

R Analyze Constant Elasticity of Substitution (Atkinson Family Utility)

Fan Wang

2020-04-14

Contents

Atkinson Family Utility	1
-----------------------------------	---

Atkinson Family Utility

Go to the [RMD](#), [R](#), [PDF](#), or [HTML](#) version of this file. Go back to [fan's REconTools](#) Package, [R4Econ](#) Repository ([bookdown site](#)), or [Intro Stats with R](#) Repository.

Individual Outcomes and Preference How does the Atkinson Family utility function work? The Atkinson Family Utility has the following functional form.

$$V^{\text{social}} = (\alpha \cdot A^\lambda + \beta \cdot B^\lambda)^{\frac{1}{\lambda}}$$

Several key issues here:

1. V^{social} is the utility of some social planner
2. A and B are allocations for Alex and Ben.
3. α and β are biases that a social planner has for Alex and Ben: $\alpha + \beta = 1$, $\alpha > 0$, and $\beta > 0$
4. $-\infty < \lambda \leq 1$ is a measure of inequality aversion
 - $\lambda = 1$ is when the planner cares about weighted total allocations (efficient, Utilitarian)
 - $\lambda = -\infty$ is when the planner cares about only the minimum between A and B allocations (equality, Rawlsian)

What if only care about Alex? Clearly, if the planner only cares about Ben, $\beta = 1$, then:

$$V^{\text{social}} = (B^\lambda)^{\frac{1}{\lambda}} = B$$

Clearly, regardless of the value of λ , as B increases V increases. What Happens to V when A or B increases? What is the derivative of V with respect to A or B ?

$$\frac{\partial V}{\partial A} = \frac{1}{\lambda} (\alpha A^\lambda + \beta B^\lambda)^{\frac{1}{\lambda}-1} \cdot \lambda \alpha A^{\lambda-1}$$

$$\frac{\partial V}{\partial A} = (\alpha A^\lambda + \beta B^\lambda)^{\frac{1-\lambda}{\lambda}} \cdot \alpha A^{\lambda-1} > 0$$

Note that $\frac{\partial V}{\partial A} > 0$. When $\lambda < 0$, $Z^\lambda > 0$. For example $10^{-2} = \frac{1}{100}$. And For example $0.1^{-\frac{3}{2}} = \frac{1}{0.1^{1.5}}$. Still Positive.

While the overall V increases with increasing A , but if we did not have the outer power term, the situation is different. In particular, when $\lambda < 0$:

$$\text{if } \lambda < 0 \text{ then } \frac{d(\alpha A^\lambda + \beta B^\lambda)}{dA} = \alpha \lambda A^{\lambda-1} < 0$$

Without the outer $\frac{1}{\lambda}$ power, negative λ would lead to decreasing weighted sum. But:

$$\text{if } \lambda < 0 \text{ then } \frac{dG^{\frac{1}{\lambda}}}{dG} = \frac{1}{\lambda} \cdot G^{\frac{1-\lambda}{\lambda}} < 0$$

so when G is increasing and $\lambda < 0$, V would decrease. But when $G(A, B)$ is decreasing, as is the case with increasing A when $\lambda < 0$, V will actually increase. This confirms that $\frac{\partial V}{\partial A} > 0$ for $\lambda < 0$. The result is symmetric for $\lambda > 0$.

Indifference Curve Graph Given V^* , we can show the combinations of A and B points that provide the same utility. We want to be able to potentially draw multiple indifference curves at the same time. Note that indifference curves are defined by α, λ only. Each indifference curve is a set of A and B coordinates. So to generate multiple indifference curves means to generate many sets of A, B associated with different planner preferences, and then these could be graphed out.

```
# A as x-axis, need bounds on A
fl_A_min = 0.01
fl_A_max = 3
it_A_grid = 10000

# Define parameters
# ar_lambda <- 1 - (10^(c(seq(-2,2, length.out=3))))
ar_lambda <- c(1, 0.6, 0.06, -6)
ar_beta <- seq(0.25, 0.75, length.out = 3)
ar_beta <- c(0.3, 0.5, 0.7)
ar_v_star <- seq(1, 2, length.out = 1)
tb_pref <- as_tibble(cbind(ar_lambda)) %>%
  expand_grid(ar_beta) %>% expand_grid(ar_v_star) %>%
  rename_all(~c('lambda', 'beta', 'vstar')) %>%
  rowid_to_column(var = "indiff_id")

# Generate indifference points with apply and anonymous function
# tb_pref, whatever is selected from it, must be all numeric
# if there are strings, would cause conversion error.
ls_df_indiff <- apply(tb_pref, 1, function(x){
  indiff_id <- x[1]
  lambda <- x[2]
  beta <- x[3]
  vstar <- x[4]
  ar_fl_A_indiff <- seq(fl_A_min, fl_A_max, length.out=it_A_grid)
  ar_fl_B_indiff <- (((vstar^lambda) -
    (beta*ar_fl_A_indiff^(lambda)))/(1-beta))^(1/lambda)
  mt_A_B_indiff <- cbind(indiff_id, lambda, beta, vstar,
    ar_fl_A_indiff, ar_fl_B_indiff)
  colnames(mt_A_B_indiff) <- c('indiff_id', 'lambda', 'beta', 'vstar',
    'indiff_A', 'indiff_B')
  tb_A_B_indiff <- as_tibble(mt_A_B_indiff) %>%
    rowid_to_column(var = "A_grid_id") %>%
    filter(indiff_B >= 0 & indiff_B <= max(ar_fl_A_indiff))
  return(tb_A_B_indiff)
```

```
})
df_indiff <- do.call(rbind, ls_df_indiff) %>% drop_na()
```

