15326 - Computational Microeconomics

Homework Assignment 4: voting rule properties, decision and game theory (due April 2 before 5pm)

Please read the rules for assignments on the course web page (`http://www.cs.cmu.edu/~15326-s26/`). Use Piazza for questions and Gradescope to turn this in. For all questions, always hand in both code and output, typically .mod and .out files (and do not simply put everything in a .pdf).

Please use clear variable names and write comments in your code where appropriate (you can put comments between `/*` and `*/`, or start a line with `#`).

1. (Properties of voting rules.)

Alice likes to analyze the outcomes of elections; specifically, she is interested in the different outcomes that different voting rules produce on the same votes. To do so, she executes many different rules on the same set of votes, a painstaking process. She likes knowing about properties of voting rules that ease her task. For example, she likes to know which voting rules satisfy the Condorcet criterion, so that if there is a Condorcet winner, she immediately knows that that will be the winner for those rules, without having to go through the trouble of executing each rule individually.

Recently, Alice has become interested in the phenomenon of votes "cancelling out." Let us say that a set[1] S of votes *cancels out with respect to voting rule* r if for **every** set T of votes, the winner[2] that r produces for T is the same as the winner that r produces for $S \cup T$. For example, the set of votes $\{a \succ b \succ c, b \succ a \succ c, c \succ a \succ b\}$ cancels out with respect to the plurality rule: each candidate is ranked first once in this set of votes, so it has no net effect on the outcome of the election. The same set does not cancel out with respect to Borda, though, because from these votes, a gets 4 points, b gets 3, and c gets 2, which may affect the outcome of the election. Alice likes to know when a set of votes cancels out with respect to a rule, so that she can just ignore these votes, easing her computation of the winner.

Define a pair of *opposite votes* to be a pair of votes with completely opposite rankings of the candidates, i.e. the votes can be written as $c_1 \succ c_2 \succ \ldots \succ c_m$

[1] Technically, a multiset, since the same vote may occur multiple times.

[2] ... or set of winners if there are ties.

and $c_m \succ c_{m-1} \succ \ldots \succ c_1$. Let us say that a voting rule r satisfies the *Opposites Cancel Out (OCO)* criterion if every pair of opposite votes cancels out with respect to r.

1a. (12 points) From among the (reasonable[3]) voting rules discussed in class, give 3 voting rules that satisfy the OCO criterion, and 3 that do not (and say which ones are which!).

Define a *cycle* of votes to be a set of votes that can be written as $c_1 \succ c_2 \succ \ldots \succ c_m, c_2 \succ c_3 \succ \ldots c_m \succ c_1, c_3 \succ c_4 \succ \ldots \succ c_m \succ c_1 \succ c_2, \ldots, c_m \succ c_1 \succ c_2 \succ \ldots \succ c_{m-1}$. Let us say that a voting rule r satisfies the *Cycles Cancel Out (CCO)* criterion if every cycle cancels out with respect to r.

1b. (12 points) From among the (reasonable) voting rules discussed in class, give 3 voting rules that satisfy the CCO criterion, and 3 that do not.

Define a pair of *opposite cycles* of votes to be a cycle, plus all the opposite votes of votes in that cycle. Note that these opposite votes themselves constitute a cycle, the opposite of which is the original cycle. Let us say that a voting rule r satisfies the *Opposite Cycles Cancel Out (OCCO)* criterion if every pair of opposite cycles cancels out with respect to r.

1c. (12 points) From among the (reasonable) voting rules discussed in class, give 5 voting rules that satisfy the OCCO criterion, and 1 that does not.

1d. (14 points) Criterion C_1 is *stronger* than criterion C_2 if every rule that satisfies C_1 also satisfies C_2. Two criteria are *incomparable* if neither is stronger than the other. For every pair of criteria among OCO, CCO, and OCCO, say which one is stronger (or that they are incomparable).

