Analysis: Multiplayer Concept Learning Pilot #1

Sahil Chopra

March 28, 2018

Introduction

Last time, we tried replicating the results of the Piantadosi 2016 paper. While the results seemed to indicate that the participants may be learning, it was somewhat inconclusive because of errors in the experimental procedure on our behalf. Namely, there was no held out test set, and critters were sampled with replacement during training, as there were only 27 unique critters but 50 trials.

Here we correct for these errors by:

1) Increasing the unique critter stimuli set from 27 to 81
2) Creating 2 lists: 1 Train Set (50 critters) & 1 Test Set (31 critters)

Additionally, we expand the single player game into multiplayer game, as detailed below.

Experimental Setup

Stimuli Generation

For this experiment, we generate stimuli by permuting four properties, each with three different possible values. The four properties are as follows: primary color, secondary color, size, and critter species. The primary color can either be blue, green, orange. The secondary color can be either red, yellow, or purple. The size is logarithmic in the relative differences between small, medium, and large. The possible critter species are fish, bug, and bird.

We enumerate all 81 possible critters and randomly split them into two groups – train (50) and test (31).

Selected Concepts

For this pilot experiment, we only gathered data about one difficult concept:

(Primary Color = Blue) XOR (Critter Species = Fish)

Running the Experiment

We ran the experiment on MTurk with two participants per game. The participants were split into teachers and listeners, with one teacher and one listener per game.

The teacher would begin before a listener joined the game itself. The teacher would have 50 training rounds, where they had to a label the given critter as "wudsy" or "not wudsy". At first teachers had to guess, but after each round we added the previous stimulus to a table, boxing the "wudsy" ones. If the teacher incorrectly labeled a critter they had to wait 5 seconds before the next round.

Once the teacher completed their 50 rounds of training, the student met them in an online chatroom, where they forced to talk for at least 30 seconds. Here, the teacher tried to explain the definition of a "wudsy" creature to the listener. After discussing the concept sufficiently, the listener would hit "continue" – progressing both participants to the test portion of the game, where both the teacher and the listener were given the same 31 held out critters and evaluated in terms of their performance on the task.

Results

We logged performance of the teachers during training, the conversations during the chat room, and the scores of both the teacher and listener during the test phase. Here we present the results of this pilot experiment. In total, we ran 9 games.

Places of Error

There far fewer places of error in this pilot experiment. The greatest error was failure to properly log game_ids to Mturk. This made it a bit more of a hassle to properly stitch together participants from the same game.

Additionally, I did not set a properly long duration for the hit itself. In the previous experiment, I set the hit duration to 30 minutes and forgot to extend that here – given that this is a multiplayer game. There was an instance or two of games that stalled out for this reason – but I properly compensated these individuals for their time, with separate compensation hits.

I have fixed both of these issues for subsequent versions of this experiment.

Data Preprocessing

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(purrr)
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
```

```
## The following object is masked from 'package:purrr':
##
##     flatten
```

```
library(tidyjson)
```

```
##
## Attaching package: 'tidyjson'
```

```
## The following object is masked from 'package:jsonlite':
##
##     read_json
```

```
library(ggplot2)
library(rwebppl)
```

```
## using webppl version: v0.9.7 /Library/Frameworks/R.framework/Versions/3.3/Resources/library/rwebppl/
```

```
library(knitr)

# Read Raw Data
setwd("../../mturk/mp-game-3/pilot")
summary_data <- read_csv(
  file="./mp-game-3-mturk.csv",
  col_names=TRUE,
  col_types = "icccccccccccccccccccccccccccccccccccc"
)
```

