Simpler version of the problem in Gehlbach section 2.1.1

Andy Eggers

HT 2018

Gehlbach opens Chapter 2 with a challenging problem. To make it more digestible, let's look at a simpler version.

How is this different from the problem in Gehlbach 2.1.1?

First, we'll assume just two groups, group 1 and group 2. This makes the problem easier to solve (notably, you don't need to solve a Lagrangian), but you can still see most of the key points from the model.

Second, we will assume that $\omega_1 = \omega_2 = 1$. This parameter measures something like "ideological polarization" or "attention to this policy area" in the two groups – for higher ω, the vote share in the group responds more to changes in the parties' policy platforms.

I will also provide more detailed explanation of the model than Gehlbach does. I will be much more explicit about each step, provide some figures, and do some numerical checking and visualization.

The model

A society is organized into two groups, which we will call 1 and 2. The share of citizens in group 1 is α (so the share in group 2 is $1 - \alpha$). Citizens in group g have a pre-tax income of y_g.

Two parties, A and B, compete by offering tax-and-transfer platforms. Party A's platform can be denoted t_{1A}, t_{2A}, where t_{1A} is the net per-person transfer to members of group 1 proposed by party A and t_{2A} is the net per-person transfer to members of group 2. Party B's platform is similarly denoted t_{1B}, t_{2B}.

Tax-and-transfer platforms must maintain budget balance, i.e. $\alpha t_{1g} + (1 - \alpha)t_{2g} = 0$ for $g \in \{A, B\}$. This means that either $t_{1g} = t_{2g} = 0$, i.e. neither group gains or loses anything, or one group is taxed to pay the other. Also, a group can't be taxed beyond its income, e.g. t_{1A} can't be larger than y_1.

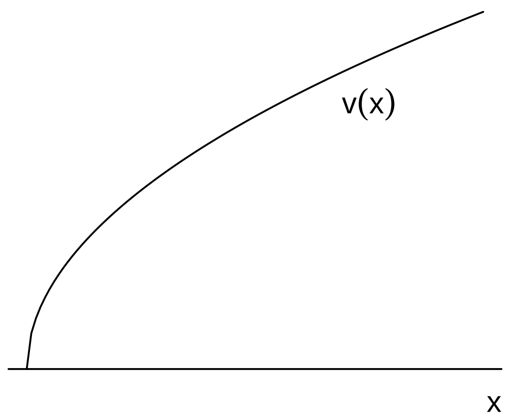
Parties seek to maximize their vote share.

Think at this point about what would happen if voters in the model cared only about the net transfer they received, e.g. if members of group 1 would all vote for party A if $t_{1A} > t_{1B}$, all vote for party B if $t_{1A} < t_{1B}$, and split their votes in half if $t_{1A} = t_{1B}$. (This is the assumption about voter behavior made in all of the models in Chapter 1.) The only Nash equilibrium of this game has both parties taxing the smaller group for all of its income to give the larger group the largest possible transfer, with support being split evenly between the two parties.

Instead, we will assume that voters have some other concern, such that if one party improves the net transfer it offers to members of a group, it will win some of the voters in that group but not all. Specifically, we assume that voter i in group g votes for party A if

$$v(y_g + t_{gA}) > v(y_g + t_{gB}) + \eta_i$$

where v is a monotonically increasing and concave function. The function v could look like this, for instance:



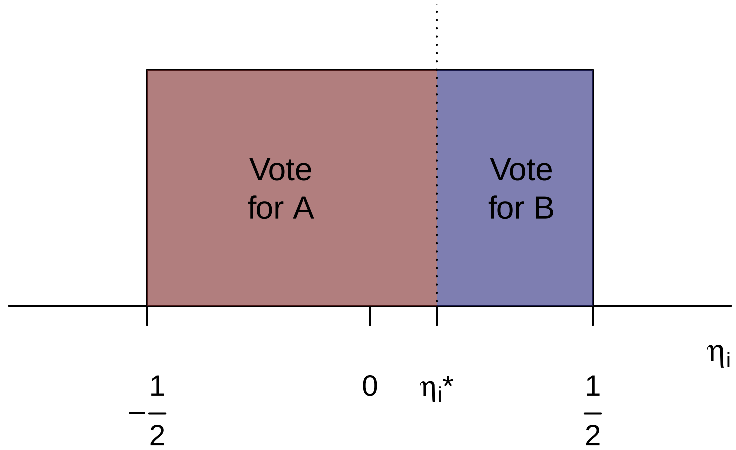
The variable η_i captures voter i's predisposition in favor (or against, if negative) party B. We assume that η_i is distributed uniformly in each group on the interval $\left[-\frac{1}{2}, \frac{1}{2}\right]$, and we will assume that neither group entirely supports one party.[1]