2. Risk attitudes. Bob is making plans for Spring Break. He most prefers to go to Cancun, a trip that would cost him \$3000. Another good option is to go to Miami, which would cost him only \$1000. Bob is really excited about Spring Break and cares about nothing else in the world right now. As a result, Bob's utility u as a function of his budget b is given by:

- $u(b) = 0$ for $b < \$1000$;
- $u(b) = 1$ for $\$1000 \leq b < \$3000$;
- $u(b) = 2$ for $b \geq \$3000$.

Bob's budget right now is \$1100 (which would give him a utility of 1, for going to Miami).

Bob's wealthy friend Alice is aware of Bob's predicament and wants to offer him a "fair gamble." Define a *fair gamble* to be a random variable with expected

[3] E.g., not dictatorial rules, rules for which there is a candidate that can't possibly win, randomized rules, etc. Also, approval cannot be one of the rules because it is not based on rankings. If you use Cup, Cup only satisfies a criterion if it satisfies it for every way of pairing the candidates.

value \$0. An example fair gamble (with two outcomes) is the following: \$-75 with probability 2/5, and \$50 with probability 3/5. Note that the expected value of this gamble is \$0, so it is indeed fair. If Bob were to accept this gamble, he would end up with \$1025 with probability 2/5, and with \$1150 with probability 3/5. In either case, Bob's utility is still 1, so Bob's expected utility for accepting this gamble is $(2/5) \cdot (1) + (3/5) \cdot (1) = 1$.

a (7 points). Find a fair gamble with two outcomes that would strictly decrease Bob's expected utility.

b (7 points). Find a fair gamble with two outcomes that would strictly increase Bob's expected utility.

3. Finding Nash equilibria of normal-form games. (30 points.)

Find *all* the Nash equilibria of each of the following five two-player normal-form games. Argue why the games have no other Nash equilibria. (Hint: for some of these games, you may wish to use strict dominance or iterated strict dominance, because any strategy eliminated by (iterated) strict dominance cannot get positive probability in any Nash equilibrium. Also keep in mind that you may want to use strict dominance by a mixed strategy.)

4, 4	8, 2
2, 8	7, 7

0, 8	4, 0
2, 0	0, 1

7, 7	6, 8
9, 2	0, 1

3, 5	5, 4
1, 7	7, 6

4, 0	4, 0	1, 2
3, 5	3, 4	2, 4
4, 0	1, 1	5, 0

4. Extensive-form games. Consider the game in Figure 1.

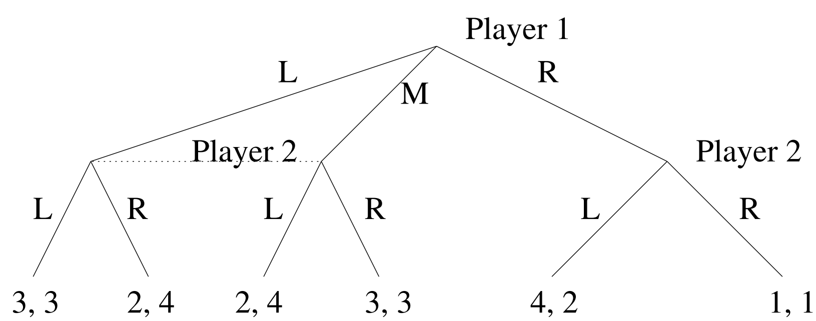

Figure 1: An extensive-form game with imperfect information.

a (7 points). Give the normal-form representation of this game.

b (7 points). Give a Nash equilibrium where player 1 sometimes plays Left. (Remember that you must specify each player's strategy at *every* information set.)

c (7 points). What are the subgame perfect equilibria of the game? (Remember that you must specify each player's strategy at *every* information set.)

5. Estimating utilities (35 points).

This question is a programming question. Please see Homework 1 for details about getting set up with GLPK, making a directory for this homework, etc.

Note: in this question, there is an example instance that you are asked to solve. However, just getting this example right is *not* enough to get full credit: your formulation should work on all instances. The example is just there to give you something to test your formulation on.