```
## Warning: Duplicated column names deduplicated: 'workerid' =>
## 'workerid_1' [21]
```

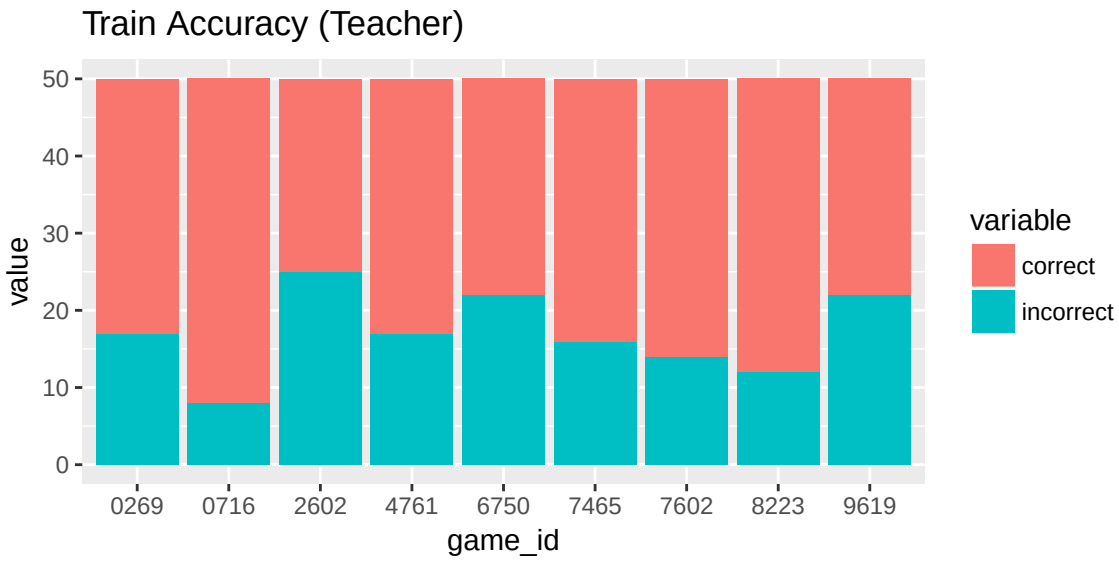
```
num_train = 50
num_test = 31

# Reduce Data to Essentials
summary_data_reduced <- select(
  summary_data,
  workerid,
  Answer.role,
  Answer.training_trials,
  Answer.testing_trials,
  Answer.training_summary_stats,
  Answer.testing_summary_stats,
  game_id
) %>%
  rename(role = Answer.role) %>%
  rename(training_trials = Answer.training_trials) %>%
  rename(testing_trials = Answer.testing_trials) %>%
  rename(training_summary_stats = Answer.training_summary_stats) %>%
  rename(testing_summary_stats = Answer.testing_summary_stats) %>%
  mutate(role = gsub("[^0-9A-Za-z///' ]", "", role)) %>% # Remove random ""
  mutate(game_id = gsub("[^0-9A-Za-z///' ]", "", game_id)) %>%
  mutate(
    training_summary_stats = map(
      training_summary_stats,
      ~fromJSON(as.character(.x))
      )
  ) %>%
  unnest() %>%
  mutate(
    testing_summary_stats = map(
      testing_summary_stats,
```

```
      ~fromJSON(as.character(.x))
      )
  ) %>%
  unnest() %>%
  mutate(
    training_trials = map(
      training_trials,
      ~fromJSON(as.character(.x))
      )
  ) %>%
  unnest() %>%
  mutate(
    testing_trials = map(
      testing_trials,
      ~fromJSON(as.character(.x))
      )
  ) %>%
  unnest()
```

Train Accuracy (Teacher)

```
train_acc_teacher <- summary_data_reduced %>%
  filter(role == "explorer")

train_teacher_perf <- train_acc_teacher$training_summary_stats %>%
  spread_values(
    hits = jnumber("hits"),
    misses = jnumber("misses"),
    correct_rejections = jnumber("correct_rejections"),
    false_alarms = jnumber("false_alarms"),
  )

