perform maintenance?
- what is a good preventive/corrective balance?

limiting factor = wind speed & wave height for boarding offshore turbine

Uncertainties

Enormous potential for saving costs by making accurate predictions of:

- wind & waves at different time scales
  avoid missing maintenance opportunities
  avoid costly transport when turbine cannot be boarded
- forecast failures before they happen
  cost of preventing $\lll$ cost of fixing

What is Decision Making: Offshore Wind Example

drastically different issues at different time scales:

Short Term: Optimize Actual Operations

what data on the wind farm should we collect
how to use it?

Medium Term: Business Case

how to convince investors to invest in offshore wind
may not have very much data to go from!

Long Term: Policy & Politics

should we encourage offshore, or look at other technologies?
very little data to go by, enormous uncertainty concerning future climate change, attitude of electorate, etc.
not just about money

What is Decision Making: Offshore Wind Example

Why Use Imprecise Probability for Decision Making?

- increases confidence in analysis based on sparse data
  may help at all levels/time horizons
- risk-averse industries: rare events with large impact

Why **NOT** Use Imprecise Probability for Decision Making?

- computational expense
- abundant data, non-critical decisions
  standard statistical treatment works as well

The Elephant in the Room

how to communicate uncertainty?
uncertainty analysis is only useful if the results can be communicated

Outline

Review of Classical Decision Theory: Example

Example: Visit Offshore Turbine by Boat in the Next Hour?

- parameter: average wave height X for next hour: unknown!
  assume only possible values are $x = 0.5$ and $x = 2$
- data: observation Y, say average wave height in last hour
  assume only possible values are $y = 0.5$ and $y = 2$
- decision: d = take boat, or d = do not take boat
- decision strategy δ:
  which decision to make based on data y?

Review of Classical Decision Theory: Example

Example: Visit Offshore Turbine by Boat in the Next Hour?

- utility function $U(d, x)$: each combination of decision and parameter leads to a different final reward value
  - can only board offshore turbine for maintenance if $X < 1$
  - taking boat costs €1000
  - doing maintenance saves €4000

  for example, expressed in units of €1000:

$U(d, x)$	$x = 0.5$	$x = 2$
$d =$ boat	3	−1
$d =$ no boat	0	0

- likelihood: probability of data given parameter $p(y \mid x)$

$p(y \mid x)$	$y = 0.5$	$y = 2$
$x = 0.5$	0.9	0.1
$x = 2$	0.3	0.7

- prior: probability of parameter $p(x)$ before you have seen the data

$p(x)$	$x = 0.5$	$x = 2$
	0.4	0.6

Review of Classical Decision Theory: Example

Frequentist Solution: Wald's Expected Utility, Admissibility

frequentist = only use likelihood

1. for every possible decision strategy δ mapping observations to decisions
   and for every possible value x of X
   calculate Wald's expected utility

   expected utility = −risk

$$U(\delta|x) := E(U(\delta(Y), x)|x) = \sum_y U(\delta(y), x)p(y|x) \tag{1}$$

2. a strategy δ is inadmissible if there is another strategy δ' such that
   $U(\delta'|x) \geq U(\delta|x)$ for all x, and
   $U(\delta'|x) > U(\delta|x)$ for at least one x

   partial ordering of strategies
3. **optimal Wald strategy**
   all admissible (non-inadmissible) strategies

   maximal elements w.r.t. partial ordering

Review of Classical Decision Theory: Example

Bayesian Solution: Maximize Posterior Expected Utility

Bayesian = only use posterior ($\propto$ likelihood $\times$ prior)

1. calculate the posterior

$$p(x|y) = \frac{p(y|x)p(x)}{\sum_{x'} p(y|x')p(x')} \tag{2}$$

2. for every possible observation y
and every possible decision d
calculate the posterior expected utility:

$$U(d|y) = E(U(d,X)|y) = \sum_x U(d,x)p(x|y) \tag{3}$$

3. **optimal Bayes strategy** δ^* maximizes posterior expected utility

$$\delta^*(y) = \arg\max_d U(d|y) \tag{4}$$

much easier to calculate than Wald's inadmissible strategies! (why?)

Review of Classical Decision Theory: Wald's Theorem

Wald's Theorem (1939 [17])

The set of Wald admissible strategies can always be recovered from a Bayesian analysis, simply by varying the prior over all possible distributions.

[Technical details omitted!]

'equivalence' between $\underbrace{\text{robust Bayesian statistics}}_{\text{sets of priors}}$ and frequentist statistics

Plan

- develop decision making directly from sets of distributions
- first 'one-shot' decisions without data — pretend you already know y: find $\delta(y)$
- then 'sequential' decisions with data — backward induction

Outline

A Very Simple Example

Example (Machinery, Overtime, or Nothing?)

A company makes a product, and believes in increasing future demand. The manager asks you, the decision expert, whether he should buy new machinery, use overtime, or do nothing. The upcoming year, demand can either increase or remain the same.

If we buy new machinery, then the profit at the end of the year will be 440 (in thousands of pounds) if demand increases, and 260 otherwise. On the other hand, if we use overtime, then the profit will be 420 if demand increases, and 300 otherwise. If we do nothing, profit will be 370.

According to our best current judgement, demand will increase with probability at least 0.5, and at most 0.8.

What advice can we give the manager?

The Basic Elements of a Decision Problem

- decisions: $\{$buy new machinery, use overtime, do nothing$\}$
- events: $\{$demand increases, demand stays$\}$
- rewards: a monetary value, depending on decisions and events
- decision maker may have information about the events (e.g. bounds on the probabilities of the events)

Outline

Decision Trees

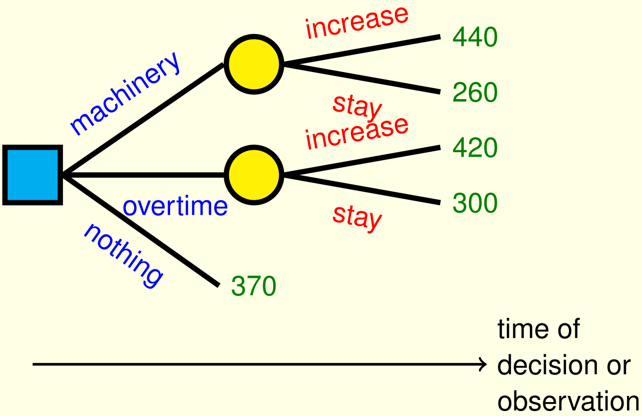
not to be confused with decision trees in classification!

Graphical representation of decisions, events, and rewards:

decision nodes + decisions

rewards

chance nodes + events



Static and Sequential Decision Problems

The problem we are investigating is of a very simple type...

- we must make a single decision,
- which is followed by the observation of an event, and
- which is in turn followed by a reward for us,
  depending on the decision we made, and the event that occurred

Informally...

Definition (Static Decision Problem)

Any decision tree that has

- a single decision node at its root,
- and no other decision node.

Definition (Sequential Decision Problem)

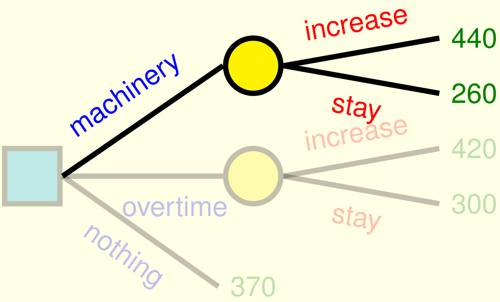
Any other decision tree.

