Math 3272: Linear Programming[1] Mikhail Lavrov

Lecture 19: Solving zero-sum games

October 20, 2022 *Kennesaw State University*

1 The "best worst-case" (maximin) strategy

1.1 Alice's plan

Let us summarize the previous lecture. In a matrix game, two players named Alice and Bob simultaneously pick one of several choices; we abstractly say that Alice's options are $\{1, 2, \ldots, m\}$ and Bob's options are $\{1, 2, \ldots, n\}$. There is an $m \times n$ matrix A listing Alice's payoffs: when Alice picks option $i \in \{1, 2, \ldots, m\}$ and Bob picks option $j \in \{1, 2, \ldots, n\}$, Alice receives a payoff of a_{ij} (which can be positive or negative). Alice would like to maximize her payoff.

(There is also an $m \times n$ matrix B for Bob's payoffs, and in a zero-sum game, we have $B = -A$. For now, we will ignore B, because we are only focusing on Alice's choices.)

A mixed strategy for Alice, corresponding to choosing between her strategies at random with some probabilities, is given by a vector $\mathbf{y} \in \mathbb{R}^m$ with $y_1 + y_2 + \cdots + y_m = 1$ and $\mathbf{y} \geq \mathbf{0}$. A mixed strategy for Bob is given by a vector $\mathbf{x} \in \mathbb{R}^n$ with $x_1 + x_2 + \cdots + x_n = 1$ and $\mathbf{x} \geq \mathbf{0}$. We saw that if Alice and Bob play these mixed strategies against each other, the expected payoff for Alice is given by $\mathbf{y}^\mathsf{T} A \mathbf{x}$.

Alice would like to choose the best probability vector $\mathbf{y}$ for her mixed strategy. But she can't use the formula $\mathbf{y}^\mathsf{T} A \mathbf{x}$ to evaluate how good a strategy is directly, because she doesn't know which vector $\mathbf{x} \in \mathbb{R}^n$ represents Bob's strategy. Instead, one thing Alice can do is evaluate her strategies by what happens if Bob *knows* her strategy and chooses the option that's worst for Alice. This is called Alice's **maximin strategy**.

For all we know, this could be a terrible idea! In fact, we can cook up lots of examples of games that *aren't* zero-sum, in which the maximin strategy does terribly.

Consider the "Win, Lose, and Copy" game, defined as follows. Bob has two options: "Win \$100" and "Lose \$100". Alice also has two options: "Don't play" and "Copy Bob". This game has payoff matrix

	Bob: Win \$100	Bob: Lose \$100
Alice: Don't play	$(0, 100)$	$(0, -100)$
Alice: Copy Bob	$(100, 100)$	$(-100, -100)$

In this game, Alice's maximin strategy is not to play: copying Bob has the risk that Bob will pick "Lose \$100", and not playing can't lose any money. But Bob isn't stupid and will never pick "Lose \$100", so "Copy Bob" is guaranteed to earn Alice \$100 as well.

[1] This document comes from the Math 3272 course webpage: https://facultyweb.kennesaw.edu/mlavrov/courses/3272-fall-2022.php

We will see that in the case of zero-sum games, the maximin strategy is reasonable, but it will take us some time to get there.

1.2 Writing down a linear program

First, here is a claim to simplify the analysis.

Theorem 1. *For any strategy played by Alice, at least one worst-case response from Bob is a pure strategy.*

Proof. If Alice is playing the mixed strategy given by some probability vector $\mathbf{y} \in \mathbb{R}^m$, then $\mathbf{y}^\mathsf{T} A$ is the vector of her possible payoffs, depending on Bob's choices. If Bob plays the pure strategy "always pick option i" for some $i \in \{1, 2, \ldots, n\}$, then Alice's payoff is going to be $(\mathbf{y}^\mathsf{T} A)_i$: the i^{th} component of this vector.

If Bob plays a mixed strategy $\mathbf{x} \in \mathbb{R}^n$, then Alice's expected payoff $\mathbf{y}^\mathsf{T} A \mathbf{x}$ is a weighted average of the payoffs above, where payoff $(\mathbf{y}^\mathsf{T} A)_i$ is multiplied by weight x_i. The weighted average can't be lower than the smallest of the payoffs $(\mathbf{y}^\mathsf{T} A)_1, (\mathbf{y}^\mathsf{T} A)_2, \ldots, (\mathbf{y}^\mathsf{T} A)_n$. So the smallest of those payoffs is Alice's worst case.

Or in other words: if option $j \in \{1, 2, \ldots, n\}$ is the best response for Bob, then playing a mixed strategy $\mathbf{x}$ instead could be described as "with probability x_j, do the best thing; the rest of the time, do something worse." That's obviously suboptimal.

