Guideline for the Rational Agent Framework

2023-06-27

This document gives a guideline for how to apply the rational agent framework to your visualization experiment.

Before we get started

Before we get started to applying the framework to a visualization experiment, we first need to know the decision-making problem underneath the task in the experiment, which includes…

- **states** (the description of reality we want help users understand with visualization, e.g. whether it is frozen in our weather forecast example).
- **data generating models** (all possible distributions that states are drawn from, e.g. the possibility of freezing, which is also what the visualization displays).
- an **action space** (the report that the experiment requires users to give, e.g. whether to salt the parking lot).
- **signals** (the visualizations you show to users for helping them make decisions).
- a **scoring rule** (the quality or payoff that the experiment scores to users depending on their action and the state in reality, e.g. how many you may lose if you salt but no freezing or don't salt but freezing).

For example, in a weather forecast example (slightly different from what's in the paper), where we are asking users to report their belief about the possibility of freezing in discrete levels after view a visualization about daily lowest temperature, the decision-making problem could be formalized as

- states: $\theta \in \Theta = \{0 = \text{not freezing}, 1 = \text{freezing}\}$.
- data generating models: daily low temperature $t \sim N(\mu, \sigma^2)$; $\Pr[\theta = 1] = \Pr[t \leq 0]$; $\mu = 5$ fixed and σ uniformly from $\{2, 3, 4, 5\}$.
- action space: $b \in B = \{0, 0.02, 0.04, 0.06, ..., 0.96, 0.98, 1\}$, reporting the belief of freezing in discrete levels.
- signals: $v \in V$ showing daily low temperature t.
- scoring rule: $S(b, \theta) = -(b - \theta)^2$ for reporting belief of freezing.

Simulated generation of data using quantile response

We generate the experimental data for weather forecast example using quantal response equilibrium, where we keep the distribution of sigma following uniform distribution.

```
action_space = seq(0, 1, 0.02)
p_d = c(0.25, 0.25, 0.25, 0.25)

data = data.frame(
  mu = 5,
  sigma = c(2, 3, 4, 5),
  vis = c("mean", "mean+interval", "gradient", "HOPs")
) %>%
  data_grid(mu, sigma, vis) %>%
  rowwise() %>%
  mutate(
    freezing_prob = pnorm(0, mu, sigma),
```

```r
    behavioral_action = list(sample(action_space, 100,
                                    replace = TRUE,
                                    prob = exp(-100 * (action_space - freezing_prob) * (action_sp
  ) %>%
  unnest(cols = c(behavioral_action))

data
```

```
## # A tibble: 1,600 x 5
##       mu sigma vis      freezing_prob behavioral_action
##    <dbl> <dbl> <chr>            <dbl>             <dbl>
##  1     5     2 gradient       0.00621              0.08
##  2     5     2 gradient       0.00621              0.16
##  3     5     2 gradient       0.00621              0.04
##  4     5     2 gradient       0.00621              0.04
##  5     5     2 gradient       0.00621              0
##  6     5     2 gradient       0.00621              0.02
##  7     5     2 gradient       0.00621              0.04
##  8     5     2 gradient       0.00621              0.12
##  9     5     2 gradient       0.00621              0.2
## 10     5     2 gradient       0.00621              0.12
## # i 1,590 more rows
```

Pre-experiment analysis

In this section, we will show how to generate the three pre-experiment quantities (rational baseline, rational benchmark, and value of information) that the rational agent framework offers. These three quantities can be calculated by the rational agent computationally based on the belief. Assuming that the rational agent's prior belief is the distribution of all possible data generating models, the rational agent will update their belief on data generating models by the implied distribution of them in visualizations. Trying to make it clear what happening in the framework, we provide pseudo code for a general description and show a specific example about weather forecast providing real codes with the simulated data set generated above.

Rational baseline

To calculate the rational baseline, we first extract the prior belief of rational agent and its action. Then we test what score would the rational agent get if they take the experiment with only prior knowledge.

- Pseudo code:

Input: the experimental data D with each row representing one experiment data, the distribution of data generating models p, action space A, scoring rule S

Output: rational baseline

$prior\_belief \leftarrow E_{d \sim p}(d)$

$max\_payoff \leftarrow$ -inf

$optimal\_action \leftarrow 0$

FOR a in A

IF $E_{\theta \sim prior\_belief}(S(a, \theta)) > max\_payoff$

$max\_payoff \leftarrow E_{\theta \sim prior\_belief}(S(a, \theta))$

$optimal\_action \leftarrow a$

END IF

END FOR

$sum\_payoff \leftarrow 0$

FOR row in D

dgm = data generating model used in row

sum_payoff = sum_payoff + $E_{\theta \sim dgm}(S(optimal_action, \theta))$

END FOR

avg_payoff = sum_payoff / nrow(D)

