DECISION TREES AND BACKWARD INDUCTION

DIRECTED GRAPHS

Partial Order (PO) $\prec$ on set X: A transitive relation such that $\forall x \neq y$ at most one of $x \prec y,\ y \prec x$ holds. In the following $\prec$ will be a PO.

Example: Inclusion relation $\subset$ on the set of subsets of a given set S.

Observation 1. *In $\langle \prec, X \rangle$ there are no cycles ($x_1 \prec \cdots \prec x_n \prec x_1$).*

Proof. Otherwise by transitivity it would be $x_1 \prec x_n$ and $x_n \prec x_1$. □

$x \in X$ is **maximal** for $\prec$ if $\nexists y \in X$ such that $x \prec y$.

Proposition 1. *In $\langle \prec, X \rangle$ with X finite any x, if not maximal, is below a maximal.*

Proof. If x is not maximal $x \prec x_1$ for some x_1; if this is maximal we are done, otherwise continue thus; the process must stop because in going up we pick distinct elements (no cycles) and X is finite. □

Directed Graph: A pair (V, E) with V finite set ('nodes') and $E \subset V \times V$ ('edges'). For $e = (u, v) \in E$ we say e starts at u and ends at v. We assume that $\forall v,\ (v, v) \notin E$. We draw directed graphs representing edge (u, v) by an arrow going from u to v.
Example: $(\{x, y, w, z\}, \{(x, y), (y, x), (x, z), (z, w)\})$ (draw it).

Path: An ordered n-tuple of edges $(e_1, \ldots, e_n)$ such that e_{i+1} starts where e_i ends. The path with e_1 starting at u and e_n ending at v is denoted $P(u, v)$. It touches $(n + 1)$ nodes, u and the end points of the e_i's in P. Abusing notation we write $w \in P(u, v)$ if w is one of them.

Terminology: If $(u, v) \in E$ then u is v's mother, and if also $(u, w) \in E$ then v and w are sisters. If $\exists P(u, v)$ then v is u's descendant and u is v's ancestor. A node is *terminal* if it has no children. A path $P(u, v)$ is terminal if v is terminal.

TREES

Tree: A directed graph (V, E) with a 'root' node R such that (i) $\nexists (v, R) \in E$, and (ii) $\forall v \in V \setminus \{R\}\ \exists!\ P(R, v)$.

Observation 2. *In a tree there are no loops (paths $P(u, u)$).*

Proof. Otherwise u could be reached from R in two ways (using and not using the loop). □

Proposition 2. *In a tree $\exists P(u, v)$ iff $u \in P(R, v)$.*

Proof. 'If' is clear. 'Only if': if $u \notin P(R, v)$ and $\exists P(u, v)$, v can be reached in two ways (passing and not passing through u). □

Proposition 3. *In a tree there are no paths linking sisters or descendants of sisters.*

Proof. If $P(u, v)$ were such a path then (previous Proposition) $u \in P(R, v)$ hence v could be reached from R through two sisters, contradicting (ii) of definition of tree. □

Thus trees look like trees. The edges are only those from R to her children, from them to their children, and so on. Note also that we don't need to draw the arrowheads, since the direction of any existing edge between u and v is always determined by looking whether $u \in P(R, v)$ or $v \in P(R, u)$ (only one of these alternatives can hold, otherwise u could be reached by passing or not passing through v). To complete the formal representation of the trees we draw we show that each node has a terminal descendant; we'll then see how terminal parts of a tree look like.

Date: January 7, 2009.

Lemma 1. *The set of paths in a tree is finite.*

Proof. In each path all nodes are distinct (no loops), hence the family of all paths is contained in that of all subsets of V, which is finite (we are using the result that a subset of a finite set is finite, which is quite obvious intuitively but not as obvious to prove formally). □

Now partially order the set of paths by inclusion: $P \prec P'$ if P is part of P', formally if $P = (e_1, \ldots, e_n)$ and $P' = (\epsilon_1, \ldots, \epsilon_h, e_1, \ldots, e_n, \eta_1, \ldots, \eta_k)$ with $h+k > 0$ (check that this $\prec$ is indeed a partial order). Note that by construction $\prec$-maximal paths end in terminal nodes.

Proposition 4. *Any node has a terminal descendant.*

Proof. Any node u is in $P(R, u)$, which is contained in a $\prec$-maximal path by Proposition 1. □

Flower: A mother and her children, when children are all terminal.

Proposition 5. *A tree has flowers.*

Proof. The *length* of a path is the number of the edges defining it. Since the set of paths is finite, the set of path lengths is a finite set of integers. Letting L be their maximum, any u such that $P(R, u)$ has length $L - 1$ together with her children form a flower. □

DECISION AND BACKWARD INDUCTION

Idea is that any path ending in a terminal node results in a consequence, and that on the set on consequences there is a preference relation represented by a utility function. The aim of the decision maker is to choose, at root node, a path ending in a most preferred consequence. So:

Decision Tree: A tree with a payoff, in utility units, attached to its terminal nodes. A *strategy* is choosing a child at *each* node. A strategy thus determines a terminal path from R.

Optimal Strategy: A strategy which determines a path from R ending in a terminal node with maximal payoff. Such paths are *optimal paths*.

Branch (or Subtree): A node of a tree and all her descendants, with edges from the original tree. Denote it by T_u if the node is u and the original tree is T.