Note that many more A grid points are needed to fully plot out the leontief line.

```
# Labeling
st_title <- paste0('Indifference Curves Aktinson Atkinson Utility (CES)')
st_subtitle <- paste0('Each Panel Different beta=A\'s Weight lambda=inequality aversion\n',
  'https://fanwangecon.github.io/',
  'R4Econ/math/func_ineq/htmlpdf/fs_atkinson_ces.html')
st_caption <- paste0('Indifference Curve 2 Individuals, ',
  'https://fanwangecon.github.io/R4Econ/')

st_x_label <- 'A'
st_y_label <- 'B'

# Graphing
plt_indiff <-
  df_indiff %>% mutate(lambda = as_factor(lambda),
    beta = as_factor(beta),
    vstar = as_factor(vstar)) %>%

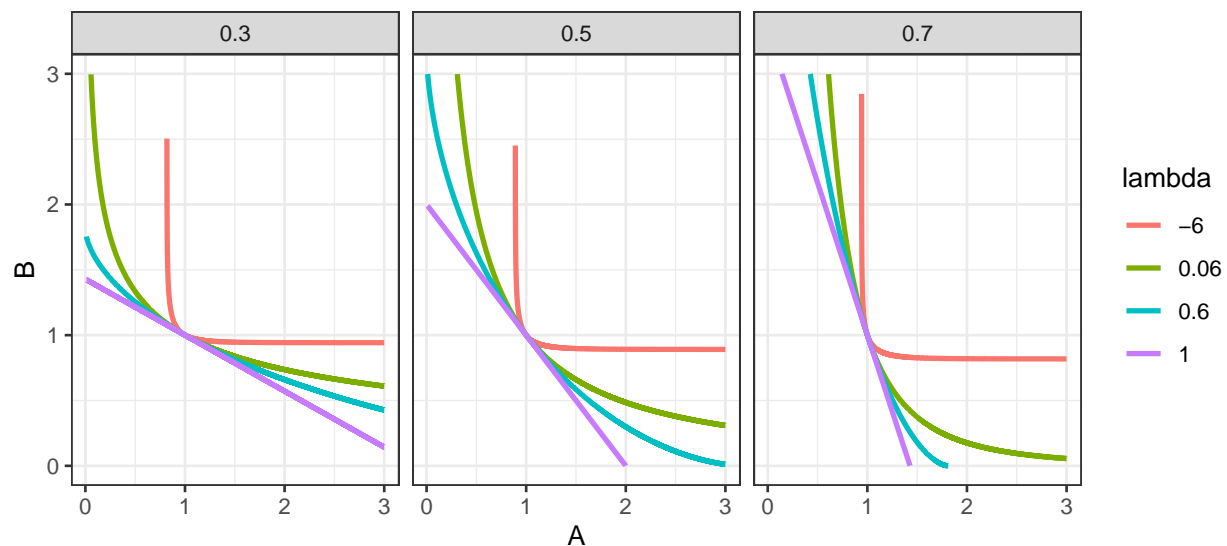
  ggplot(aes(x=indiff_A, y=indiff_B,
    colour=lambda)) +
  facet_wrap(~ beta) +
  geom_line(size=1) +
  labs(title = st_title, subtitle = st_subtitle,
    x = st_x_label, y = st_y_label, caption = st_caption) +
  theme_bw()

# show
print(plt_indiff)
```

Indifference Curves Aktinson Atkinson Utility (CES)

Each Panel Different beta=A's Weight lambda=inequality aversion

https://fanwangecon.github.io/R4Econ/math/func_ineq/htmlpdf/fs_atkinson_ces.html



Indifference Curve 2 Individuals, <https://fanwangecon.github.io/R4Econ/>