RETURN avg_payoff

- R in weather forecast example

Define the scoring rule…

```
scoring_rule = function(action, state) {
  -(action - state) * (action - state)
}
```

Calculate the prior belief of freezing, i.e. cdf(0) of the expectation of four normal distributions. The prior belief of not freezing could be deduced from freezing.

```
sigma_choices = c(2, 3, 4, 5)
mu = 5
prior_belief = Reduce("+", sapply(sigma_choices, function(m) {pnorm(0, mu, m)})) / 4
prior_belief
```

```
## [1] 0.07957626
```

The prior belief says the expectation of the possibility of freezing would be 0.07957626, so without viewing any visualization signals, the rational agent beliefs the possibility of freezing is very low. Then we calculate the optimal payoff and action.

```
payoffs = lapply(action_space, function(a) {
  scoring_rule(a, prior_belief)
})
prior_action = action_space[[which.max(payoffs)]]

prior_payoff = mean((data %>%
  mutate(payoff = scoring_rule(prior_action, freezing_prob)))$payoff)

c(prior_payoff, prior_action)
```

```
## [1] -0.003331759  0.080000000
```

The result says the optimal expected payoff of the rational agent with only prior knowledge is -0.003331759 and optimal action is 0.08, meaning that the rational agent with only prior belief would always report 0.08 as the possibility of freezing and the rational baseline is -0.003331759.

Rational benchmark

Then we simulate the rational agent on the experimental data. After viewing the visualization signals, the rational agent will update the belief about states based on their belief about the distribution about data generating model. For example, the visualization signals vis provides an information about the possibility that this visualization could occur in data generating models d as $p(vis|d)$ and the prior belief about the distribution of data generating models is $p(d)$. Then the posterior belief about the distribution of data generating models could be $p(d|vis) = p(vis|d) \cdot p(d)/p(vis)$.

- Pseudo code

Input: the experimental data D with each row representing one experiment data, the distribution of data generating models p, action space A, scoring rule S

Output: rational benchmark

$sum\_payoff \leftarrow 0$

FOR row in D

vis = visualization used in row

the posterior belief of the distribution of data generating models $p(d|vis) = p(vis|d) \cdot p(d)/p(vis)$

$max\_payoff \leftarrow$ -inf

$optimal\_action \leftarrow 0$

FOR a in A

IF $E_{\theta \sim posterior\_belief}(S(a, \theta)) > max\_payoff$

$max\_payoff \leftarrow E_{\theta \sim posterior\_belief}(S(a, \theta))$

$optimal\_action \leftarrow a$

END IF

END FOR

dgm = data generating model used in row

sum_payoff = sum_payoff + $E_{\theta \sim dgm}(S(optimal_a ction, \theta))$

END FOR

avg_payoff = sum_payoff / nrow(D)

RETURN avg_payoff

- R in weather forecast example

In the weather forecast example, the visualizations provide perfect information about the variance of data except "mean" visualization, so the updated distribution of data generating models could converge to a situation that only one data generating model has possibility with almost 1 and others are all almost 0. In this case, we assume that the rational agent would know which data generating model is using with certainty after viewing one visualization. Formally, assuming that vis provides perfect information about data generating model d_1, then

$$p(vis|d_1) \approx 1 p(vis|d_2) \approx 0 ... p(vis|d_n) \approx 0$$

so $p(d_1|vis) = p(vis|d) \cdot p(d)/p(vis) \approx 1$ when the visualization is fixed.

```r
sum_payoff = 0
for (i in 1:nrow(data)) {
  posterior_belief = data[[i, "freezing_prob"]]
  posterior_action = action_space[[which.max(lapply(action_space, function(a) {
    scoring_rule(a, posterior_belief)
  }))]]
  sum_payoff = sum_payoff + scoring_rule(posterior_action, data[[i, "freezing_prob"]])
}
posterior_payoff = sum_payoff / nrow(data)
posterior_payoff
```

```
## [1] -3.324445e-05
```

The score of rational agent with posterior knowledge, i.e. the rational benchmark, is $-3.324445e-05$.

Value of information

We define the value of information in rational agent framework as the difference between rational benchmark and baseline, i.e. the maximum improvement can be made when agent has the right prior knowledge and follows the Bayesian rule to update the belief.

In the example of weather forecast, the value of information is `posterior_payoff - prior_payoff`, i.e. 0.003298514.

Post-experiment analysis

In this section, we will calculate the score of behavioral agents and the score of the calibrated behavioral.

Behavioral score

We use the scoring rule on behavioral agents' actions.