Outline

The Problem of Choice: Gambles

Observation

in a static decision problem,
each decision branch corresponds to a gamble



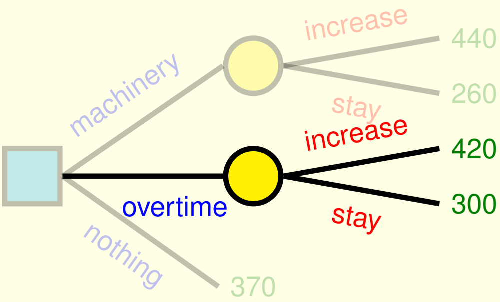
$$X_{\text{machinery}}(\text{increase}) = 440$$

$$X_{\text{machinery}}(\text{stay}) = 260$$

The Problem of Choice: Gambles

Observation

in a static decision problem,
each decision branch corresponds to a gamble

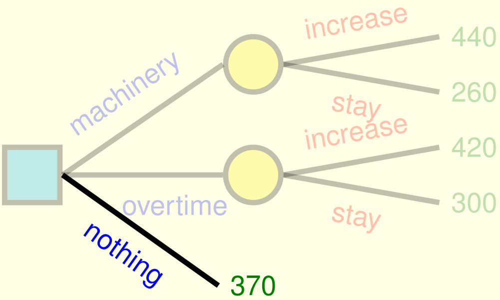


$$X_{\text{overtime}}(\text{increase}) = 420$$
$$X_{\text{overtime}}(\text{stay}) = 300$$

The Problem of Choice: Gambles

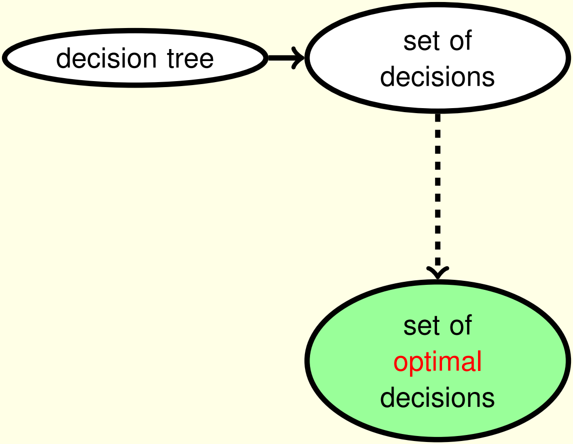
Observation

in a static decision problem,
each decision branch corresponds to a gamble

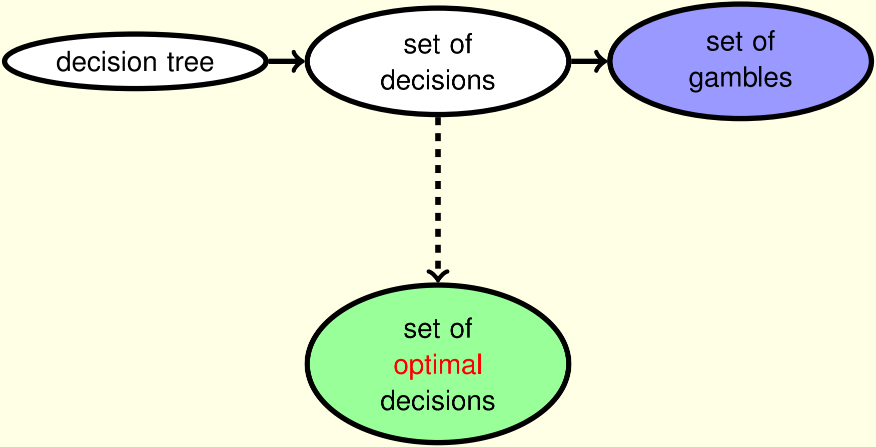


$$X_{\text{nothing}}(\text{increase}) = 370$$
$$X_{\text{nothing}}(\text{stay}) = 370$$

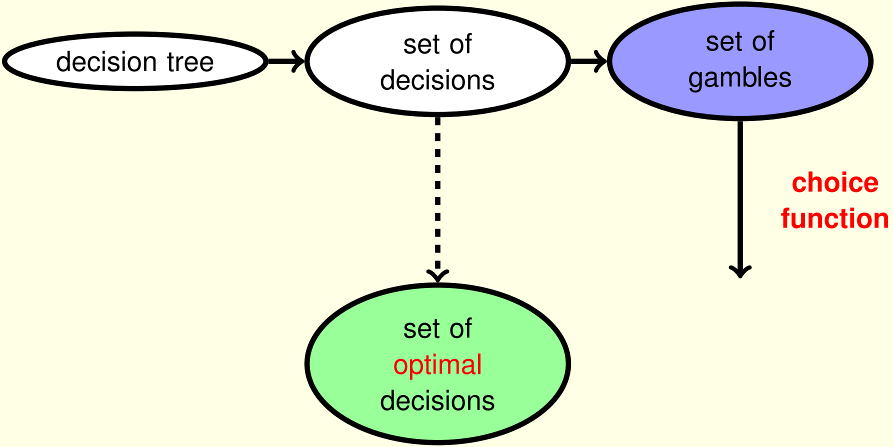
The Problem of Choice: Choice Function



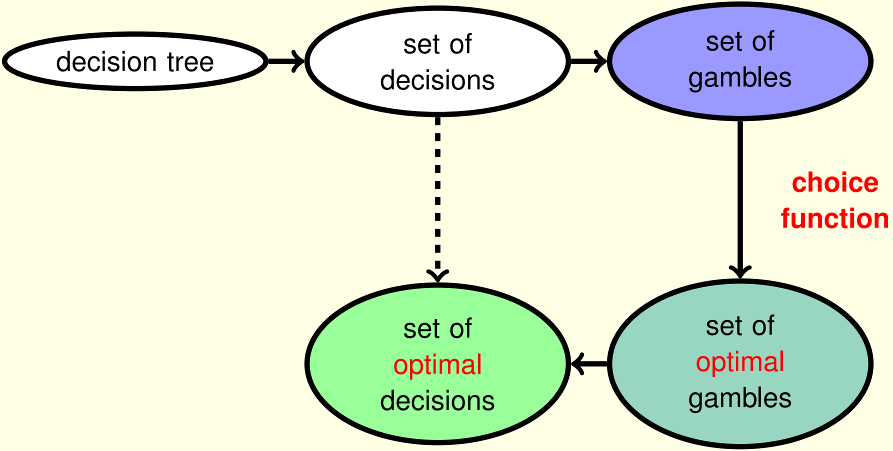
The Problem of Choice: Choice Function



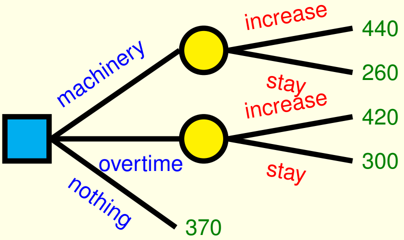
The Problem of Choice: Choice Function



The Problem of Choice: Choice Function



The Problem of Choice: Example



set of decisions

machinery
overtime
nothing

each row is a gamble

set of gambles

increase	stay
440	260
420	300
370	370

choice function

what is a good choice function, under severe uncertainty?

set of optimal gambles

increase	stay
420	300
370	370

set of optimal decisions

overtime
nothing

The Problem of Choice

- we know how to go from decision trees to gambles
- only one problem left to solve:

what is a good choice function?

- a standard choice function: maximize the expectation of the reward

however, as argued in the last few days. . .

- under severe uncertainty, we may not be able to identify a unique probability mass function p which describes our knowledge accurately
- still, we may be able to identify a set of probability mass functions $\mathcal{M}$

credal set

what are good choice functions
for sets of probability mass functions?

The Problem of Choice

Theorem (Approximate Representation Theorem)

For every coherent lower prevision $\underline{P}$ there is a finite set $\mathcal{M}$ of probability mass functions such that, for every gamble X on Ω,

$$\underline{P}(X) \simeq \min_{p \in \mathcal{M}} E_p(X) \qquad \text{where } E_p(X) := \sum_{\omega \in \Omega} p(\omega) X(\omega) \tag{5}$$

Example (Machinery, Overtime, or Nothing?)

In our example, increase has probability at least 0.5 and at most 0.8, so

$$\mathcal{M} = \begin{array}{r|cc} & p_1 & p_2 \\ \hline \text{increase} & 0.5 & 0.8 \\ \text{stay} & 0.5 & 0.2 \end{array}$$ (each column is a probability mass function)

what are good choice functions for finite sets of probability mass functions?

Recapitulating

		events		set of probabilities $\mathcal{M}$	
		increase	stay	p_1	p_2
events	increase			0.5	0.8
	stay			0.5	0.2
set of decisions	machinery	440	260		
	overtime	420	300		
	nothing	370	370		
		set of gambles			

which of the gambles are optimal?

Outline

Γ-Maximin

(Wald 1945 [18], Gilboa & Schmeidler 1989 [5])

Definition (Γ-Maximin Optimality Criterion)

Choose any gamble whose lower prevision is maximal.

Recipe (Γ-Maximin Optimality Criterion)

1. set up the table with gambles and probabilities
2. calculate the expectation of each gamble with respect to each probability mass function — matrix multiplication
3. calculate the minimum expectation of each gamble — minimum of each row
4. choose the decision with the highest minimum expectation — maximize

$$\arg\max_{d\in D} \underline{P}(X_d) \tag{6}$$

Γ-Maximin: Example

Example (Machinery, Overtime, or Nothing)

	increase	stay	p_1	p_2	$\underline{P}$
increase			0.5	0.8	
stay			0.5	0.2	
machinery	440	260			
overtime	420	300			
nothing	370	370			
	(1)		(2)		(3) & (4)

```
pmfs = c(
  0.5, 0.5,
  0.8, 0.2)
rvars = c(
  440, 260,
  420, 300,
  370, 370)
getexpectations = getexpectationsfunc(2, pmfs)
getlowerprevisions = getlowerprevisionsfunc(getexpectations)
isgammamaximin = isgammamaxisomethingfunc(getlowerprevisions)
isgammamaximin(rvars)
```

Γ-Maximax

(Satia and Lave 1973 [12], probably others as well)

- Γ-maximin seems overly pessimistic; something more optimistic?

