# Online Appendix for The Limited Value of a Second Opinion: Competition and Exaggeration in Experimental Cheap Talk Games

William Minozzi[*] Jonathan Woon[†]

June 6, 2019

## A1 Equilibrium Specifications

Here, we explicitly state the most informative equilibria of the games corresponding to each of the experimental conditions. We focus on equilibria in which senders who are indifferent between sending informative messages and babbling send informative messages. There are many equilibria that are informationally equivalent to the ones stated here that relax this requirement.

In the single sender condition, the sender uses the message strategy

$$m^{\text{single}}(t) \in \begin{cases} U[-100,-80] & \text{if } t \leq -80 \\ U[-80,100] & \text{otherwise,} \end{cases}$$

and the receiver uses the action strategy

$$a^{\text{single}}(m) = \begin{cases} -90 & \text{if } m \leq -80 \\ 10 & \text{otherwise,} \end{cases}$$

with the beliefs

$$q^{\text{single}}(t|m) = \begin{cases} U[-100,-80] & \text{if } m \leq -80 \\ U[-80,100] & \text{otherwise.} \end{cases}$$

[*]Associate Professor, Department of Political Science, 2137 Derby Hall, Ohio State University, Columbus, OH 43210. minozzi.1@osu.edu

[†]Corresponding author. Professor, Department of Political Science, Department of Economics (secondary), and Pittsburgh Experimental Economics Laboratory, 4443 Wesley W. Posvar Hall, University of Pittsburgh, Pittsburgh, PA 15260. woon@pitt.edu

In the Aligned condition, the extremist sender, who moves first and has shift $s_E = 80$, uses $m_E^{\text{aligned}}(t) = m^{\text{single}}(t)$, the message strategy from the single sender condition. The moderate sender, who moves second and has shift $s_M = 40$, uses a similar strategy that simply ignores the extremist's message, so that $m_M^{\text{aligned}}(t, m_E) = m^{\text{single}}(t)$ for all $m_E$. The actions and beliefs of the receiver in the Aligned condition similarly ignore the extremist: $a^{\text{aligned}}(m_E, m_M) = a^{\text{single}}(m_M)$ and $q^{\text{aligned}}(t|m_E, m_M) = q^{\text{single}}(t|m_M)$. Finally, in the opposed condition, the extremist sender, who moves first and has shift $s_E = -80$, uses

$$m_E^{\text{opposed}}(t) = \begin{cases} t & \text{if } t < 20 \\ U[20, 100] & \text{otherwise.} \end{cases}$$

The moderate sender, who moves second and has shift $s_M = 40$, plays

$$m_M^{\text{opposed}}(t, m_E) = \begin{cases} t & \text{if } m_E = t < 20 \\ U[20, 100] & \text{if } t \geq 20 \\ m_E & \text{if } t < m_E \;\&\; t < 20 \text{ (off-the-path)} \\ \max(m_E + 80, t + 40) & \text{if } m_E < t < 20 \text{ (off-the-path).} \end{cases}$$

Finally, the receiver uses

$$a^{\text{opposed}}(m_M, m_E) = \begin{cases} m_E & \text{if } m_E < 20 \;\&\; m_M = m_E \\ 60 & \text{if } m_E > 20 \text{ or } m_M > 20 \\ m_E & \text{if } m_E, m_M < 20 \;\&\; m_M < m_E + 80 \text{ (off-the-path)} \\ m_M & \text{if } m_E, m_M < 20 \;\&\; m_M \geq m_E + 80 \text{ (off-the-path),} \end{cases}$$

and has the beliefs

$$q^{\text{opposed}}(t|m_M, m_E) = \begin{cases} \mathbb{I}[t = m_E] & \text{if } m_E < 20 \;\&\; m_M = m_E \\ 60 & \text{if } m_E > 20 \text{ or } m_M > 20 \\ \mathbb{I}[t = m_E] & \text{if } m_E, m_M < 20 \;\&\; m_M < m_E + 80 \text{ (off-the-path)} \\ \mathbb{I}[t = m_M] & \text{if } m_E, m_M < 20 \;\&\; m_M \geq m_E + 80 \text{ (off-the-path).} \end{cases}$$