- Pseudo code

Input: the experimental data D with each row representing one experiment data, scoring rule S

Output: rational benchmark

$sum\_payoff \leftarrow 0$

FOR row in D

dgm = data generating model used in row

$action$ = action made in row

$sum\_payoff = sum\_payoff + E_{\theta \sim dgm}(S(action, \theta))$

END FOR

$avg\_payoff = sum\_payoff$ / nrow(D)

RETURN avg_payoff

- R in weather forecast example

```
behavioral_payoff = data %>%
  rowwise() %>%
  mutate(payoff = scoring_rule(behavioral_action, freezing_prob)) %>%
  dplyr::group_by(vis) %>%
  summarise(payoff = mean(payoff))
behavioral_payoff
```

```
## # A tibble: 4 x 2
##   vis             payoff
##   <chr>            <dbl>
## 1 HOPs          -0.00407
## 2 gradient      -0.00403
## 3 mean          -0.00421
## 4 mean+interval -0.00430
```

Since the behavioral agent's actions are generated randomly but not from real persons, their payoffs on all visualization types are below rational baseline. Then there comes the calibrated behavioral score, which calibrates the behavioral agent's action by Bayesian rule and calibrates the score into the interval between baseline and benchmark.

Calibrated behavioral score

Now there is another benefit of the rational agent framework. We can use the rational agent to calibrate the non-Bayesian behavioral agent into Bayesian and also calibrate the behavioral score into the interval between rational baseline and benchmark. Here is how it works.

In the calibration, the rational agent is not using the visualizations as signals but using the actions of behavioral agent as signals. To do this, the rational agent uses the joint distribution between behavioral agent's action and data generating models, and when getting a behavioral agent's action, they will look at what's the possibilities that this action will appear in different data generating models ($p(a|d)$) and use these possibilities to update the prior belief ($p(d|a) = p(a|d) \cdot p(d)/p(a)$).

The pseudo code is as following…

- Pseudo code

Input: the experimental data D with each row representing one experiment data, the distribution of data generating models p, action space A, scoring rule S

Output: rational benchmark

$sum\_payoff \leftarrow 0$

FOR row in D

a_beh = behavioral action in row

the posterior belief of the distribution of data generating models in calibration $p(d|a_beh) = p(a_beh|d) \cdot p(d)/p(a_beh)$

$max\_payoff \leftarrow$ -inf

$optimal\_action \leftarrow 0$

FOR a in A

IF $E_{\theta \sim posterior\_belief}(S(a, \theta)) > max\_payoff$

$max\_payoff \leftarrow E_{\theta \sim posterior\_belief}(S(a, \theta))$

$optimal\_action \leftarrow a$

END IF

END FOR

dgm = data generating model used in row

$sum\_payoff = sum\_payoff + E_{\theta \sim dgm}(S(optimal_action, \theta))$

END FOR

$avg\_payoff = sum\_payoff$ / nrow(D)

RETURN avg_payoff

- R in weather forecast example

```
joint_action_distribution = data %>%
  dplyr::group_by(mu, sigma, freezing_prob, behavioral_action) %>%
  summarise(count = n()) %>%
  dplyr::group_by(mu, sigma, freezing_prob) %>%
  mutate(count = count / sum(count))
```

```
## `summarise()` has grouped output by 'mu', 'sigma', 'freezing_prob'. You can
## override using the `.groups` argument.
```

```
sum_payoff = 0
for (i in 1:nrow(data)) {
  action = data[[i, "behavioral_action"]]
  p_a = nrow(data %>% filter(behavioral_action == action)) / nrow(data)
  action_joint = joint_action_distribution %>%
    filter(behavioral_action == action)
  sigmas = action_joint$sigma
  distributions = pnorm(0, mu, sigma_choices)
  p_a_d = as.list(rep(0, length(sigma_choices)))
  names(p_a_d) = as.character(sigma_choices)
  p_a_d[as.character(sigmas)] = action_joint$count  # p(a|d)
  p_a_d = unname(unlist(p_a_d))
  p_d_a = p_a_d * p_d / p_a  # p(a|d) * p(d) / p(a)

  calibrated_belief = sum(distributions * p_d_a)  # E_p_d_a(d)
  calibrated_action = action_space[[which.max(lapply(action_space, function(a) {
    scoring_rule(a, calibrated_belief)
  }))]]
  data[[i, "calibrated_payoff"]] = scoring_rule(calibrated_action, data[[i, "freezing_prob"]])
}
calibrated_payoff = data %>%
  dplyr::group_by(vis) %>%
  summarise(calibrated_payoff = mean(calibrated_payoff))
calibrated_payoff
```

```
## # A tibble: 4 x 2
##   vis           calibrated_payoff
##   <chr>                     <dbl>
## 1 HOPs                   -0.00229
## 2 gradient               -0.00233
## 3 mean                   -0.00232
## 4 mean+interval          -0.00206
```

Now we get the calibrated behavioral scores. The calibrated scores of all visualization types are between rational baseline and rational benchmark. The score for “mean” visualization is more near the baseline, which meets our previous assumption that “mean” visualization provides no information for this decision-making task because it shows nothing about the variance.