To start analyzing how the parties choose platforms to maximize their vote share, we must first state the parties' vote shares in terms of the choice variables (i.e. policy platforms, i.e. tax-and-transfer programs) and model parameters. Note first that we can rearrange the expression above to say that a voter in group g votes for party A if

$$\eta_i < v(y_g + t_{gA}) - v(y_g + t_{gB}).$$

Let $\eta_i^* = v(y_g + t_{gA}) - v(y_g + t_{gB})$ denote the value of η_i such that the voter is indifferent between the two parties. Then voting behavior in group g can be graphically described like this:

[1] Technically, this condition is $|v(y_g + t_{gA}) - v(y_g + t_{gB})| < 1/2$ for $g \in \{1, 2\}$.



Now, to convert this into party A's vote share in this group, we need to calculate the area of the red rectangle. The height of a uniform density that runs from $-1/2$ to $1/2$ must be one. (This is because the area under a density must be one.) Thus the area of the red rectangle (i.e. party A's vote share in this group) must be $\eta_i^* - \left(-\frac{1}{2}\right) = \frac{1}{2} + v(y_g + t_{gA}) - v(y_g + t_{gB})$. Combining the two groups, party A's overall vote share can be written

$$V_A = \frac{1}{2} + \alpha\big(v(y_1 + t_{1A}) - v(y_1 + t_{1B})\big) + (1-\alpha)\big(v(y_2 + t_{2A}) - v(y_2 + t_{2B})\big).$$

Party A's problem is to choose t_{1A}, t_{2A} to maximize this expression, subject to the constraint that $\alpha t_{1A} + (1-\alpha)t_{2A} = 0$.

If we had more groups, at this point we would use the Lagrangian technique used in Gehlbach's more complicated version of this problem. Instead we will do something simpler. From the budget constraint, we know that $t_{2A} = -\frac{\alpha}{1-\alpha} t_{1A}$. We can then substitute that into the expression for party A's vote share and choose t_{1A} to find a maximum. (This turns a "constrained optimization" problem into an "unconstrained optimization" problem.)

That is, we're going to solve this problem:

$$\max_{t_{1A}} V_A = \frac{1}{2} + \alpha\big(v(y_1 + t_{1A}) - v(y_1 + t_{1B})\big) + (1-\alpha)\left(v\left(y_2 - \frac{\alpha}{1-\alpha} t_{1A}\right) - v(y_2 + t_{2B})\right).$$

Before we do that, let's make it look less intimidating by taking out terms that don't depend on t_{1A}. We rewrite the maximization problem as

$$\max_{t_{1A}} V_A = \alpha v(y_1 + t_{1A}) + (1-\alpha) v\left(y_2 - \frac{\alpha}{1-\alpha} t_{1A}\right) + C.$$

Now we get the first-order condition:

$$\frac{\partial \tilde{V}_A}{\partial t_{1A}} = \alpha v'(y_1 + t_{1A}) - \alpha v'\left(y_2 - \frac{\alpha}{1-\alpha} t_{1A}\right) = 0$$

Then we rearrange and solve for t_{1A}:

$$\begin{aligned} v'(y_1 + t_{1A}) &= v'\left(y_2 - \frac{\alpha}{1-\alpha} t_{1A}\right) && (1) \\ y_1 + t_{1A} &= y_2 - \frac{\alpha}{1-\alpha} t_{1A} && (2) \\ t^*_{1A} &= (1-\alpha)(y_2 - y_1) && (3) \end{aligned}$$

The second line follows from the assumption that v is concave, meaning that the first derivative is montonically decreasing (and thus $v'(x) = v'(y) \iff x = y$).