# A2 Statistical Model for Figure 4

To model message strategies, we used a simultaneous, censored, mixed effects regression model written and estimated in the Stan modeling language (Carpenter et al., 2016). Code for the model appears below.

```
data {
  // observations are split into four datasets based on whether
  // (1) message midpoint hits the boundary and
  // (2) message range hits the maximum size

  int<lower = 1> N_id; // number of subjects
  int<lower = 1> N_period; // number of periods

  // observations in which midpoint does not hit boundary and range
  //  does not hit maximum value
  int<lower = 1> N_1;
  int<lower = 1, upper = N_id> id_1[N_1];
  int<lower = 1, upper = N_period> period_1[N_1];
  vector[N_1] Target_1;
  vector[N_1] Message_Midpoint_1;
  vector[N_1] log_Message_Range_Size_1;

  // observations in which midpoint does not hit boundary but range
  //  does hit maximum value
  int<lower = 0> N_2;
  int<lower = 1, upper = N_id> id_2[N_2];
  int<lower = 1, upper = N_period> period_2[N_2];
  vector[N_2] Target_2;
  vector[N_2] Message_Midpoint_2;
  vector[N_2] log_Max_Message_Range_Size_2;

  // observations in which midpoint hits upper boundary
  int<lower = 0> N_3;
  int<lower = 1, upper = N_id> id_3[N_3];
  int<lower = 1, upper = N_period> period_3[N_3];
  vector[N_3] Target_3;
  real U; // right censor point for Message_Midpoint

  // observations in which midpoint hits lower boundary
  int<lower = 0> N_4;
  int<lower = 1, upper = N_id> id_4[N_4];
  int<lower = 1, upper = N_period> period_4[N_4];
  vector[N_4] Target_4;
  real L; // left censor point for Message_Midpoint
}

parameters {
  real a0;              // grand mean for intercept in midpoint model
  vector[N_id] a_e_id; // (unscaled) subject-level random intercept
  real<lower = 0> a_s_id; // scale for subject-level random intercept
  real b0;              // grand mean for slope in midpoint model
  vector[N_id] b_e_id; // (unscaled) subject-level random slope
  real<lower = 0> b_s_id; // scale for subject-level random slope
  real<lower = 0> s_midpoint; // scale for error in midpoint model
  real g0;              // grand mean for intercept in range model
  vector[N_id] g_e_id; // (unscaled) subject-level random intercept
  real<lower = 0> g_s_id; // scale for subject-level random intercept
  real h0;              // grand mean for slope in range model
```

```
  vector[N_id] h_e_id; // (unscaled) subject-level random slope
  real<lower = 0> h_s_id; // scale for subject-level random slope
  real<lower = 0> s_logrange; // scale for error in range model
}

model {
  // local declarations
  // containers for subject-level coefficients
  vector[N_id] a_id;
  vector[N_id] b_id;
  vector[N_id] g_id;
  vector[N_id] h_id;
  // containers for midpoint model linear predictors
  vector[N_1] m_midpoint_1;
  vector[N_2] m_midpoint_2;
  vector[N_3] m_midpoint_3;
  vector[N_4] m_midpoint_4;
  // containers for range model linear predictors
  vector[N_1] m_logrange_1;
  vector[N_2] m_logrange_2;
  vector[N_3] m_logrange_3;
  vector[N_4] m_logrange_4;

  // local definitions
  a_id = a0 + a_s_id * (a_e_id - mean(a_e_id));
  b_id = b0 + b_s_id * (b_e_id - mean(b_e_id));
  g_id = g0 + g_s_id * (g_e_id - mean(g_e_id));
  h_id = h0 + h_s_id * (h_e_id - mean(h_e_id));
  for (n in 1:N_1) {
    m_midpoint_1[n] = a_id[id_1[n]] + b_id[id_1[n]] * Target_1[n];
    m_logrange_1[n] = g_id[id_1[n]] + h_id[id_1[n]] * Target_1[n];
  }
  for (n in 1:N_2) {
    m_midpoint_2[n] = a_id[id_2[n]] + b_id[id_2[n]] * Target_2[n];
    m_logrange_2[n] = g_id[id_2[n]] + h_id[id_2[n]] * Target_2[n];
  }
  for (n in 1:N_3) {
    m_midpoint_3[n] = a_id[id_3[n]] + b_id[id_3[n]] * Target_3[n];
    m_logrange_3[n] = g_id[id_3[n]] + h_id[id_3[n]] * Target_3[n];
  }
  for (n in 1:N_4) {
    m_midpoint_4[n] = a_id[id_4[n]] + b_id[id_4[n]] * Target_4[n];
    m_logrange_4[n] = g_id[id_4[n]] + h_id[id_4[n]] * Target_4[n];
  }

  // prior
  a0 ~ normal(0, 10);
  a_s_id ~ cauchy(0, 2.5);
  a_e_id ~ normal(0, 1);
  b0 ~ normal(0, 10);
  b_s_id ~ cauchy(0, 2.5);
  b_e_id ~ normal(0, 1);
  s_midpoint ~ cauchy(0, 2.5);
  g0 ~ normal(0, 10);
```

```
  g_s_id ~ cauchy(0, 2.5);
  g_e_id ~ normal(0, 1);
  h0 ~ normal(0, 10);
  h_s_id ~ cauchy(0, 2.5);
  h_e_id ~ normal(0, 1);
  s_logrange ~ cauchy(0, 2.5);

  // posterior
  Message_Midpoint_1 ~ normal(m_midpoint_1, s_midpoint);
  Message_Midpoint_2 ~ normal(m_midpoint_2, s_midpoint);
  log_Message_Range_Size_1 ~ cauchy(m_logrange_1, s_logrange);
  // increment log probability for censoring
  target +=
    cauchy_lccdf(log_Max_Message_Range_Size_2 | m_logrange_2, s_logrange) +
    normal_lccdf(U | m_midpoint_3, s_midpoint) +
    normal_lcdf(L | m_midpoint_4, s_midpoint);
}
```

