Two Player Zero-Sum Games

CS 149 Staff

December 5, 2006

1 Introduction

No polynomial time algorithm is known for finding equilibria in general games, but an important special case, two player zero-sum games, can be solved in polynomial time via linear programming. Zero sum means that one of player's gain is the other's loss; there are no win-win or lose-lose outcomes. Rock paper scissors is perhaps the simplest game of this type. The purest form of game theory concentrates on games where there is only one move and both players move simultaneously.[1] Two player zero-sum games are sometimes referred to as matrix games.

Conventions: the *row player* chooses a row of the matrix, the *column player* chooses a column. The entries of the matrix are amounts of money that the row player gives to the column player, so the column player tries to maximize the selected matrix entry and the row player tries to minimize it. The slides number the column and row players 1 and 2 respectively.

In many games it is risky to fix your strategy in advance because if the opponent successfully reads your mind they can choose a strategy that works well against yours. For example, if you always play rock in rock paper scissors, your opponent would probably respond by always playing paper. The foolproof way to prevent this from happening is to choose your move randomly. Assume that the players choose their strategies independently.

The way the players used to randomly choose their moves can be characterized by probility distributions. If the probilitity distribution has one component equal to 1 and the rest zero (always choose a particular row), it is called a *pure strategy*, otherwise it is a *mixed strategy*. A probability distribution can be characterized by a vector with non-negative components that sum to 1. Call the column player's strategy x and the row player's π.

The expected amount of money that the row player gives to the column player can be readily seen to be $\pi^T Ax$. The column player tries to maximize $\pi^T Ax$, whereas the row player tries to minimize it.

[1] In theory you can model games with many turns as a game with one turn by having the players write down at the beginning of the game what they will do in any possible eventuality. Most games have exponential state spaces so doing this in practice would be unbelievably tedious, but it's a convenient trick to analyze such games. These techniques have been successfully used to approximately solve Texas hold'em poker.

2 LP Formulation

How do a mathematically model to players simultaneously optimizing something in opposite directions? One way is to suppose that the column player first writes down his probability distribution, shows it to the row player, and then asks the row player to make his choice (or suppose the row player has spies that will tell him what probability distribution the column player chooses his play from). To choose the best strategy the column player needs to solve $\max_x \min_\pi \pi^T Ax$. This may seem like this is treating the two players differently, but it turns out that this will find strategies for the players such that even if both players learn the other's strategy neither will want to change their own. This is called a Nash equilibrium. It turns out that Nash equilibrium always exist, even in general games with more players and non-zero sum.[2]

The expression $\max_x \min_\pi \pi^T Ax$ has many of the same symbols as linear programming uses but it has two problems: it's not linear and it has the nested max and min. With a little trickery we can simplify it and express it as linear program. Consider hypothetically evaluating $\max_x \min_\pi \pi^T Ax$ by exhaustive search: for every one of the (infinitely many) possible values for x, compute $f(x)$ and choose the best one, where $f(x) = \min_\pi \pi^T Ax$. This gives an idea that perhaps we should analyze $\min_\pi \pi^T Ax$ for a fixed x.

For a fixed x, Ax is just a vector. Since π is a probibility distribution, $\pi^T Ax$ is a convex combination of components of Ax, so from linear programming we know that the maximum is acheived at an extreme point of the probibility distribution polytope. The extreme points are precisely the pure strategies. Therefore, $\min_\pi \pi^T Ax = \min_i (Ax)_i$, where $(Ax)_i$ denotes the ith component of Ax. Therefore our simplified objective function is $\max_x \min_i (Ax)_i$. To improve this further, define $z = \min_i (Ax)_i$, so our problem is $\max z$ subject to π being a probability distribution and $z = \min_i (Ax)_i$. Now if we replace that constraint with $z \le \min_i (Ax)_i$, that doesn't change optimal solutions since any feasible solution where that inequality isn't tight can be improved by increasing z and leaving π unchanged. Now $z \le \min_i (Ax)_i$ is equivilent to $z \le (Ax)_i$ for all i, yielding the linear program $\max z$ subject to $x \ge 0$, $1^T x = 1$ and for all i $z \le (Ax)_i$.

This linear program shows that two player zero-sum games can in fact be solved in polynomial time. The dual of this linear program has a dual variable for each constraint, most of which correspond to strategies for the row player. One might hope that these correspond to an optimal strategy for the row player but the normalization constraint gets in the way. There is, however, a trick that gets around this and makes the dual linear program correspond to the row player's minimization problem. Suppose we somehow knew that the optimal value of the game was positive. The key is a change of variables from x_i and z to $y_i = x_i/z$ and noting that $\sum y_i = z$ so $\sum y_i$ can be substituted for z in the objective, eliminating that constraint and the z variable. Finally, replace the maximization of a quantity by the minimization of its reciprocal. The resulting LP is $\min 1^T y$ subject to $y \ge 0$ and $1 \le (Ay)$. One can show that the dual of this LP is the row player's problem.

[2] The proof is via the Kakutani fixed point theorem.

One thing to remember is to multiply the y_is by z to get the strategy probabilities x_i. If you forget to do this you may scratch your head wondering why the y_is don't add up to 1.

3 Aside: game theory

General game theory (beyond zero-sum) has many interesting paradoxes. For example, in the prisoner's dilemma, it is rational to backstab your partner even though you'd be a lot better off if both of you cooperated. If you agree to play that game hundred times it is intuitive that you should try cooperating and only switch to backstabbing in revenge for your partner doing the same against you. However, it is obviously rational to backstab your partner at the last game, and by induction you can see that you should therefore backstab your partner at the second to last game and so on. So why is it that if you run a competition of this game the winning strategy is usually just do this round what your opponent did last round? The answer is that while always backstabbing is the only pure Nash equilibrium to the iterated game, there are mixed equilibria which involve cooperation.

A similar paradox involves the pop-quiz problem: a teacher announces that there will be a pop-quiz sometime next week, and it wouldn't be a pop quiz if given Friday, etc. This is resolved by rather than insisting the quiz is always a surprise, model the preparation/quizing game as a game with a payoff whenever the student studies the night before the quiz is given and a penalty whenever the student studies needlessly. I think that in equilibrium the quiz would occur more often earlier in the week, but it would occasionally occur on Friday.