(Technically, multiple options could be tied for Bob's best response, in which case choosing randomly between them is just as good as choosing one of them; but choosing randomly will never be strictly better.) □

Based on this claim, the worst-case payoff when Alice plays a mixed strategy given by $\mathbf{y} \in \mathbb{R}^m$ is

$$\min \left\{ (\mathbf{y}^\mathsf{T} A)_1, (\mathbf{y}^\mathsf{T} A)_2, \ldots, (\mathbf{y}^\mathsf{T} A)_n \right\}.$$

Therefore Alice can find a maximin strategy by solving the following optimization problem:

$$\begin{aligned} \underset{\mathbf{y} \in \mathbb{R}^m}{\text{maximize}} \quad & \min \left\{ (\mathbf{y}^\mathsf{T} A)_1, (\mathbf{y}^\mathsf{T} A)_2, \ldots, (\mathbf{y}^\mathsf{T} A)_n \right\} \\ \text{subject to} \quad & y_1 + y_2 + \cdots + y_m = 1 \\ & \mathbf{y} \geq \mathbf{0} \end{aligned}$$

This is not a linear program. But there is a trick that turns it into one!

To maximize the minimum of multiple options, we can maximize an auxiliary variable u, subject to the constraint that u is smaller than each option. In other words, we maximize u, adding the constraints $u \leq (\mathbf{y}^\mathsf{T} A)_1$, $u \leq (\mathbf{y}^\mathsf{T} A)_2$, …, $u \leq (\mathbf{y}^\mathsf{T} A)_n$.

Let $\mathbf{1}$ denote the vector $(1, 1, \ldots, 1)$ in which every component is 1. (In this case, we'll want to have $\mathbf{1} \in \mathbb{R}^n$, but in general, we'll abuse notation and write $\mathbf{1}$ for the all-ones vector of whichever dimension we need.) Then a quick way to write down these constraints on u is $u\mathbf{1}^\mathsf{T} \leq \mathbf{y}^\mathsf{T} A$. Similarly, the constraint $y_1 + y_2 + \cdots + y_m = 1$ can be written as $\mathbf{y}^\mathsf{T} \mathbf{1} = 1$, where $\mathbf{1} \in \mathbb{R}^m$.

So we get the following linear program:

$$\begin{array}{ll} \underset{\mathbf{y}\in\mathbb{R}^m, u\in\mathbb{R}}{\text{maximize}} & u \\ \text{subject to} & \mathbf{y}^\mathsf{T} A \leq u\mathbf{1}^\mathsf{T} \\ & \mathbf{y}^\mathsf{T}\mathbf{1} = 1 \\ & \mathbf{y} \geq \mathbf{0} \end{array}$$

1.3 Example: the odd-even game

As an example, let's consider the odd-even game from the previous lecture, with the payoff matrix below:

	Bob: 1	Bob: 2
Alice: 1	$(-1, 1)$	$(1, -1)$
Alice: 2	$(2, -2)$	$(-2, 2)$

If Alice plays a mixed strategy given by the probability vector (y_1, y_2), what happens?

- When Bob counters with playing "1" (holding up 1 finger), Alice's expected payoff is $y_1(-1) + y_2(2) = -y_1 + 2y_2$.
- When Bob counters with playing "2" (holding up 2 fingers), Alice's expected payoff is $y_1(1) + y_2(-2) = y_1 - 2y_2$.

Alice wants to choose (y_1, y_2) to maximize her expected payoff in the worst case: she wants to maximize $\min\{-y_1 + 2y_2, y_1 - 2y_2\}$. For this, we write down the linear program

$$\begin{array}{ll} \underset{u, y_1, y_2\in\mathbb{R}}{\text{maximize}} & u \\ \text{subject to} & -y_1 + 2y_2 \geq u \\ & y_1 - 2y_2 \geq u \\ & y_1 + y_2 = 1 \\ & y_1, y_2 \geq 0 \end{array}$$

2 Zero-sum games and duality

Of course, Bob can write down a similar linear program to Alice's. In the case of a general matrix game, Bob's payoffs are given by the $m \times n$ matrix B unrelated to A. When Bob plays a mixed strategy given by a probability vector $\mathbf{x} \in \mathbb{R}^n$, and Alice counters it by doing whatever is worst for Bob, Bob's expected payoff is

$$\min\left\{(B\mathbf{x})_1, (B\mathbf{x})_2, \ldots, (B\mathbf{x})_m\right\}.$$