Table A1: Model of Midpoint & Range

| | Estimate | Std. Error | 95% CI |
|---|---|---|---|
| Midpoint Equation | | | |
| Intercept | 0.28 | 0.01 | [0.27, 0.30] |
| Target | 0.62 | 0.01 | [0.60, 0.64] |
| Subject Intercept Scale | 0.48 | 0.03 | [0.43, 0.54] |
| Subject Slope Scale | 0.23 | 0.02 | [0.19, 0.27] |
| Error Scale | 0.43 | 0.01 | [0.42, 0.44] |
| log(Range) Equation | | | |
| Intercept | 0.21 | 0.00 | [0.20, 0.22] |
| Target | 0.04 | 0.01 | [0.02, 0.05] |
| Subject Intercept Scale | 0.13 | 0.01 | [0.11, 0.15] |
| Subject Slope Scale | 0.09 | 0.01 | [0.07, 0.11] |
| Error Scale | 0.08 | 0.00 | [0.08, 0.09] |

$n = 2190$. This table presents the estimated model based on the above code, and depicted in Figure 4. The constant is suppressed, as are the constituent terms for Moderate and Extremist, as the use of all conditions as interaction terms permits direct estimation of quantites of interest. The model was fitted in Stan 2.10, with four chains, each with 1000 warmup iterations and 1000 sampling iterations, after which diagnostics indicated convergence (e.g., $\hat{R} \leq 1.1$ and $n_{\text{eff}} > 400$ for all parameters).

## Table A2: Message Models over Time (Min and Max)

| | Single | | Aligned Extremist | | Opposed Extremist | | Aligned Moderate | | Opposed Moderate | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Min | Max | Min | Max | Min | Max | Min | Max | Min | Max |
| Target | 0.48*** | 0.74*** | 0.82*** | 0.30*** | 0.41*** | 0.57*** | 0.65*** | 0.59*** | 0.69*** | 0.65*** |
| | (0.03) | (0.02) | (0.03) | (0.03) | (0.03) | (0.04) | (0.03) | (0.03) | (0.05) | (0.04) |
| Second Half | 54.60*** | 0.30*** | 0.35*** | 12.67*** | −8.47** | −18.38*** | 4.84 | 0.36 | 19.62*** | 16.50*** |
| | (4.24) | (0.02) | (0.03) | (2.85) | (2.87) | (3.79) | (5.25) | (4.06) | (5.83) | (4.93) |
| Target × Second Half | 0.02 | −5.06 | −5.94 | −0.11* | −0.11* | −0.22*** | −0.01 | −0.01 | −0.18* | −0.18** |
| | (0.04) | (4.53) | (5.62) | (0.05) | (0.05) | (0.07) | (0.05) | (0.04) | (0.07) | (0.06) |
| Extremist Message | | | | | | | 0.04 | 0.07 | −0.05 | −0.06 |
| | | | | | | | (0.05) | (0.04) | (0.07) | (0.05) |
| Extremist Message × Second Half | | | | | | | 0.03 | 0.06 | 0.00 | 0.02 |
| | | | | | | | (0.07) | (0.05) | (0.09) | (0.07) |
| Intercept | −22.51*** | −22.73*** | −29.18*** | 65.29*** | −63.64*** | −29.21** | 11.38 | 37.58*** | 14.31** | 37.02*** |
| | (3.13) | (2.94) | (3.82) | (3.54) | (5.82) | (11.01) | (8.48) | (3.52) | (5.41) | (6.87) |
| $n$ Observations | 792 | 792 | 696 | 696 | 696 | 696 | 696 | 696 | 696 | 696 |
| $n$ Subjects | 33 | 33 | 29 | 29 | 29 | 29 | 29 | 29 | 29 | 29 |
| $n$ Sessions | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 |
| *Error terms* | | | | | Group SD | | | | | |
| Subject | 3.0 | 1.7 | 8.0 | 4.3 | 3.9 | 3.6 | 3.8 | 2.0 | 8.1 | 5.1 |
| Session | 23.1 | 11.6 | 15.7 | 6.4 | 13.3 | 27.3 | 19.3 | 6.3 | 8.1 | 14.5 |
| Residual | 39.8 | 30.8 | 51.1 | 37.2 | 37.3 | 50.6 | 39.4 | 29.6 | 49.4 | 40.1 |

The table presents the results of mixed effects models of the the minima and maxima of senders' message intervals on *Target*, an indicator for the *Second Half* of periods, and their interaction. Separate models are presented by sender type and outcome variable. The models also include random intercepts for Session and Subject to control for the panel structure of the data. $^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$.

# References

Carpenter, Bob, Andrew Gelman, Matt Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Michael A. Brubaker, Jiqiang Guo, Peter Li and Allen Riddell. 2016. "Stan: A probabilistic programming language." *Journal of Statistical Software* in press. http://mc-stan.org/.
**URL:** *http://mc-stan.org/*