The Game Theory of Mutually Assured Destruction

Pranay Gundam

June 23, 2024

Introduction

The game theory we usually talk about in introductory/intermediate undergrad classes and high-school always left me really dissatisfied (which to be fair, it is my fault that this is the extent of classes that I have taken that cover game theory). I know there are textbooks that cover models and games that are a quintessential part of the literature but I am going through these modeling exercises myself just to practice creativity. Specifically I want to talk about the idea of Mutually Assured Destruction (MAD) that became so popular during the Cold War.

The Basic 2x2

In highschool, AP Econ classes we are taught about simple games where there are two players each of whom can take one of two possible actions. In the context of MAD, we could label the two agents as the "US" and the "Soviet Union", both of whom can decide to "Launch nukes" or "Don't launch nukes". The corresponding chart we would use to work this problem out would look something like

	Russia Launch	**Russia Don't Launch**
US Launch	-100, -100	0, -99
US Don't Launch	-99, 0	0, 0

I made the chart above with a set of payoffs for each combination of actions such that there is only one Nash equilibrium at both agents choosing "Don't Launch". The equilibria of this game could have changed if I had chosen different payoffs and one interesting concept to explore is the conditions on the payoffs in order for each outcome in the state space to become a Nash equilibrium. Specifically, consider the chart below, here the x payoffs belong to the US and the y payoffs belong to the Soviet Union. The $x_{DL,DL}$ payoff for example is the payoff that the US experiences when both the US and the Soviet Union chooses not to launch.

	SU Launch	**SU Don't Launch**
US Launch	$x_{L,L}, y_{L,L}$	$x_{L,DL}, y_{L,DL}$
US Don't Launch	$x_{DL,L}, y_{DL,L}$	$x_{DL,DL}, y_{DL,DL}$

Visualizing the domains at which these conditions hold is a bit difficult to do all at once since we have to higher dimensional spaces but we can at least talk about it.

We can also simplify the problem of visualizing when certain outcomes become Nash equilibria by imposing some functional constraints on the payoffs. For example, we can say that a country has a utilitarian mindset applying to people of all nationalities (where all life has at least some positive value) and when getting launched at it is weakly more utility for them to not retaliate. In such a case, if the US were to adopt this mindset, we would then have that $x_{DL,L} \geq x_{L,L}$. This combined with the not so wild assumption that the US would prefer not to get launched at would yield the relationship $x_{DL,DL} \geq x_{DL,L} \geq x_{L,L}$. We can tack on one final assumption, that the US attributes more value to their own citizens than citizens of other countries, and we can completely describe the relationship between all the payoffs that the US experiences

$$x_{DL,DL} \geq x_{L,DL} \geq x_{DL,L} \geq x_{L,L}.$$

What is further interesting to discuss is how countries/agents with different functional paradigms of determining their payoffs fare in contest with each other.

N Discrete Choices and M players

The world is of course not a simple place where there are only two options. In the case of the Cold War, for example, the US and the Soviet Union fought in many ways other than just launching nuclear weapons such as in proxy wars or defensive posturing of their allies and weapons. With discrete choices, as long as the choices are finite, we can always still make a table and do the same exercises as we would with a 2x2 table to analyze equilibria. If we are to add multiple players, however, the dimensionality of the charts would have to increase which is a bit difficult to write down (on a 2 dimensional page).

Let's set up some notation; let there be M many players $\{p_1, p_2, \ldots, p_M\}$, where each agent m has N_m actions (note N_m may be different for each agent) $\{a_{m,1}, a_{m,2}, \ldots, a_{m,N_m}\}$. So finally, let $\{i_1, i_2, \ldots, i_M\}$ be indices such that for any $m \in [M]$ we have that $i_m \in [N_m]$. This lets us write, for a vector of any combinations of actions $\mathbf{a} = (a_{1,i_1}, a_{2,i_2}, \ldots, a_{M,i_M})$ we can denote the payoff vector as $(p_1(\mathbf{a}), p_2(\mathbf{a}), \ldots, p_M(\mathbf{a}))$. There's a lot of indexing that's going on but I'm trying to make sure I'm being rigorous; regardless the main idea to takeaway from the notation is that each agent has their own set of actions that they can take and each agent's payoff is a separate function whose domain is the set of all combination of actions each agent can take. With this construction, we can already see how in the continuous space we can work with objects such as jacobians to do the analysis that we want to do.

In the discrete space, however, we can make claims about nash equilibria at a action vector $\mathbf{a}^*$

when for each agent $m \in [M]$ we have that for any other $\mathbf{a}'$ in the action space, where a' is an action vector such that all the other players' are fixed, that

$$p_m(\mathbf{a}^*) \geq p_m(\mathbf{a}').$$

To offer some context, consider the same Cold War scenario except this time there are three actors: US, Soviet Union, and China. Each country can choose one of three actions, to launch a nuclear missile at one of the other two countries or not. Each country places some heterogeneous political value in having one of their rival countries destroyed, for example the US sees the Soviet Union as more of a threat and so is more keen on launching against the Soviet Union than China, which creates rich dynamics in the gridlock between the three countries. If one were to launch against another then that country has lost their deterrent and the final country could launch to become the sole survivor. You can see how this can become a lot more complicated than the 2 by 2 setup we discussed before.

Continuous Choices

The setup here is quite analogous to the generalized discrete case above except we work with a continuous measure of actions. Again let there be M many players $\{p_1, p_2, \ldots, p_M\}$, where each agent m can take an action a_m which we say is indexed by $a_m \in [0, 1]$. We can denote the payoff vector as $(p_1(\mathbf{a}), p_2(\mathbf{a}), \ldots, p_M(\mathbf{a}))$.
In my mind we can find nash equilibria in this setup by identifying points where the jacobian matrix of the system of payoffs at that point turns into a zero matrix and the hessian is negative (which tends to be the case anyways given we are using "well-behaved" payoff functions for each of the agents). This to me seems like a nuke all condition (pun intended) for claiming equilibrium but I think is all we can claim in such a generalized setup. From my experience in Macro so far, calculating objects such as the jacobian can be difficult in certain setups: such as instances where actions are temporally sequentially related (as is the case in most Macro models).

This brings us to another discussion about the model we have crafted thus far, what exactly the action space involves and how to structure problems. The most generalized setup that I discussed above can account for a problem where actors make actions over different points in time by writing a large action space that accounts for each possible sequence of actions. This of course is quite the brute force method that creates an incredibly large state space to work with. In macro, we try to reduce this problem by writing a dynamic recursively defined problem with the help of bellman equations but this also relies on an assumption that optimal actions only depend on the current state and don't have a temporal dependence.

Concluding Thoughts

As a first aside, the more I think about this, the more I realize there are many parallels all the difference fields in econ (to be quite reductionist everything just seems like various ways to form optimization problems and how we can take advantage of the structural relationships that we impose to offer causal insights or take shortcuts to solve the problem).

The generalized setup does make me excited for the types of problems and relationships that can arise from it. The big issue right now is that I've done a lot of boring notation and setup but haven't crafted a problem that we could make interesting insights from. I'll save that for the second semester Micro of grad school.