There is not much else to say in the general case, but in the case of zero-sum games, the matrix B is equal to $-A$. In that case, the formula above can be rewritten as

$$\min\left\{(-A\mathbf{x})_1, (-A\mathbf{x})_2, \ldots, (-A\mathbf{x})_m\right\} = -\max\left\{(A\mathbf{x})_1, (A\mathbf{x})_2, \ldots, (A\mathbf{x})_m\right\}.$$

For Bob, trying to maximize $-\max\left\{(A\mathbf{x})_1, (A\mathbf{x})_2, \ldots, (A\mathbf{x})_m\right\}$ is equivalent to trying to minimize the same expression *without* the negative sign. In other words, it is equivalent for Bob to pick the

mixed strategy so that the highest payoff Alice can ensure for herself is as low as possible. This is called a **minimax strategy**.

(In a general matrix game, playing the minimax strategy is needlessly spiteful: you're not trying to do what's best for yourself, but instead trying to hurt your opponent as much as possible. In a zero-sum game, these are one and the same.)

We can use the same trick as earlier to rewrite this as a linear program: to minimize

$$\max\{(A\mathbf{x})_1, (A\mathbf{x})_2, \ldots, (A\mathbf{x})_m\},$$

Bob can minimize a quantity $v \in \mathbb{R}$ with the constraints that $v \geq (A\mathbf{x})_1$, $v \geq (A\mathbf{x})_2$, $\ldots$, $v \geq (A\mathbf{x})_m$. This results in the following linear program for Bob's minimax strategy:

$$\begin{array}{ll} \underset{\mathbf{x}\in\mathbb{R}^n, v\in\mathbb{R}}{\text{minimize}} & v \\ \text{subject to} & A\mathbf{x} \leq \mathbf{1}v \\ & \mathbf{1}^\mathsf{T}\mathbf{x} = 1 \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

Once again, let's look at this linear program in the case of the odd-even game. Here, if Bob plays a mixed strategy given by the probability vector (x_1, x_2), Alice can do the following:

- Counter by playing "1" (holding up 1 finger), obtaining an expected payoff of $-x_1 + x_2$.
- Counter by playing "2" (holding up 2 fingers), obtaining an expected payoff of $2x_1 - 2x_2$.

Bob wants to choose (x_1, x_2) to minimize Alice's expected payoff in her best case: he wants to minimize $\max\{-x_1 + x_2, 2x_1 - 2x_2\}$. For this, we write down the linear program

$$\begin{array}{ll} \underset{v,x_1,x_2\in\mathbb{R}}{\text{minimize}} & v \\ \text{subject to} & -x_1 + x_2 \leq v \\ & 2x_1 - 2x_2 \leq v \\ & x_1 + x_2 = 1 \\ & x_1, x_2 \geq 0 \end{array}$$

The amazing thing that happens is that Alice and Bob's linear programs are duals of each other! We can see this by rewriting them side-by-side in a more standardized form, and pairing variables and constraints appropriately:

$$\textbf{(P)} \begin{cases} \underset{u,y_1,y_2\in\mathbb{R}}{\text{maximize}} & u & \\ \text{subject to} & y_1 + y_2 = 1 & (v) \\ & u + y_1 - 2y_2 \leq 0 & (x_1) \\ & u - y_1 + 2y_2 \leq 0 & (x_2) \\ & y_1, y_2 \geq 0 & \end{cases} \qquad \textbf{(D)} \begin{cases} \underset{v,x_1,x_2\in\mathbb{R}}{\text{minimize}} & v & \\ \text{subject to} & x_1 + x_2 = 1 & (u) \\ & v + x_1 - x_2 \geq 0 & (y_1) \\ & v - 2x_1 + 2x_2 \geq 0 & (y_2) \\ & x_1, x_2 \geq 0 & \end{cases}$$

In particular, pay attention to u and v: these are unconstrained variables, and each one is paired with an equation constraint in the other linear program.

It is also true that in general, the linear program for Alice's maximin strategy is dual to the linear program for Bob's minimax strategy. We can write the linear programs in the following form to expose the duality:

$$\textbf{(P)}\begin{cases}\underset{u\in\mathbb{R},\mathbf{y}\in\mathbb{R}^m}{\text{maximize}} & u & & \\ \text{subject to} & & \mathbf{y}^\mathsf{T}\mathbf{1} = 1 & (v)\\ & & u\mathbf{1}^\mathsf{T} - \mathbf{y}^\mathsf{T}A \le \mathbf{0}^\mathsf{T} & (\mathbf{x})\\ & & \mathbf{y} \ge \mathbf{0} & \end{cases} \qquad \textbf{(D)}\begin{cases}\underset{v\in\mathbb{R},\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} & v & & \\ \text{subject to} & & \mathbf{1}^\mathsf{T}\mathbf{x} = 1 & (u)\\ & & \mathbf{1}v - A\mathbf{x} \ge \mathbf{0} & (\mathbf{y})\\ & & \mathbf{x} \ge \mathbf{0} & \end{cases}$$

The matrix of coefficients in the constraints is, in both cases, the matrix with block structure

$$\begin{bmatrix} 0 & \mathbf{1}^\mathsf{T} \\ \mathbf{1} & -A \end{bmatrix}$$

where A is Alice's payoff matrix.

