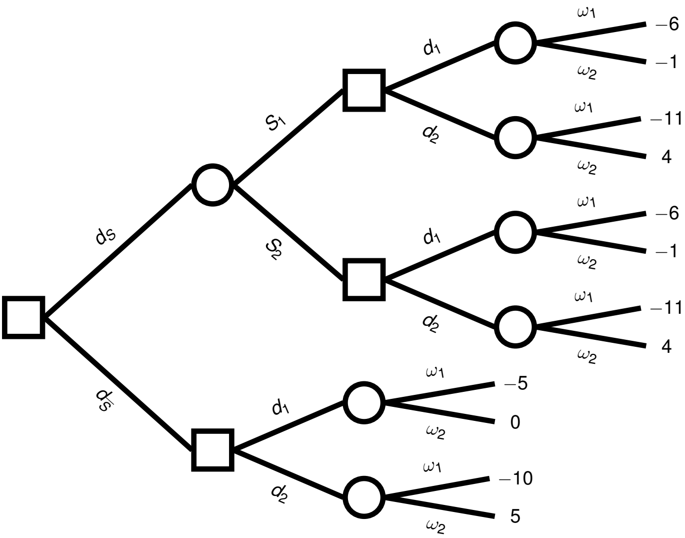
Decision Trees and Imprecise Probabilities

Nathan Huntley Matthias Troffaes

Durham University

6th September 2010

Example



Normal Form and Backward Induction

Normal Form

- Turn tree into set of gambles and apply preferred choice function to those
- Uses unconditional lower prevision

Backward Induction

- Find sets of gambles at ultimate decision nodes
- Apply choice function here (conditional)
- Remove redundant arcs
- Move to next layer of decision nodes

Strategic Equivalence

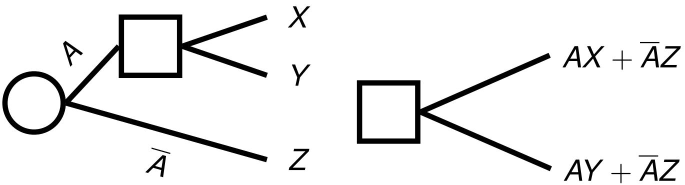
Strategic Equivalence

- Can represent any non-trivial decision tree in a number of ways
- Most simply, only have one decision node, with each decision arc representing a normal form gamble
- Two trees inducing the same set of gambles are called strategically equivalent
- It would be nice if a solution method respects strategic equivalence

Normal Form and Backward Induction

- By definition, the normal form solution preserves strategic equivalence
- Backward Induction may not

Example: Interval Dominance



Assessments

	A	$\overline{A}$
X	1	1
Y	1.5	3.5
Z	0	4

	$\underline{P}(\cdot\|B)$	$\overline{P}(\cdot\|B)$
X	1	1
Y	2	3
Z	1	3

	$\underline{P}$	$\overline{P}$
$BX + \overline{B}Z$	1	2
$BY + \overline{B}Z$	1.5	3

Interval Dominance: A Problem?

Discussion

- In this case, the problem is just the elimination of some interval dominant gambles
- Should we be worried about losing these? After all, interval dominance is rather indecisive anyway

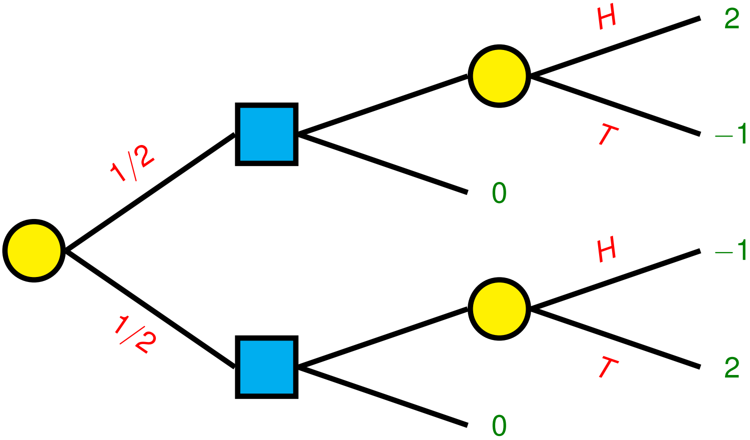
Theorem

For any decision tree, the set of gambles found by interval dominance backward induction is always a subset of the interval dominant normal form gambles.

Theorem

For any decision tree, the set of gambles found by interval dominance always includes every maximal normal form gamble.

