

Multi-Agent Systems - Lecture 10 (10.01.06)

Summarized by Itai Ashlagi

1 On the Foundations

This lecture is based on the paper: On The Foundations of Qualitative Decision Theory (by Brafman and Tennenholtz 96).

Work on the foundations is motivated by two major applications: agent modeling and decision making. Generally, we would like to know, when one can assign a utility function to an agent, i.e. when do there exist numbers which capture the behavior of the agent. Here we look at qualitative decision makers, and in particular the *maximin* criteria.¹

1.1 The Model

We begin by defining a standard agent model and a policy, which describes an agent's behavior.

1. S - set of possible states in the environment.
2. A - set of possible actions.
3. $L = 2^S \setminus \{\emptyset\}$ - set of possible information states (e.g. in $l = \{s_1, s_2\}$ the state is either s_1 or s_2).

Each state in S describes one possible state of the world, or the agent's environment. Intuitively, information states correspond to the agent's knowledge (about the true state).

The model considers one shot decision making. For every information state the agent chooses a possible action. This function is called a *policy*. Formally

Definition 1 A policy is a function $P : L \rightarrow A$.

Since the number of information states is $2^{|S|}$ it is unrealistic to supply this method explicitly. We would like some method that will capture a policy implicitly. We will use a qualitative representation.

¹A lot of work has been done on quantitative approaches to decision making, usually based on the expected utility maximization, e.g. (Savage 72).

Definition 2 A policy P has a maximin representation if there exists a utility function on $A \times S$ such that for every $l \in L$ we have

$$P(l) = \arg \max_{a \in A} \min_{s \in PW(l)} \{u(a, s)\}.$$

Here $PW(l)$ denotes the set of states which belong to the information state l .

1.2 Binary Policies

In this section we show a representation theorem for agents with two possible actions. The following is a basic property of maximin policies:

Definition 3 We say that a policy P is closed under unions if $P(u) = P(w)$ implies that $P(u \cup w) = P(u)$ for every $u, w \subseteq S$.

For example, suppose $S = \{c, o, h\}$. Hence $L = \{\{c\}, \{o\}, \{h\}, \{c, o\}, \{c, h\}, \{o, h\}, \{c, o, h\}\}$. Let $A = \{Y, N\}$. Let P be the policy in which chooses Y for the information states $\{c\}$ and $\{c, o\}$, and chooses N for every other information state. It is easy to see that P is closed under unions. Observe that the following utilities represent (maximin) the policy P :

	c	o	h
Y	3	2	0
N	1	3	3

The following theorem shows that closure under unions is sufficient for a given policy, to be maximin representable.

Theorem 1 Let P be a policy assigning only one of two possible actions at each information state, and assume that P is closed under unions. Then P is maximin representable.

Before the proof we need some preparations.

Definition 4 $u >_P w$ where $u, w \subseteq S$, if $P(u \cup w) = P(u)$ and $P(u) \neq P(w)$.

Claim 1: Let P be a binary policy closed under unions. If $u_1 >_P u_2 >_P \dots >_P u_k$ then $P(u_1) \neq P(u_k)$ implies that $u_1 >_P u_k$.

Proof. We will prove the claim for $k = 4$. Assume that $u >_P v >_P w >_P x$. Let $A = \{a, a'\}$. Suppose w.l.o.g. that $P(u) = a$. Since $u >_P v >_P w >_P x$, $P(v) = a'$, $P(w) = a$ and $P(x) = a'$. We need to show that $P(u \cup x) = a$. Suppose in negation that $P(u \cup x) = a'$. Since $v >_P w$, $P(v \cup w) = a'$. Closure under unions implies that $P(u \cup x \cup v \cup w) = a'$. $u >_P v$ and

$w \succ_P x$ imply that $P(u \cup v) = a$ and $P(w \cup x) = a$ respectively. Closure under unions implies that $P(u \cup v \cup w \cup x) = a$ which is a contradiction. \square

The following algorithm assigns utilities $u(a, s)$, for every action $a \in A$ and every state $s \in S$, to the policy a given policy P . In the proof of Theorem 1 we will show that the utility function provided by the following algorithm is a maximin representation for the given policy.

Algorithm

1. Set i to 0 and T to S .
2. For each $v \in S$ let $u(v, P(v)) = n = |S|$.
3. Let $U = \{s \in T \mid \text{there is no } v \in T \text{ such that } v \succ_P s\}$.
4. Let $u(a, s) = i$ for every $s \in U$ and a such that $a \neq P(s)$.
5. Let $i = i + 1$ and $T = T \setminus U$.
6. If $T = \phi$ terminate. Otherwise go to step 3.

Claim 2: For every information state l there is a unique preferred action.

Proof. Actions are equally preferred only if they have the same worst case payoff. This can be the case only if the set U at this stage contains states s and v , on which P assigns different actions. However in this case $s \succ_P v$ or $v \succ_P s$ (based on the action assigned to $\{s, v\}$), and therefore s and v can not be in U simultaneously. \square

Proof of Theorem 1: We will show that by using maximin for the constructed utilities (by the Algorithm), one obtains behavior identical to the original policy P . Notice that this trivially holds for every l , where $|PW(l)| = 1$ (under certainty). Let $A = \{a, a'\}$.

Let $U \subseteq S$. Suppose that maximin chooses a on U . Then for some $w \in U$ $u(w, a')$ is the worst utility on U (and in w P selects a). We need to show that P selects a on U . Denote by A_a the states in U where a is selected, and by $A_{a'}$ the states in U where a' is selected. Our claim is that if $v \in A_{a'}$, then $w \succ_P v$. To see this, consider our construction process. Since $u(w, a')$ is minimal, w must have been chose in step 3, before all other elements in U . Hence, there is no $v \in A_{a'}$ such that $v \succ_P w$. However, since the action taken on the elements of $A_{a'}$ and w are differen, then we must have $w \succ_P v$. Therefore, $w \succ_P v$ for every $v \in A_{a'}$. This implies that for every $\{v, w\}$ the action chosen is a . $U = A_a \cup A_{a'} \cup \{w\}$. Therefore closure under unions imply that P selects a on U . \square

It turns out that closure under unions is a sufficient condition to minimax-regret and competitive-ratio behaviors.