Definition (Γ-Maximax Optimality Criterion)

Choose any gamble whose *upper* prevision is maximal.

Recipe (Γ-Maximax Optimality Criterion)

1. set up the table with gambles and probabilities
2. calculate the expectation of each gamble
   with respect to each probability mass function — matrix multiplication
3. calculate the *maximum* expectation of each gamble — maximum of each row
4. choose the decision with the highest maximum expectation — maximize

$$\arg\max_{d\in D} \overline{P}(X_d) \tag{7}$$

Γ-Maximax: Example

Example (Machinery, Overtime, or Nothing)

	increase	stay	p_1	p_2	$\overline{P}$
increase			0.5	0.8	
stay			0.5	0.2	
machinery	440	260			
overtime	420	300			
nothing	370	370			
	(1)		(2)		(3) & (4)

```
pmfs = c(
  0.5, 0.5,
  0.8, 0.2)
rvars = c(
  440, 260,
  420, 300,
  370, 370)
getexpectations = getexpectationsfunc(2, pmfs)
getupperprevisions = getupperprevisionsfunc(getexpectations)
isgammamaximax = isgammamaxisomethingfunc(getupperprevisions)
isgammamaximax(rvars)
```

Interval Maximality

literature: 'interval dominance'

(Condorcet 1785 [3], Sen 1977 [15], Satia and Lave 1973 [12], Kyburg 1983 [9], *many* others)

- get every reasonable option (from pessimistic to optimistic) at once?

Definition (Partial Ordering by Interval Comparison)

We say that a gamble X interval dominates Y, and write

$$X \sqsupset Y \tag{8}$$

whenever

$$\underline{P}(X) > \overline{P}(Y) \tag{9}$$

$[\overline{P}(X), \underline{P}(X)]$ dominates $[\overline{P}(Y), \underline{P}(Y)]$

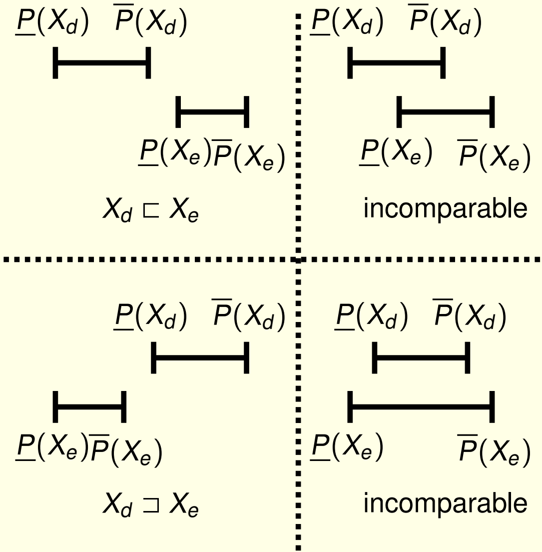
Definition (Interval Maximality Optimality Criterion)

Choose any gamble which is undominated with respect to $\sqsupset$.

$$\{d\colon (\forall e \in D)(X_d \not\sqsubset X_e)\} \tag{10}$$

Interval Maximality: Partial Ordering

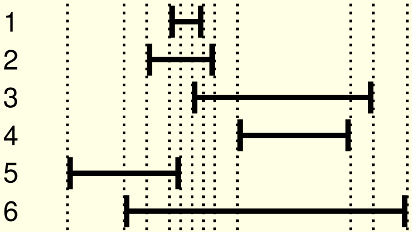
- $\sqsubset$ determines a partial ordering between gambles



Interval Maximality: Hasse Diagram & Algorithm

maximal elements with partial ordering = undominated elements

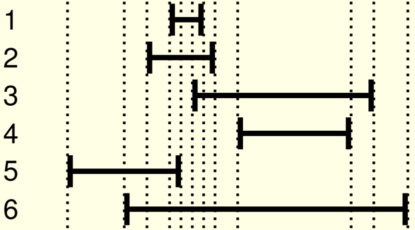
example:



Interval Maximality: Hasse Diagram & Algorithm

maximal elements with partial ordering = undominated elements

example:



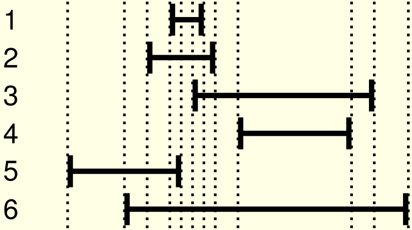
Hasse diagram

1

Interval Maximality: Hasse Diagram & Algorithm

maximal elements with partial ordering = undominated elements

example:



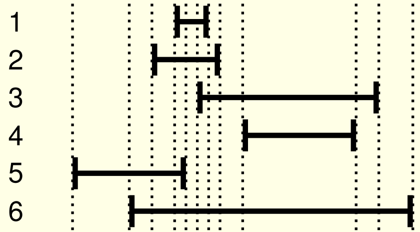
Hasse diagram

1 2

Interval Maximality: Hasse Diagram & Algorithm

maximal elements with partial ordering = undominated elements

example:



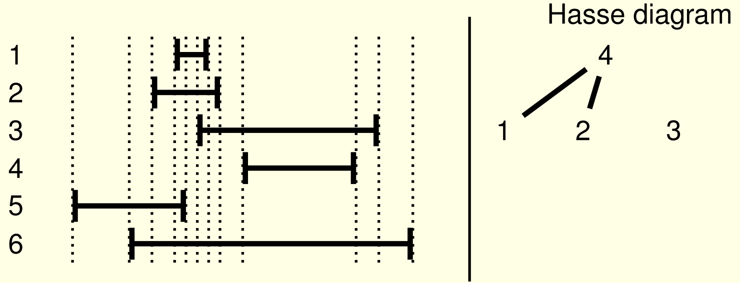
Hasse diagram

1 2 3

Interval Maximality: Hasse Diagram & Algorithm

maximal elements with partial ordering = undominated elements

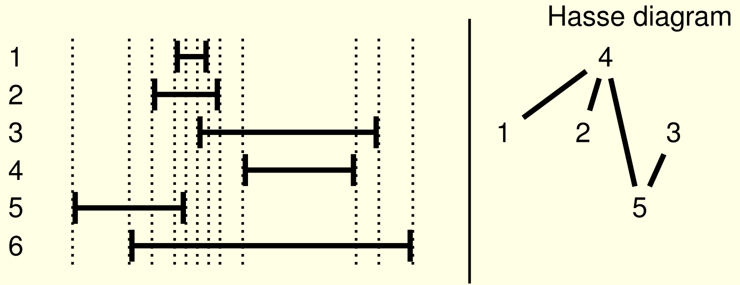
example:



Interval Maximality: Hasse Diagram & Algorithm

maximal elements with partial ordering = undominated elements

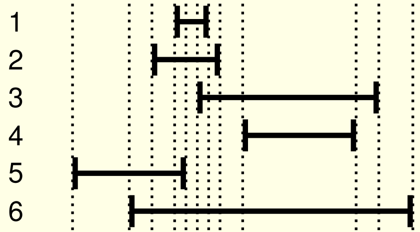
example:



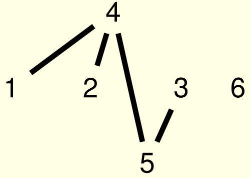
Interval Maximality: Hasse Diagram & Algorithm

maximal elements with partial ordering = undominated elements

example:



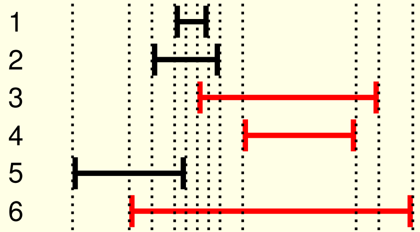
Hasse diagram

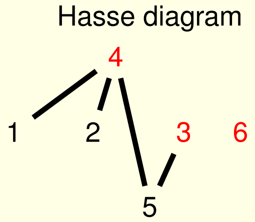


Interval Maximality: Hasse Diagram & Algorithm

maximal elements with partial ordering = undominated elements

example:

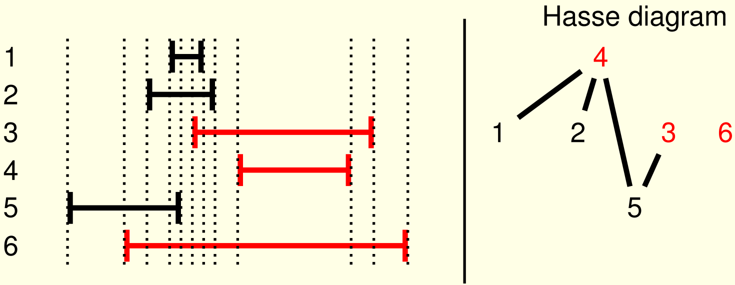




Interval Maximality: Hasse Diagram & Algorithm

maximal elements with partial ordering = undominated elements

example:



Theorem

All non-interval-maximal elements are dominated by the interval that has the highest lower bound.