Proposition 6. *T_u is a tree with root u.*

Proof. v is in T_u if $\exists P(u, v)$, i.e. by Proposition 2 iff $u \in P(R, v)$, so in the definition of tree u satisfies (i) because then $v \notin P(R, u)$ (already observed) and (ii) because more than one path from u to v would imply more than one from R to v. □

Proposition 7. *If $P(R, v)$ is optimal in T and $u \in P(R, v)$, then $P(u, v)$ is optimal in T_u.*

Proof. If $P(u, v')$ were better in T_u, then $P(R, v')$ would be better in T. □

Observation 3. *Note that Proposition 7 can be rephrased thus: if $P(u, v)$ is suboptimal in T_u, then $P(R, v)$ is suboptimal in T.*

Backward Induction. Thus, to find an optimal strategy in T chop each flower and assign to the mother the payoff of her best child (or one of them if there are several); do the same in the resulting tree, and continue thus. By so doing you only eliminate suboptimal paths, and also reduce the maximum length of the remaining paths. You end up with a flower based at R, and a best petal is the first step of your optimal strategy. The optimal continuation remains determined in the previous stages of the process: if the first step is (R, u), the second is the best first in T_u, and so on. To visualize, or prove it formally by induction, start with a tree with maximun path length equal to 2: at first stage you determine the best final choice conditional on reaching one of R's children, at second stage you find the optimal path.

In Practice. Prescription is, when faced with a sequential decision process ask yourself what you would do next if you initially chose an alternative. In other words, look forward before you move. *Example*: if you are on a scooter approaching a red traffic light and the cars you see aren't moving, if you don't want to get stuck look forward at the first line of cars, then find backwards a clear way down to it.

UNCERTAINTY UNDER EXPECTED UTILITY

Uncertainty concerns the consequences of your actions. In a tree, at some nodes it's your turn, at some others it is Chance who moves. So:

Decision Tree with Uncertainty. A decision tree where each node has a name attached, either *You* or *Chance*. Conventionally your nodes ('decision nodes') are drawn as squares and Chance's as circles. Payoffs accrue to you. A strategy (of you) is as before choosing a child at each of your nodes. At a chance node Chance chooses a child drawing from a probability distribution on the node's children. Drawings at different chance nodes are independent. The root node is your node (there may be past chance events, but you are interested in the situation when you come into play). A chance flower (a flower based at a chance node) is then a lottery on utility payoffs.

Expected Utility. Under Expected Utility you evaluate a chance flower by taking expectation. That is, if payoff is π_i with probability $p_i,\ i = 1, \ldots, n$ then the value of the flower (i.e. of its base node) is $\sum_i p_i \pi_i$. More generally, by the same formula you evaluate a distribution on terminal paths if path i occurs with probability p_i and yields payoff π_i.

Example: At R, either you go up to a terminal node, or down to a chance flower.

Example: Two stages; draw the tree with edges

$$(R, D), (R, c), (c, u), (c, d), (u, x_u), (u, y_u), (d, x_d), (d, y_d)\,,$$

and suppose c belongs to Chance. Here if you go to D get a sure payoff, but if you go to c it's Chance turn before your next decision.

In decision trees with uncertainty, as in the last example, a strategy does not determine a terminal path as it is the case with no uncertainty. There strategies and terminal paths are effectively the same thing; here a terminal path will be co-determined by your strategy and Chance. Since Chance chooses a probability distribution on children at each of her nodes, a strategy of yours determines a (possibly degenerate) probability distribution on terminal paths, hence on payoffs, as in the last example (in general there are more than two stages; by independence probabilities at successive chance nodes must be multiplied).

Optimal Strategy. A strategy which determines a probability on terminal paths from R with maximal expected utility.

As in the case of certainty we want to find an optimal strategy by backward induction, but there is a slight complication which will be understood by looking at the last example. The problem is that whereas with no uncertainty we may talk about optimal paths because they are the same thing as optimal strategies, with uncertainty we have to deal with path distributions, so Proposition 7 is not adequate. Incidentaly, note that since a strategy is a function assigning a child to each of your nodes there are finitely many possible strategies, so an optimal one always exists. Let π be the function which assigns payoffs to terminal nodes.

Last Example Continued: Suppose payoffs are

$$\pi(x_u) = \pi(x_d) = 10,\ \pi(y_u) = \pi(y_d) = 0,\ \pi(D) = 100\,.$$

Then an optimal strategy is the S defined by

$$S(R) = D,\ S(u) = y_u,\ S(d) = y_d\,.$$

Here it doesn't matter that you choose suboptimal moves at u and d, for under S they won't be reached. Note that this S will not be discovered by backward induction. On the other hand, we do no harm if we modify S by taking optimal continuations in the unreached nodes; that is, if we define S' by

$$S'(R) = D,\ S'(u) = x_u,\ S'(d) = x_d$$

we still have an optimal strategy. And this is what we need to legitimate use of the backward induction method to find an optimal strategy.

Proposition 8. *There is an optimal strategy whose continuations are optimal in all subtrees based at decision nodes.*

Proof. Let S be an optimal stategy. If u is a decision node which is reached with positive probability under S (i.e. is along a path which has positive probability under S), then the continuation of S from u must be optimal in T_u by an argument analogous to that proving Proposition 7.[1] If u is unreached under S, replace the continuation of S in T_u by an optimal continuation. The resulting S' has the same expected utility of S (because zero-probability events do not affect expectations), and has the property we want. □

Backward Induction with Uncertainty. By the last Proposition we can find an optimal strategy much like we did under certainty. Replace chance flowers with their expected utility; then replace decision flowers with their best petals' payoff; then start again with chance flowers of the resulting tree, and go on like this until you get a root flower to pick your optimal first move. The optimal continuation is determined in the previous stages as in the certainty case.

In Practice under Uncertainty. Prescription is the same: look forward before moving.

Salvatore Modica, Università di Palermo
E-mail address: modica@unipa.it

[1] The argument only involves writing down the fact that probabilities on paths on subtrees are rescalings of probabilities on original paths and then adding up, but you will do this in the Game Theory Course, so I spare you a possibly different notation.