3 Solving the linear program

Let's return to Alice and Bob's linear programs for the odd-even game:

$$\textbf{(P)}\begin{cases}\underset{u,y_1,y_2\in\mathbb{R}}{\text{maximize}} & u & & \\ \text{subject to} & & y_1 + y_2 = 1 & (v)\\ & & u + y_1 - 2y_2 \le 0 & (x_1)\\ & & u - y_1 + 2y_2 \le 0 & (x_2)\\ & & y_1, y_2 \ge 0 & \end{cases} \qquad \textbf{(D)}\begin{cases}\underset{v,x_1,x_2\in\mathbb{R}}{\text{minimize}} & v & & \\ \text{subject to} & & x_1 + x_2 = 1 & (u)\\ & & v + x_1 - x_2 \ge 0 & (y_1)\\ & & v - 2x_1 + 2x_2 \ge 0 & (y_2)\\ & & x_1, x_2 \ge 0 & \end{cases}$$

Can we figure out their optimal strategies?

We can solve these linear programs directly, but they have a couple of unsavory features: each has an equational constraint and an unrestricted variable. These are both things we can deal with, but in this case, there are shortcuts to take that make our life easier.

3.1 Complementary slackness

In cases where one player has only two options to choose from, we can solve the other player's linear program quickly by using complementary slackness. In this case, this works for both Alice and Bob; let's use it to solve Alice's linear program.

The logic is this: looking at the payoff matrix, we can quickly check that Bob does not have a dominant strategy. Therefore Bob's optimal strategy should be a mixed strategy with $x_1 > 0$ and $x_2 > 0$.

By complementary slackness, this means that in Alice's linear program, $u + y_1 - 2y_2 = 0$ and $u - y_1 + 2y_2 = 0$. From this, we can deduce that $-y_1 + 2y_2 = y_1 - 2y_2$, because both are equal to u. (Intuitively, this means Alice wants to pick (y_1, y_2) so that Bob is indifferent between his options.) This simplifies to $y_1 = 2y_2$, and from $y_1 + y_2 = 1$, we conclude that Alice's optimal strategy is given by $(y_1, y_2) = (\frac{1}{3}, \frac{2}{3})$: she should hold up one finger $\frac{1}{3}$ of the time and two fingers $\frac{2}{3}$ of the time.

This reasoning does not *always* work in bigger problems. If Bob has many options, even if there is no option that dominates another by itself, it's possible that one of Bob's options is dominated by a mixed strategy formed by several others. So we can't necessarily say that Alice wants to make Bob indifferent between *all* of his options: only the ones that make sense for Bob to sometimes play!

3.2 Simplifying the linear program

There are a couple of things we can do to make the linear program easier to solve; they can be done independently or together.

First, it would be nice if we could make u a nonnegative variable in Alice's linear program, because then we can use it in the simplex method. There is a trick to do this: modify the game so that every time Alice and Bob play, Bob gives Alice an extra \$2 for free. The new payoff matrix becomes:

	Bob: 1	Bob: 2
Alice: 1	$(1, -1)$	$(3, -3)$
Alice: 2	$(4, -4)$	$(0, 0)$

In this payoff matrix, Alice's payoffs are always nonnegative, so u will also be nonnegative. However, Alice's optimal strategy is unaffected by Bob giving her \$2 unconditionally. Therefore solving the linear program with this new payoff matrix will produce the same optimal (y_1, y_2), and now we can assume that $u \geq 0$.

Also, though in general an equational constraint is hard to deal with, in this problem we don't have to go to the trouble of using a two-phase method. Any pure strategy, such as for example $(y_1, y_2, \ldots, y_m) = (1, 0, \ldots, 0)$, is a feasible solution to Alice's linear program. This means that after adding slack variables $w_1, w_2, \ldots, w_n$ to Alice's linear program, we can solve for the basic variables $(y_1, w_1, w_2, \ldots, w_n)$ to get an initial feasible basis.