$\Longrightarrow$ no need for Hasse diagram to find interval maximal elements

Interval Maximality: Practical Implementation

Recipe (Interval Maximality Optimality Criterion)

1. set up the table with gambles and probabilities
2. calculate the expectation of each gamble
   with respect to each probability mass function — matrix multiplication
3. calculate the *minimum and maximum* expectation of each gamble
   = interval expectation — minimum & maximum of each row
4. choose the decisions whose maximum expectation
   exceeds the overall largest minimum expectation — undominated intervals

$$\left\{ d \colon \overline{P}(X_d) \geq \sup_{e \in D} \underline{P}(X_e) \right\} \tag{11}$$

Interval Maximality: Example

Example (Machinery, Overtime, or Nothing)

	increase	stay	p_1	p_2	$\underline{P}$	$\overline{P}$
increase			0.5	0.8		
stay			0.5	0.2		
machinery	440	260				
overtime	420	300				
nothing	370	370				
	(1)		(2)		(3) & (4)	

```
pmfs = c(
  0.5, 0.5,
  0.8, 0.2)
rvars = c(
  440, 260,
  420, 300,
  370, 370)
getexpectations = getexpectationsfunc(2, pmfs)
isintervalmaximal = ismaximalfunc(getexpectations, intervalcompare)
isintervalmaximal(rvars)
```

Robust Bayes Maximality

literature: 'maximality'

(Condorcet 1785 [3], Sen 1977 [15], Walley 1991 [19])

- exploits the behavioural interpretation of lower previsions
- refines interval maximality (see Exercise 3 later!)

Definition (Partial Ordering by Robust Bayesian Comparison)

We say that X robust Bayes dominates Y, and write

$$X \succ Y \tag{12}$$

whenever any of the following equivalent conditions hold:

$$(\forall p \in \mathcal{M})\,(E_p(X) > E_p(Y)) \tag{13}$$

$$\underline{P}(X - Y) > 0 \tag{14}$$

(willing to pay a small amount in order to trade Y for X)
($X - Y + \epsilon$ is desirable for some $\epsilon > 0$)

Remember, for any probability mass function p and any gamble X:

$$E_p(X) := \sum_{\omega \in \Omega} p(\omega) X(\omega) \tag{15}$$

Robust Bayes Maximality: Hasse Diagram & Algorithm

Definition (Robust Bayes Maximality Optimality Criterion)

Choose any gamble which is undominated with respect to $\succ$.

example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	−1
X_2	0	0	0
X_3	0.5	−1	−2
X_4	0.2	−2	−3
X_5	2	1	−0.5

Robust Bayes Maximality: Hasse Diagram & Algorithm

Definition (Robust Bayes Maximality Optimality Criterion)

Choose any gamble which is undominated with respect to $>$.

example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	−1
X_2	0	0	0
X_3	0.5	−1	−2
X_4	0.2	−2	−3
X_5	2	1	−0.5

Hasse diagram

1

Robust Bayes Maximality: Hasse Diagram & Algorithm

Definition (Robust Bayes Maximality Optimality Criterion)

Choose any gamble which is undominated with respect to $>$.

example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	−1
X_2	0	0	0
X_3	0.5	−1	−2
X_4	0.2	−2	−3
X_5	2	1	−0.5

Hasse diagram

1 2

Robust Bayes Maximality: Hasse Diagram & Algorithm

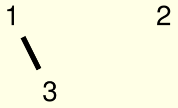
Definition (Robust Bayes Maximality Optimality Criterion)

Choose any gamble which is undominated with respect to $\succ$.

example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	-1
X_2	0	0	0
X_3	0.5	-1	-2
X_4	0.2	-2	-3
X_5	2	1	-0.5

Hasse diagram



Robust Bayes Maximality: Hasse Diagram & Algorithm

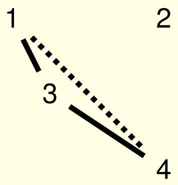
Definition (Robust Bayes Maximality Optimality Criterion)

Choose any gamble which is undominated with respect to $>$.

example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	−1
X_2	0	0	0
X_3	0.5	−1	−2
X_4	0.2	−2	−3
X_5	2	1	−0.5

Hasse diagram



Robust Bayes Maximality: Hasse Diagram & Algorithm

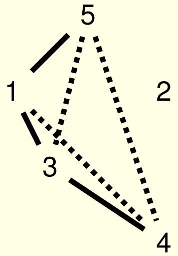
Definition (Robust Bayes Maximality Optimality Criterion)

Choose any gamble which is undominated with respect to $>$.

example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	-1
X_2	0	0	0
X_3	0.5	-1	-2
X_4	0.2	-2	-3
X_5	2	1	-0.5

Hasse diagram



Robust Bayes Maximality: Hasse Diagram & Algorithm

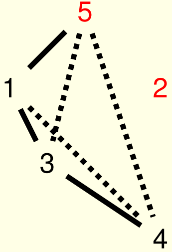
Definition (Robust Bayes Maximality Optimality Criterion)

Choose any gamble which is undominated with respect to $>$.

example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	-1
X_2	0	0	0
X_3	0.5	-1	-2
X_4	0.2	-2	-3
X_5	2	1	-0.5

Hasse diagram



Robust Bayes Maximality: Hasse Diagram & Algorithm

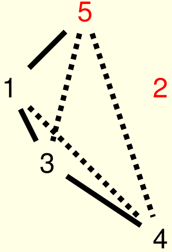
Definition (Robust Bayes Maximality Optimality Criterion)

Choose any gamble which is undominated with respect to $>$.

example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	-1
X_2	0	0	0
X_3	0.5	-1	-2
X_4	0.2	-2	-3
X_5	2	1	-0.5

Hasse diagram



Robust Bayes Maximality: Hasse Diagram & Algorithm

Definition (Robust Bayes Maximality Optimality Criterion)

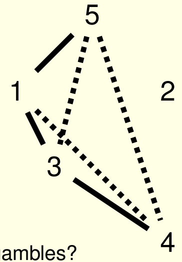
Choose any gamble which is undominated with respect to $>$.

example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	-1
X_2	0	0	0
X_3	0.5	-1	-2
X_4	0.2	-2	-3
X_5	2	1	-0.5

for browny points: interval maximal gambles?

Hasse diagram



Maximality

1

Theorem

Every non-maximal element is dominated by a maximal element.

holds for arbitrary partial ordering!

$\Longrightarrow$ no need for Hasse diagram to find maximal elements:
once non-maximal element removed, no need to consider further!

Robust Bayes Maximality: Hasse Diagram & Algorithm

Definition (Robust Bayes Maximality Optimality Criterion)

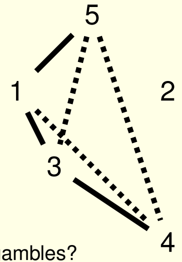
Choose any gamble which is undominated with respect to $>$.

example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	-1
X_2	0	0	0
X_3	0.5	-1	-2
X_4	0.2	-2	-3
X_5	2	1	-0.5

for browny points: interval maximal gambles?

Hasse diagram



Maximality

1 2

Theorem

Every non-maximal element is dominated by a maximal element.

holds for arbitrary partial orderings!

$\Longrightarrow$ no need for Hasse diagram to find maximal elements: once non-maximal element removed, no need to consider further!

Robust Bayes Maximality: Hasse Diagram & Algorithm

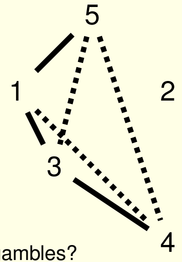
Definition (Robust Bayes Maximality Optimality Criterion)

Choose any gamble which is undominated with respect to $>$.

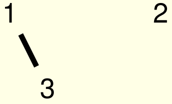
example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	−1
X_2	0	0	0
X_4	0.2	−2	−3
X_5	2	1	−0.5

Hasse diagram



Maximality



for browny points: interval maximal gambles?