Example: Γ-maximin



Γ-maximin: A Problem

Discussion

- Backward induction forces us to choose a strategy that is clearly inferior to another available strategy
- Using backward induction for Γ-maximin is not to be recommended in general
- It could be claimed that this example shows that Γ-maximin is a flawed choice function

Does Anything Work?

Theorem (Backward Induction Theorem)

Backward induction using a choice function opt *agrees with the normal form solution if and only if*

- *Backward Conditioning Property. If $AX = AY$ and $\{X, Y\} \subseteq \mathcal{X}$, then $X \in \mathrm{opt}(\mathcal{X}|A) \iff Y \in \mathrm{opt}(\mathcal{X}|A)$, subject to some technicalities*
- *Path Independence.*

$$\mathrm{opt}\left(\bigcup_{i=1}^{n} \mathcal{X}_i \middle| A\right) = \mathrm{opt}\left(\bigcup_{i=1}^{n} \mathrm{opt}(\mathcal{X}_i|A) \middle| A\right)$$

- *Backward Mixture Property.*

$$\mathrm{opt}\left(\{AX + \overline{A}Z \colon X \in \mathcal{X}\}|B\right) \subseteq \{AX+\overline{A}Z \colon X \in \mathrm{opt}(\mathcal{X}|A\cap B)\}$$

Strategic Equivalence and Backward Induction

Respecting Strategic Equivalence?

- If backward induction gives the normal form solution, then clearly it respects strategic equivalence
- If PI or BMP fail, then backward induction does not respect strategic equivalence
- The BCP does not influence things

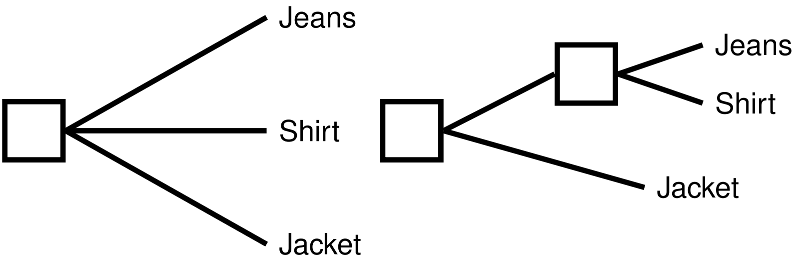
Role of Conditioning Property

- If opt satisfies PI and BMP but not BCP, then

$$\text{Optimal nfds} \supseteq \text{Backward induction nfds},$$

but for any optimal normal form decision, there is a backward induction normal form decision with the same gamble.

Problems Are Still Here...



Subtree Perfectness

Definition

- Consider a subtree U of a decision tree T
- We could solve T and then restrict the solution to U
- We could solve U (conditional on any preceding events in T)
- If the two approaches coincide for any T and U, the solution method is subtree perfect

Subtree Perfectness Theorem

Theorem

The normal form solution induced by a choice function opt is subtree perfect if and only if it satisfies:

- Conditioning property. If $\{X, Y\} \subseteq \mathcal{X}$ and $AX = AY$, then

$$X \in \mathrm{opt}(\mathcal{X}|A) \iff Y \in \mathrm{opt}(\mathcal{X}|A)$$

- Intersection property. If $\mathcal{Y} \subseteq \mathcal{X}$ and $\mathrm{opt}(\mathcal{X}|A) \cap \mathcal{Y} \neq \emptyset$, then

$$\mathrm{opt}(\mathcal{Y}|A) = \mathrm{opt}(\mathcal{X}|A) \cap \mathcal{Y}$$

- Mixture property.

$$\mathrm{opt}(\{AX + \overline{A}Z : X \in \mathcal{X}\}|B) = \{AX + \overline{A}Z : X \in \mathrm{opt}(\mathcal{X}|A \cap B)\}$$

Improvements in simpler problems

simple decision tree

What is subtree perfect here?

- Not Γ-maximin
- Not interval dominance
- Don't know about maximality
- E-admissibility is

E-admissibility

Let $\mathcal{M}$ be a set of probability mass functions. A gamble X is E-admissible in a set $\mathcal{X}$ if there is a $p \in \mathcal{M}$ such that X maximizes expected utility in $\mathcal{X}$.

Even better...

Useful Restriction On $\underline{P}$

- Suppose that $\underline{P}$ satisfies, for all relevant gambles,

$$\underline{P}(X) = \underline{P}(\underline{P}(X|\mathcal{A}))$$

- Then maximality starts working for the simple problem
- So does Γ-maximin

Even even better

- If the above holds for all relevant gambles and partitions, then Γ-maximin becomes subtree perfect.
- However, even if it works for one decision tree, I can draw another decision tree where $\underline{P}$ stops working
- So this doesn't make all the bad problems go away.