Elena Umberta Massima is an expected utility maximizer. When presented with two probability distributions over a set of possible outcomes, E.U.M. says, without hesitation, which she prefers, and you will not catch her in any inconsistencies.

We have four outcomes: A, B, C, D. We will accordingly represent probability distributions as vectors of four probabilities of the respective outcomes. $(p_A, p_B, p_C, p_D) \succ (p'_A, p'_B, p'_C, p'_D)$ will denote that E.U.M. prefers distribution p to p'. We learn the following four preferences:

- $(.1, .2, .3, .4) \succ (.1, .2, .4, .3)$
- $(.4, .4, .1, .1) \succ (.4, .2, .2, .2)$
- $(.6, .1, 0, .3) \succ (.4, .3, .3, 0)$
- $(.4, .3, .2, .1) \succ (.5, .5, 0, 0)$

Obviously, we jump on the opportunity to estimate the utilities (for outcomes A, B, C, D) of this fascinating woman.

Our goal will be to assign utilities in the interval $[0, 1]$ to the four outcomes that are consistent with E.U.M.'s preferences. Write a linear program formulation for this. You should add an objective to satisfy the consistency constraints by as large a margin as possible (similar to our linear program for strict dominance by mixed strategies). You should use the MathProg (`.mod`) language to model the general problem (you should allow for more than four outcomes and more than four preferences, because the second instance, the one you need to turn in, requires this).

First use this to solve the specific instance above, by completing the following code. (Hint: the optimal objective value for this example is 0.02.) Do **not** turn this one in; instead, turn it in with the **second** instance further below.

```
param n; # Number of outcomes
param m; # Number of preferences

# Define the probabilities for each preference (p and p')
param p{1..m, 1..n};
param p_prime{1..m, 1..n};

# Utilities of the outcomes, constrained to be between 0 and 1
var u{1..n} >= 0, <= 1;

# Slack variables to ensure each constraint is satisfied by at least margin delta
var delta >= 0;

# Define the objective to maximize delta
maximize Objective: delta;

# Constraints for each preference
s.t.    # YOUR TASK IS TO COMPLETE THIS

# Instance 1 for testing
data;

param n := 4;
param m := 4;

param p: 1 2 3 4 :=

    1 0.1 0.2 0.3 0.4

    2 0.4 0.4 0.1 0.1

    3 0.6 0.1 0.0 0.3

    4 0.4 0.3 0.2 0.1;

param p_prime: 1 2 3 4 :=

    1 0.1 0.2 0.4 0.3

    2 0.4 0.2 0.2 0.2

    3 0.4 0.3 0.3 0.0

    4 0.5 0.5 0.0 0.0;

end;
```

Now modify the data part as follows, corresponding to Instance 2. Turn it in with this.

```
# Instance 2 for submitting
data;

param n := 6;
param m := 8;

param p: 1 2 3 4 5 6 :=

    1 0.2 0.3 0.1 0.2 0.1 0.1

    2 0.5 0.2 0.1 0.1 0.05 0.05

    3 0.3 0.1 0.2 0.2 0.1 0.1

    4 0.4 0.2 0.1 0.1 0.1 0.1

    5 0.2 0.2 0.2 0.2 0.1 0.1

    6 0.35 0.15 0.15 0.15 0.1 0.1

    7 0.4 0.1 0.1 0.1 0.2 0.1

    8 0.3 0.25 0.15 0.1 0.1 0.1;

param p_prime: 1 2 3 4 5 6 :=

    1 0.1 0.2 0.3 0.2 0.1 0.1

    2 0.4 0.3 0.1 0.1 0.05 0.05

    3 0.2 0.2 0.1 0.3 0.1 0.1

    4 0.3 0.3 0.1 0.1 0.1 0.1

    5 0.3 0.1 0.3 0.2 0.05 0.05

    6 0.25 0.2 0.2 0.15 0.1 0.1

    7 0.3 0.2 0.2 0.1 0.1 0.1

    8 0.35 0.15 0.15 0.1 0.1 0.15;

end;
```