Theorem

Every non-maximal element is dominated by a maximal element.

holds for arbitrary partial orderings!

$\Longrightarrow$ no need for Hasse diagram to find maximal elements:
once non-maximal element removed, no need to consider further!

Robust Bayes Maximality: Hasse Diagram & Algorithm

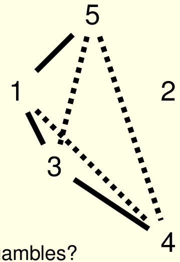
Definition (Robust Bayes Maximality Optimality Criterion)

Choose any gamble which is undominated with respect to $>$.

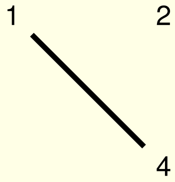
example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	-1
X_2	0	0	0
X_5	2	1	-0.5

Hasse diagram



Maximality



for browny points: interval maximal gambles?

Theorem

Every non-maximal element is dominated by a maximal element.

holds for arbitrary partial orderings!

$\Longrightarrow$ no need for Hasse diagram to find maximal elements:
once non-maximal element removed, no need to consider further!

Robust Bayes Maximality: Hasse Diagram & Algorithm

Definition (Robust Bayes Maximality Optimality Criterion)

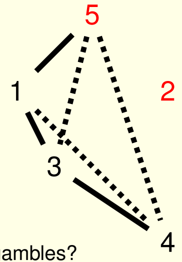
Choose any gamble which is undominated with respect to $>$.

example:

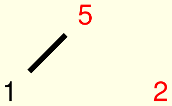
	E_{p_1}	E_{p_3}	E_{p_3}
X_2	0	0	0
X_5	2	1	-0.5

for browny points: interval maximal gambles?

Hasse diagram



Maximality



Theorem

Every non-maximal element is dominated by a maximal element.

holds for arbitrary partial orderings!

$\Longrightarrow$ no need for Hasse diagram to find maximal elements:
once non-maximal element removed, no need to consider further!

Robust Bayes Maximality: Practical Implementation

Recipe (Robust Bayes Maximality Optimality Criterion)

1. set up the table with gambles and probabilities
2. calculate the expectation of each gamble
   with respect to each probability mass function — matrix multiplication
3. sequentially remove all decisions
   whose expectation rows are point-wise dominated

Robust Bayes Maximality: Example

Example (Machinery, Overtime, or Nothing)

	increase	stay	p_1	p_2
increase			0.5	0.8
stay			0.5	0.2
machinery	440	260		
overtime	420	300		
nothing	370	370		
	(1)		(2)	

```
pmfs = c(
  0.5, 0.5,
  0.8, 0.2)
rvars = c(
  440, 260,
  420, 300,
  370, 370)
getexpectations = getexpectationsfunc(2, pmfs)
isrbayesmaximal = ismaximalfunc(getexpectations, rbayescompare)
isrbayesmaximal(rvars)
```

Robust Bayes Admissibility

literature: 'E-admissibility'

(Pascal 1662 [11], Levi 1980 [10], Berger 1984 [2], Walley 1991 [19])

- refines robust Bayes maximality

Definition (Robust Bayes Admissibility Optimality Criterion)

Choose any gamble which maximizes expectation
with respect to some $p \in \mathcal{M}$.

example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	-1
X_2	0	0	0
X_3	0.5	-1	-2
X_4	0.2	-2	-3
X_5	2	1	-0.5

notes:

- computational challenge if $\mathcal{M}$ is large
- not invariant under convex hull operation:
  not enough just to look at extreme points

Robust Bayes Admissibility

literature: 'E-admissibility'

(Pascal 1662 [11], Levi 1980 [10], Berger 1984 [2], Walley 1991 [19])

- refines robust Bayes maximality

Definition (Robust Bayes Admissibility Optimality Criterion)

Choose any gamble which maximizes expectation
with respect to some $p \in \mathcal{M}$.

example:

	E_{p_1}	E_{p_3}	E_{p_3}
X_1	1	0	−1
X_2	0	0	0
X_3	0.5	−1	−2
X_4	0.2	−2	−3
X_5	2	1	−0.5

notes:

- computational challenge if $\mathcal{M}$ is large
- not invariant under convex hull operation:
  not enough just to look at extreme points

Robust Bayes Admissibility: Practical Implementation

Recipe (Robust Bayes Admissibility Optimality Criterion)

1. set up the table with gambles and probabilities
2. calculate the expectation of each gamble
   with respect to each probability mass function — matrix multiplication
3. take all decisions that achieve a maximum
   in some expectation column

Robust Bayes Admissibility: Example

Example (Machinery, Overtime, or Nothing)

	increase	stay	p_1	p_2
increase			0.5	0.8
stay			0.5	0.2
machinery	440	260		
overtime	420	300		
nothing	370	370		
	(1)		(2) & (3)	

```
pmfs = c(
  0.5, 0.5,
  0.8, 0.2)
rvars = c(
  440, 260,
  420, 300,
  370, 370)
getexpectations = getexpectationsfunc(2, pmfs)
isrbayesadmissible = isrbayesadmissiblefunc(getexpectations)
isrbayesadmissible(rvars)
```

Robust Bayes Admissibility: Extreme Points Issue

Example (Machinery, Overtime, or Nothing)

	increase	stay	p_1	p_2	p_3
increase			0.5	0.8	0.65
stay			0.5	0.2	0.35
machinery	440	260			
overtime	420	300			
nothing	370	370			

```
pmfs = c(
  0.5, 0.5,
  0.8, 0.2,
  0.65, 0.35)
rvars = c(
  440, 260,
  420, 300,
  370, 370)
getexpectations = getexpectationsfunc(2, pmfs)
isrbayesadmissible = isrbayesadmissiblefunc(getexpectations)
isrbayesadmissible(rvars)
```

Outline

A Simple Example

(adapted from Kikuti et al. [8, Fig. 2])

Example (The Oil Wildcatter)

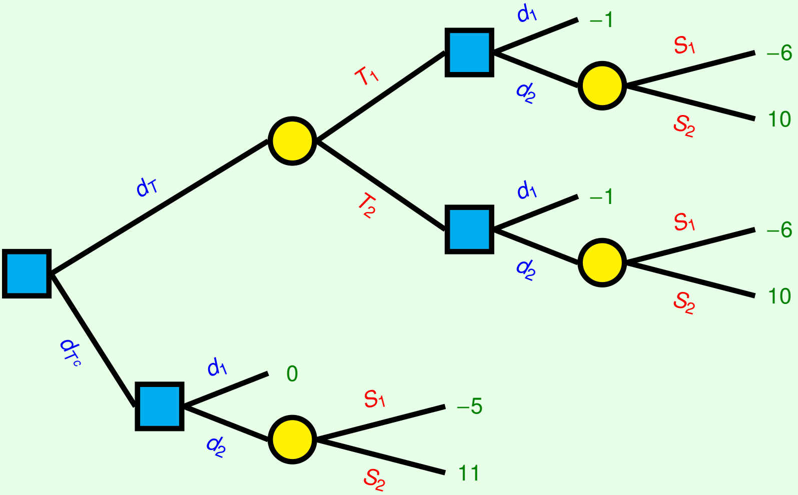
An oil wildcatter must decide whether to drill for oil (d_2) or not (d_1). Drilling costs 5 and provides a return of 0 or 16 depending on the richness of the site. The events S_1 and S_2 represent the different yields, with S_1 being the least profitable and S_2 the most. The subject may pay 1 to test the site before deciding whether to drill; this gives one of two results T_1 or T_2, where T_1 is the most pessimistic and T_2 the most optimistic. (All rewards in units of €10000.)

$\mathcal{M} =$

	p_1	p_2	p_3	p_4
$T1\&S1$	0.3	0.3	0.4	0.5
$T1\&S2$	0.1	0.1	0.2	0.1
$T2\&S1$	0.1	0.2	0.1	0.1
$T2\&S2$	0.5	0.4	0.3	0.3

Should the wildcatter pay for the test or not? Then, should he drill or not?