Above we noted that $t_{2A} = -\frac{\alpha}{1-\alpha} t_{1A}$. We can now substitute t^*_{1A} into this expression, which yields $t^*_{2A} = \alpha(y_1 - y_2)$.

Note that although the vote share won by party A depends on what party B does, party A's optimal move does not. This is because increasing a group's net transfer by a given amount wins party A the same number of voters regardless of what party B is doing. (This is a benefit of assuming a uniform distribution of voters in each group – things would be even more complicated without this assumption.)

Because the game is symmetric (i.e. party A and party B are interchangeable), we know that the solution for party B is the same: $t^*_{1B}, t^*_{2B} = (1-\alpha)(y_2 - y_1), \alpha(y_1 - y_2)$.

Analysis of the solution

Our solution is that $t^*_{1A} = t^*_{1B} = (1-\alpha)(y_2 - y_1)$ and $t^*_{2A} = t^*_{2B} = \alpha(y_1 - y_2)$. Let's try to interpret this.

First, note that the net transfer to group 1 and group 2 must be of opposite signs: we must be either taking from 1 to give to 2 if $y_1 > y_2$, or the reverse if $y_2 > y_1$, or neither if $y_1 = y_2$. We knew from the budget constraint that we would be taking from one to give to the other, so it would be worrying if this were not in our solution.

Second, note that the solution involves taxing the richer group and giving the proceeds to the poorer group:

t_1^* is positive if and only if $y_2 > y_1$. So, this is redistribution driven purely by electoral incentives. Ultimately, this reflects the assumption that v is concave and the distribution of η_i is identical across groups, which ensures that voter utility (and thus vote share) is more responsive to a given transfer or tax in the poorer group.

Third, the magnitude of the transfer is inversely related to the relative size of the group: the larger group gets a smaller per-person tax or benefit. This must be true given the budget constraint.

Illustration of the solution

We can gain a little more insight into the problem (and check that our calculations above are correct) by assuming a specific function for v, plugging in numbers for α, y_1, and y_2, and checking if the solutions we calculated above really do maximize vote share. I will do this in `R`.

First, let's assume that v is the square root of its argument:

```
v.func = function(x){sqrt(x)}
```

Then we write a function for party A's vote share, mirroring the expression we worked out above for party A's total vote share:

$$V_A = \frac{1}{2} + \alpha\big(v(y_1 + t_{1A}) - v(y_1 + t_{1B})\big) + (1 - \alpha)\big(v(y_2 + t_{2A}) - v(y_2 + t_{2B})\big)$$

Here is the `R` code:

```
vote.share.1 = function(t_1A, t_2A, t_1B, t_2B, alpha, y_1, y_2) {
    1/2 + alpha * (v.func(y_1 + t_1A) - v.func(y_1 + t_1B)) + (1 - alpha) *
        (v.func(y_2 + t_2A) - v.func(y_2 + t_2B))
}
```

Next we store some assumed values for the exogenous variables:

```
alpha = .4
y_1 = 1
y_2 = 2
```

We will assume that party B is proposing the transfers implied by our solution above:

```r
(t_1B = (1 - alpha)*(y_2 - y_1))
```

```
## [1] 0.6
```

```r
(t_2B = alpha*(y_1 - y_2))
```

```
## [1] -0.4
```