A Simple Example: Decision Tree



Outline

Normal Form Decision

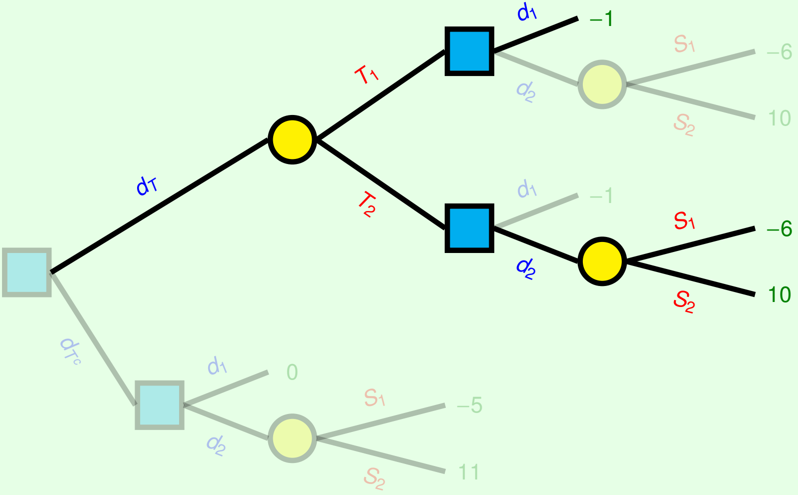
Definition

A normal form decision fixes at every decision node exactly one decision.

Observation

in a sequential decision problem,
each normal form decision corresponds to a gamble

Normal Form Decision: Example



	$T_1 S_1$	$T_1 S_2$	$T_2 S_1$	$T_2 S_2$
$d_T(T_1 d_1)(T_2 d_2)$	−1	−1	−6	10

Normal Form Decision: Example

- We can find the gamble for every normal form decision:

	T_1S_1	T_1S_2	T_2S_1	T_2S_2
$d_T d_1$	-1	-1	-1	-1
$d_T(T_1 d_1)(T_2 d_2)$	-1	-1	-6	10
$d_T(T_1 d_2)(T_2 d_1)$	-6	10	-1	-1
$d_T d_2$	-6	10	-6	10
$d_{T^c} d_1$	0	0	0	0
$d_{T^c} d_2$	-5	11	-5	11

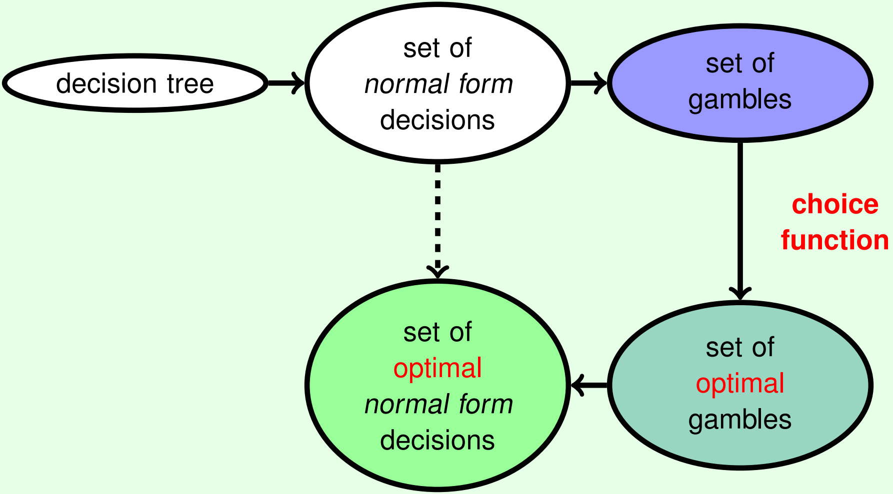
- . . . so, we have
  - a set of normal form decisions
  - a gamble for each normal form decision
  - a credal set, so we can calculate lower/upper previsions of gambles and of their differences

. . . everything keeps working as before!!

(except that now we have normal form decisions, instead of simple decisions)

Outline

The Problem of Sequential Choice in Normal Form



Example: Normal Form Solution

```
> source("improb-redux.r")
> pmfs = c(
+   0.3, 0.1, 0.1, 0.5,
+   0.3, 0.1, 0.2, 0.4,
+   0.4, 0.2, 0.1, 0.3,
+   0.5, 0.1, 0.1, 0.3)
> getexpectations = getexpectationsfunc(4, pmfs) # 4 = size of possibility
> getlowerprevisions = getlowerprevisionsfunc(getexpectations)
> getupperprevisions = getupperprevisionsfunc(getexpectations)
> isgammamaximin = isgammamaxisomethingfunc(getlowerprevisions)
> isgammamaximax = isgammamaxisomethingfunc(getupperprevisions)
> isrbayesmaximal = ismaximalfunc(getexpectations, rbayescompare)
> isintervalmaximal = ismaximalfunc(getexpectations, intervalcompare)
> isrbayesadmissible = isrbayesadmissiblefunc(getexpectations)
```

Example: Normal Form Solution

```
> rvars = c(
+   -1, -1, -1, -1,
+   -1, -1, -6, 10,
+   -6, 10, -1, -1,
+   -6, 10, -6, 10,
+   0, 0, 0, 0,
+   -5, 11, -5, 11)
> getexpectations(rvars)
     [,1] [,2] [,3] [,4]
[1,] -1.0 -1.0 -1.0 -1.0
[2,]  4.0  2.4  1.8  1.8
[3,] -1.4 -1.4 -0.8 -2.4
[4,]  3.6  2.0  2.0  0.4
[5,]  0.0  0.0  0.0  0.0
[6,]  4.6  3.0  3.0  1.4
> isgammamaximin(rvars)
[1] FALSE  TRUE FALSE FALSE FALSE FALSE
> isgammamaximax(rvars)
[1] FALSE FALSE FALSE FALSE FALSE  TRUE
> isintervalmaximal(rvars)
[1] FALSE  TRUE FALSE  TRUE FALSE  TRUE
> isrbayesmaximal(rvars)
[1] FALSE  TRUE FALSE FALSE FALSE  TRUE
> isrbayesadmissible(rvars)
[1] FALSE  TRUE FALSE FALSE FALSE  TRUE
```

Why **Not** Solve Sequential Problems This Way?

- the number of normal form decisions becomes pretty large very quickly
- a lot of calculations required, particularly with maximality criteria
- for larger problems, not even manageable by computer

the good news...

- there are backward induction algorithms
  that give an answer for any choice function

the bad news...

- but these algorithms only yield the actual optimal normal form solution
  if the choice function satisfies rather restrictive properties

but not all is lost!

- robust Bayes maximality & admissibility satisfy these properties!!
- however (almost) no other criterion does

Outline

Backward Induction

idea: use solutions of subtrees to eliminate options in the full tree

further reading:

- Seidenfeld 1988 [13] [14] (extensive form)
- Harmanec 1999 [6] (extensive form)
- De Cooman & Troffaes 2005 [4] (normal form)
- Kikuti et. al 2005 [8] (apparently, normal form)
- Huntley & Troffaes 2008 [7] (normal form)

Normal Form Backward Induction

(Huntley & Troffaes, 2008 [7])

Recipe (Normal Form Backward Induction)

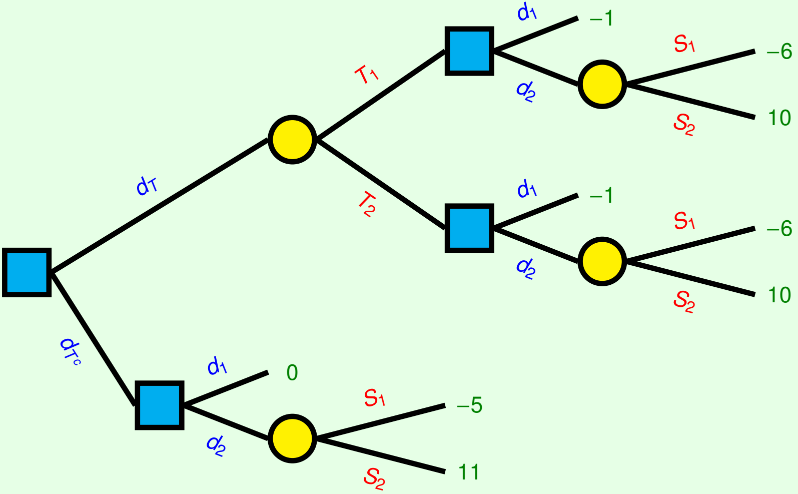
reiterate these steps, until all nodes have been dealt with:

1. find normal form decisions, and corresponding gambles, at final nodes
2. apply choice function conditional on past events, on each set of gambles
3. replace each final node by its set of optimal gambles

Theorem

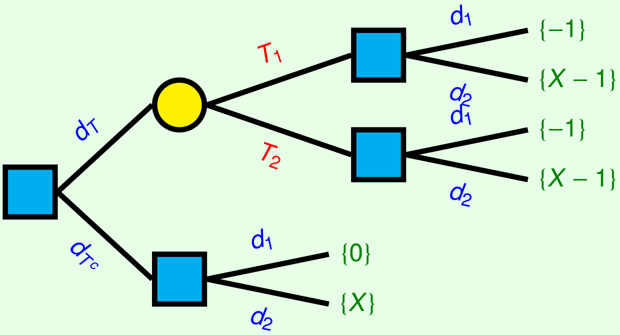
If you do this using robust Bayes maximality as optimality criterion, then you are guaranteed to end up with the optimal normal form decisions at the root.

Normal Form Backward Induction: Example



Normal Form Backward Induction: Example (Stage 1)

1. normal form decisions, and corresponding gambles: trivial
2. apply conditional choice: trivial (single gamble for each node!)
3. replace nodes with sets of optimal gambles



with $X = \begin{array}{cc} S_1 & S_2 \\ \hline -5 & 11 \end{array}$

Normal Form Backward Induction: Example (Stage 2)

T_1 branch

1. normal form decisions and gambles

	S_1	S_2
d_1	-1	-1
d_2	-6	10

2. robust Bayes maximality conditional on T_1
   - apply the definition of conditional probability on each of the given unconditional probabilities

$\mathcal{M}|T_1 =$

	p_1	p_2	p_3
S_1	0.750	0.667	0.833
S_2	0.250	0.333	0.167

   - apply robust Bayes maximality using the resulting conditional probabilities

	S_1	S_2	p_1	p_2	p_3
S_1			0.750	0.667	0.833
S_2			0.250	0.333	0.167
d_1	-1	-1			
d_2	-6	10			

Normal Form Backward Induction: Example (Stage 2)

T_2 branch

1. normal form decisions and gambles

	S_1	S_2
d_1	-1	-1
d_2	-6	10

2. robust Bayes maximality conditional on T_2
   - apply the definition of conditional probability on each of the given unconditional probabilities

$\mathcal{M}|T_2 =$

	p_1	p_2	p_3
S_1	0.167	0.333	0.250
S_2	0.833	0.667	0.750

   - apply robust Bayes maximality using the resulting conditional probabilities

	S_1	S_2	p_1	p_2	p_3
S_1			0.167	0.333	0.250
S_2			0.833	0.667	0.750
d_1	-1	-1			
d_2	-6	10			

Normal Form Backward Induction: Example (Stage 2)

d_{T^c} branch

1. normal form decisions and gambles

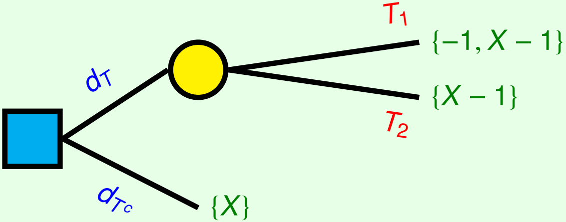
	T_1S_1	T_1S_2	T_2S_1	T_2S_2
d_1	0	0	0	0
d_2	−5	11	−5	11

2. robust Bayes maximality
   - no past events, so use unconditional probabilities
   - apply robust Bayes maximality as usual

	T_1S_1	T_1S_2	T_2S_1	T_2S_2	p_1	p_2	p_3	p_4
$T1\&S1$					0.3	0.3	0.4	0.5
$T1\&S2$					0.1	0.1	0.2	0.1
$T2\&S1$					0.1	0.2	0.1	0.1
$T2\&S2$					0.5	0.4	0.3	0.3
d_1	0	0	0	0				
d_2	−5	11	−5	11				

Normal Form Backward Induction: Example (Stage 2)

3. replace nodes with sets of optimal gambles



Normal Form Backward Induction: Example (Stage 3)

d_T branch

1. normal form decisions and gambles

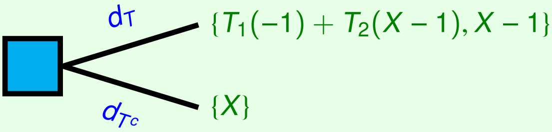
	T_1S_1	T_1S_2	T_2S_1	T_2S_2
$s_1 = d_T(T_1d_1)(T_2d_2)$	-1	-1	-6	10
$s_2 = d_Td_2$	-6	10	-6	10

2. robust Bayes maximality
   - no past events, so use unconditional probabilities
   - apply robust Bayes maximality as usual

	T_1S_1	T_1S_2	T_2S_1	T_2S_2	p_1	p_2	p_3	p_4
$T1\&S1$					0.3	0.3	0.4	0.5
$T1\&S2$					0.1	0.1	0.2	0.1
$T2\&S1$					0.1	0.2	0.1	0.1
$T2\&S2$					0.5	0.4	0.3	0.3
s_1	-1	-1	-6	10				
s_2	-6	10	-6	10				

Normal Form Backward Induction: Example (Stage 3)

3. replace nodes with sets of optimal gambles



explaining the notation $T_1(-1) + T_2(X-1)$:

	T_1S_1	T_1S_2	T_2S_1	T_2S_2
T_1	1	1	0	0
-1	-1	-1	-1	-1
T_2	0	0	1	1
$X-1$	-6	10	-6	10
$T_1(-1)$	-1	-1	0	0
$T_2(X-1)$	0	0	-6	10
$T_1(-1) + T_2(X-1)$	-1	-1	-6	10

Normal Form Backward Induction: Example (Stage 4)

root node

1. normal form decisions and gambles

	T_1S_1	T_1S_2	T_2S_1	T_2S_2
$s_1 = d_T(T_1d_1)(T_2d_2)$	-1	-1	-6	10
$s_2 = d_Td_2$	-6	10	-6	10
d_2	-5	11	-5	11

2. robust Bayes maximality
   - no past events, so use unconditional probabilities
   - apply robust Bayes maximality as usual

	T_1S_1	T_1S_2	T_2S_1	T_2S_2	p_1	p_2	p_3	p_4
$T1\&S1$					0.3	0.3	0.4	0.5
$T1\&S2$					0.1	0.1	0.2	0.1
$T2\&S1$					0.1	0.2	0.1	0.1
$T2\&S2$					0.5	0.4	0.3	0.3
s_1	-1	-1	-6	10				
s_2	-6	10	-6	10				
$d_{T^c}d_2$	-5	11	-5	11				

Normal Form Backward Induction: Example (Stage 4)

3. replace nodes with sets of optimal gambles

$$\{T_1(-1) + T_2(X-1), X\}$$

- these gambles correspond to the normal form decisions:

$$d_T(T_1 d_1)(T_2 d_2),\ d_{T^c} d_2$$

- we have solved the problem!!

Outline

What's Next...

Things I have **not** told you today:

- relationships between choice functions [16]
- more choice functions
  - E-admissibility [10]
  - info-gap, satisficing [1]
  - extensive form methods [13] [14] [6]
- nasty properties of some choice functions:
  do not rely on your general intuition about optimality
- clever things you can do when your decision problem has additional structure

we've only scratched the surface, but hopefully you have learnt something, and have some idea of how decisions could be made under severe uncertainty, and where to look further

References I

[1] Yakov Ben-Haim.
Information Gap Decision Theory: Decisions Under Severe Uncertainty.
Academic Press, 2001.

[2] James O. Berger.
The robust Bayesian viewpoint.
In J. B. Kadane, editor, *Robustness of Bayesian Analyses*, pages 63–144. Elsevier Science, Amsterdam, 1984.

[3] Marquis de Condorcet.
Essai sur l'Application de l'Analyse à la Probabilité des Décisions Rendues à la Pluralité des Voix.
L'Imprimerie Royale, Paris, 1785.

[4] Gert de Cooman and Matthias C. M. Troffaes.
Dynamic programming for deterministic discrete-time systems with uncertain gain.
International Journal of Approximate Reasoning, 39(2–3):257–278, June 2005.

[5] Itzhak Gilboa and David Schmeidler.
Maxmin expected utility with non-unique prior.
Journal of Mathematical Economics, 18(2):141–153, 1989.

References II

[6] David Harmanec.
A generalization of the concept of Markov decision process to imprecise probabilities.
In G. de Cooman, F. G. Cozman, S. Moral, and P. Walley, editors, *ISIPTA '99 – Proceedings of the First International Symposium on Imprecise Probabilities and Their Applications*, pages 175–182, Ghent, 1999. Imprecise Probabilities Project.

[7] Nathan Huntley and Matthias C. M. Troffaes.
An efficient normal form solution to decision trees with lower previsions.
In Didier Dubois, M. Asunción Lubiano, Henri Prade, María Ángeles Gil, Przemyslaw Grzegorzewski, and Olgierd Hryniewicz, editors, *Soft Methods for Handling Variability and Imprecision*, Advances in Soft Computing, pages 419–426. Springer, Sep 2008.

[8] Daniel Kikuti, Fabio G. Cozman, and Cassio P. de Campos.
Partially ordered preferences in decision trees: Computing strategies with imprecision in probabilities.
In Ronen Brafman and Ulrich Junker, editors, *Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pages 118–123, 2005.