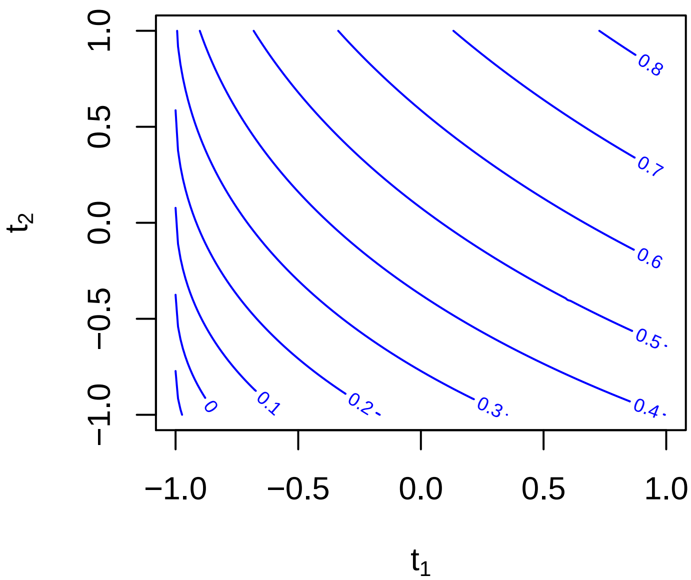
Next we calculate the vote share for a range of possible values of t_{1A} and t_{2A}:

```r
t_1s = seq(-1, 1, 0.01)  # the values of t_1A at which we will calculate vote share
t_2s = seq(-1, 1, 0.01)  # same for t_2A
z = matrix(NA, nrow = length(t_2s), ncol = length(t_1s))  # the matrix in which we
# will store the vote shares -- the 'z' matrix


# we loop over over possible combinations of t_1 and t_2:
for (i in 1:length(t_1s)) {
    for (j in 1:length(t_2s)) {
        # for each combo, we calculate vote share and store in the 'z' matrix
        z[i, j] = vote.share.1(t_1A = t_1s[i], t_2A = t_2s[j], t_1B = t_1B,
            t_2B = t_2B, alpha = alpha, y_1 = y_1, y_2 = y_2)
    }
}
```

Now we use the `contour()` command to plot vote share as a function of t_{1A} and t_{2A}:

```r
# now we plot it:
contour(t_1s, t_2s, z, col = "blue", xlab = expression(t[1]), ylab = expression(t[2]))
```

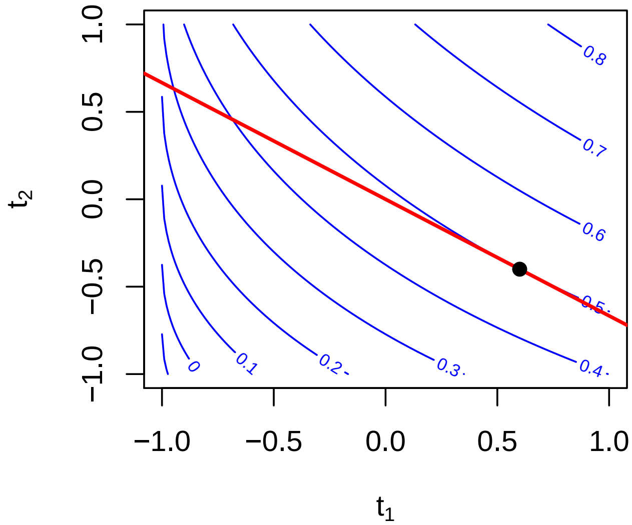


This looks nice, and it should make sense: the larger the net transfers t_{1A} and t_{2A}, the larger the vote share party A can expect. Of course, we know that not every combination of net transfers is feasible. In fact, the budget constraint imposes the condition that

$$t_{2A} = -\frac{\alpha}{1-\alpha} t_{1A}.$$

On our contour plot this is a line with intercept of 0 and slope of $-\frac{\alpha}{1-\alpha}$. So let's plot this line and show the point that we calculated as the optimum policy:

```
# plot the contour again
contour(t_1s, t_2s, z, col = "blue", xlab = expression(t[1]), ylab = expression(t[2]))
# now we add the "budget constraint"
abline(a = 0, b = -(alpha/(1-alpha)), col = "red", lwd = 2)
# and plot the solution that we calculated above
points(t_1B, t_2B, pch = 19)
```



The optimum policy is the point on the budget line that yields the highest vote share for party A. It is the point of tangency between the contour lines of the vote share function and the budget line.

This figure reassures me that my calculations are correct: it makes sense for the solution to be at the point of tangency indicated, and it makes sense that when both parties are playing the optimal strategy they split the vote evenly between them.

I can redo the exercise entering alternative parameters to confirm that it wasn't just a fluke that the figure gives me the same answer as my calculations. I can also confirm that changing the net transfers offered by party B changes the vote shares for A but does not change A's optimal action.

In practice, when I do exercises like this I very rarely get the right answer the first time. I either have an error in my calculations, an error in my code, or both. But when the two sets of analysis do agree, I am much more confident that I haven't made a mistake.