[9] H. E. Kyburg.
Rational belief.
Technical report, University of Rochester, 1983.

[10] Isaac Levi.
The Enterprise of Knowledge. An Essay on Knowledge, Credal Probability, and Chance.
MIT Press, Cambridge, 1980.

References III

[11] Blaise Pascal.
Pensées.
Maxi-Livres, Paris, 2001.
Unfinished work, published posthumously from collected fragments. First incomplete edition: Port-Royal, 1670. First complete reproduction: Michaut, Basle, 1896.

[12] Jay K. Satia and Jr. Roy E. Lave.
Markovian decision processes with uncertain transition probabilities.
Operations Research, 21(3):728–740, 1973.

[13] Teddy Seidenfeld.
Decision theory without ‘independence’ or without ‘ordering’: What is the difference?
Economics and Philosophy, 4:267–290, 1988.

[14] Teddy Seidenfeld.
A contrast between two decision rules for use with (convex) sets of probabilities: Gamma-maximin versus E-admissibility.
Synthese, 140(1–2):69–88, May 2004.

[15] Amartya Sen.
Social choice theory: A re-examination.
Econometrica, 45(1):53–89, January 1977.

References IV

[16] Matthias C. M. Troffaes.
Decision making under uncertainty using imprecise probabilities.
International Journal of Approximate Reasoning, 45(1):17–29, May 2007.

[17] Abraham Wald.
Contributions to the theory of statistical estimation and testing hypotheses.
The Annals of Mathematical Statistics, 10(4):299–326, December 1939.

[18] Abraham Wald.
Statistical decision functions which minimize the maximum risk.
The Annals of Mathematics, 46(2):265–280, 1945.

[19] Peter Walley.
Statistical Reasoning with Imprecise Probabilities.
Chapman and Hall, London, 1991.

Outline

Exercise 1: Different Approaches to Decision Making

Under what circumstances would you use any of the following statistical approaches to decision making?

1. frequentist: identify Wald's admissible decisions
2. Bayesian: maximise posterior expected utility
3. robust Bayesian: maximise posterior expected utility but also check the sensitivity of your conclusion against changes in the prior

Discuss your answer with your neighbour.

Exercise 2: Verify Wald's Theorem in the Boat Example

Consider again the boat example that we discussed.

utility function

$U(d,x)$	$x=0.5$	$x=2$
$d=$ boat	3	-1
$d=$ no boat	0	0

likelihood

$p(y\|x)$	$y=0.5$	$y=2$
$x=0.5$	0.9	0.1
$x=2$	0.3	0.7

We established that the following strategies were optimal Wald strategies:
We established that $\delta(y=0.5)=$ boat, $\delta(y=2)=$ no boat was an optimal Bayes strategy under prior $p(x=0.5)=0.4$, $p(x=2)=0.6$.

Verify that you get all Wald's admissible decisions by varying the prior.

Hint: repeat the analysis from the lecture for the following two priors:

$p(x)$	$x=0.5$	$x=2$
	1	0

$p(x)$	$x=0.5$	$x=2$
	0	1

Exercise 3: Machinery, Overtime, or Nothing?

Consider again the same very simple example. We have done additional market research, and we now know that demand will increase with probability at least 0.6, and at most 0.65.

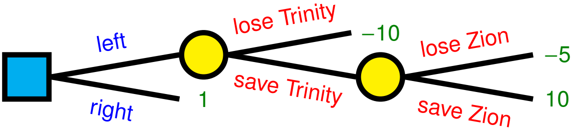
What advice can we give the manager now? Investigate with each optimality criterion.

Hint: $\mathcal{M} =$

	p_1	p_2
increase	0.6	0.65
stay	0.4	0.35

Exercise 4: Saving Zion (Or Maybe Not?)

There are two doors. The door to your right leads to the Source and the salvation of Zion. The door to your left leads back to the Matrix, to her. . . and to the end of your species. As you adequately put, the problem is choice. But we already know what you are going to do, don't we?



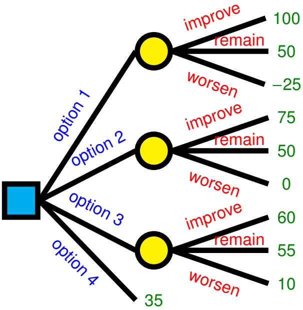
$\mathcal{M} =$

	p_1	p_2	p_3
lose Trin	0.1	0.4	0.3
save Trin & lose Zion	0.45	0.3	0.2
save Trin & save Zion	0.45	0.3	0.5

Left, or right? Investigate with your favorite optimality criterion.

Exercise 5: A Risky Investment

You have the option to invest some money. The market can either improve, remain, or worsen. The set of probabilities for your lower prevision are tabulated below. You have the choice between 4 options, summarized in the decision tree below.



$\mathcal{M} =$

	p_1	p_2
improve	0.0	0.3
remain	0.6	0.3
worsen	0.4	0.4

Which options should you definitely not consider? First consider interval maximality, then consider robust Bayes maximality. Which of these two criteria gives the better answer?

Exercise 6 (*)

Let X be any gamble, with lower prevision L and upper prevision U. Let c be any constant. Suppose you have the choice between the uncertain gain X, or the certain gain c.

Under each of the criteria, determine which of X or c (or both!) are optimal, under the following circumstances:

- $c < L$
- $L < c < U$
- $c > U$

In Exercise 5, option 4 corresponds to an investment without risk, as it yields the value $c = 35$ independently of the market, however, we found that this value was too low relative to the other options to be optimal.

For what values for c would you change your mind? Again, investigate this using each of the criteria, for $c < 37$, $37 < c < 37.5$, $37.5 < c < 38.5$, and $38.5 < c$.

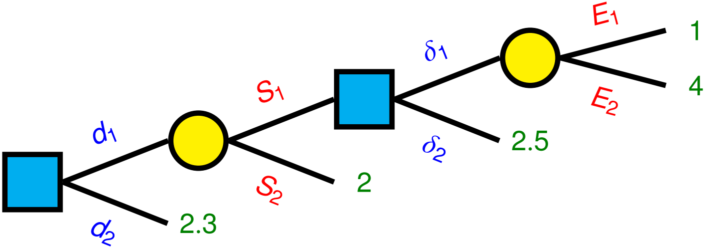
Exercise 7

State one advantage, and one disadvantage, of solving a sequential decision problem by normal form backward induction, compared to solving it by normal form.

Can you think of a situation in which normal form backward induction would be less efficient than normal form?

Exercise 8

Solve the following sequential decision problem for robust Bayes maximality, using either normal form, or normal form backward induction.

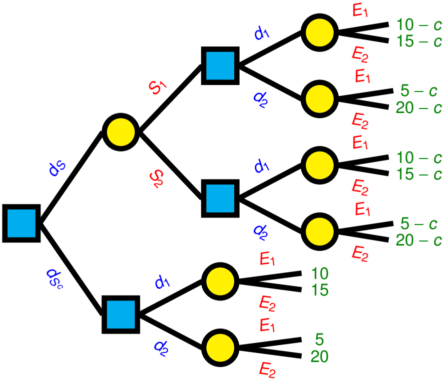


$$\mathcal{M} = \begin{array}{c|cc} & p_1 & p_2 \\ \hline S_1 E_1 & 0.2 & 0.1 \\ S_1 E_2 & 0.3 & 0.4 \\ S_2 & 0.5 & 0.5 \end{array}$$

$$\left[\text{Hint: } \mathcal{M}|S_1 = \begin{array}{c|cc} & p_1 & p_2 \\ \hline E_1 & 0.4 & 0.2 \\ E_2 & 0.6 & 0.8 \end{array}\right]$$

Exercise 9 (*)

Tomorrow, a subject is going for a walk in the lake district. It may rain (E_1), or not (E_2). The subject can either take a waterproof (d_1), or not (d_2). But the subject may also choose to buy today's newspaper, at cost c, to learn about tomorrow's weather forecast (d_S), or not (d_{S^c}), before leaving for the lake district. The forecast has two possible outcomes: predicting rain (S_1), or not (S_2). *Solve for robust Bayes maximality, with $c = 1$.*



$\mathcal{M} =$

	p_1	p_2	p_3	p_4
$S_1 E_1$	0.378	0.378	0.378	0.478
$S_1 E_2$	0.162	0.162	0.262	0.162
$S_2 E_1$	0.072	0.172	0.072	0.072
$S_2 E_2$	0.388	0.288	0.288	0.288

Exercise 10 (**)

Consider again the lake district exercise.

For which values of c is it no longer robust Bayes maximal to buy the newspaper?

(This is the value of information of the newspaper.)