train_acc_teacher <- train_acc_teacher %>%
  mutate(correct = train_teacher_perf$hits + train_teacher_perf$correct_rejections) %>%
  mutate(incorrect = train_teacher_perf$misses + train_teacher_perf$false_alarms) %>%
  select(game_id, correct, incorrect)
```

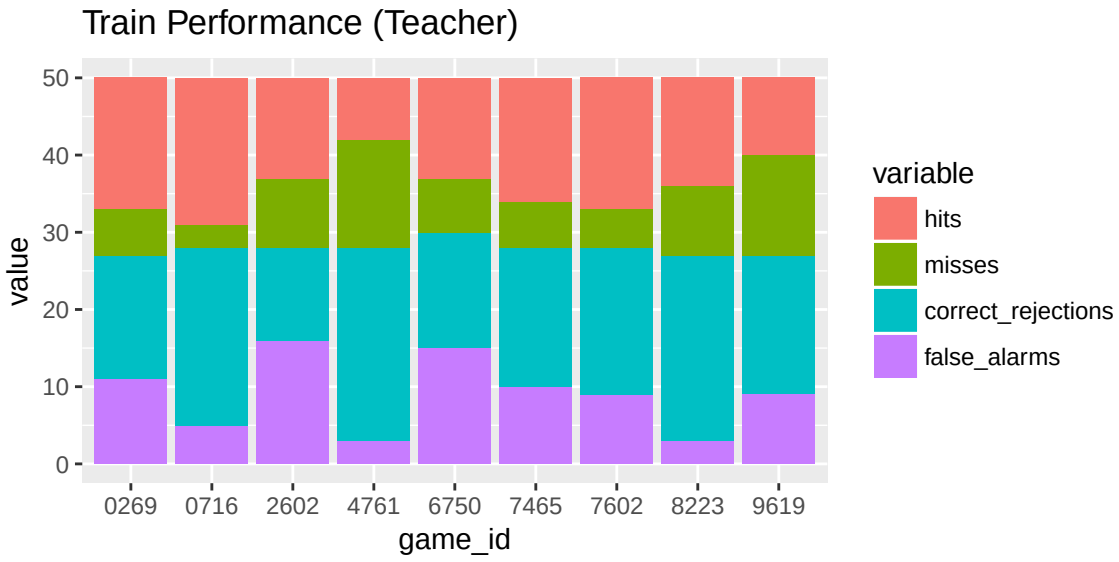
```
ggplot(melt(train_acc_teacher, id.vars="game_id")) +
  geom_bar(aes(x=game_id, y=value, fill=variable), stat="identity") +
  labs(title = "Train Accuracy (Teacher)") +
  scale_x_discrete(label=abbreviate)
```



Train Performance (Teacher)

```
train_teacher_perf <- train_teacher_perf %>%
  mutate(game_id = train_acc_teacher$game_id) %>%
  select(game_id, hits, misses, correct_rejections, false_alarms)
```

```
ggplot(melt(train_teacher_perf, id.vars="game_id")) +
  geom_bar(aes(x=game_id, y=value, fill=variable), stat="identity") +
  labs(title = "Train Performance (Teacher)") +
  scale_x_discrete(label=abbreviate)
```

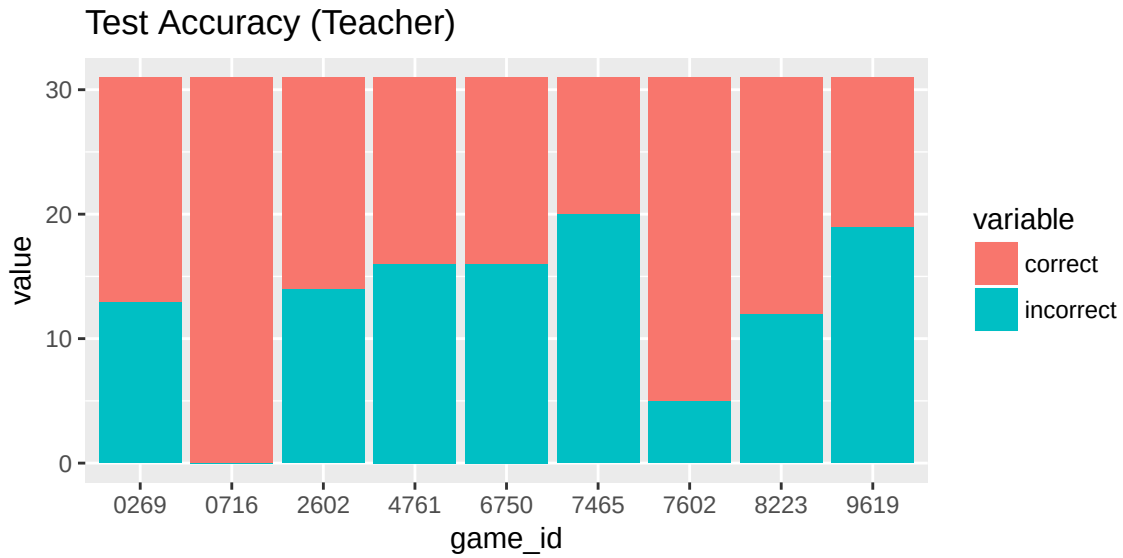


Examining the breakdown of hits/misses and correct rejections/false alarms it seems like the participants are actively playing the game – rather than simply selecting all critters as "wudsy" or "not wudsy".

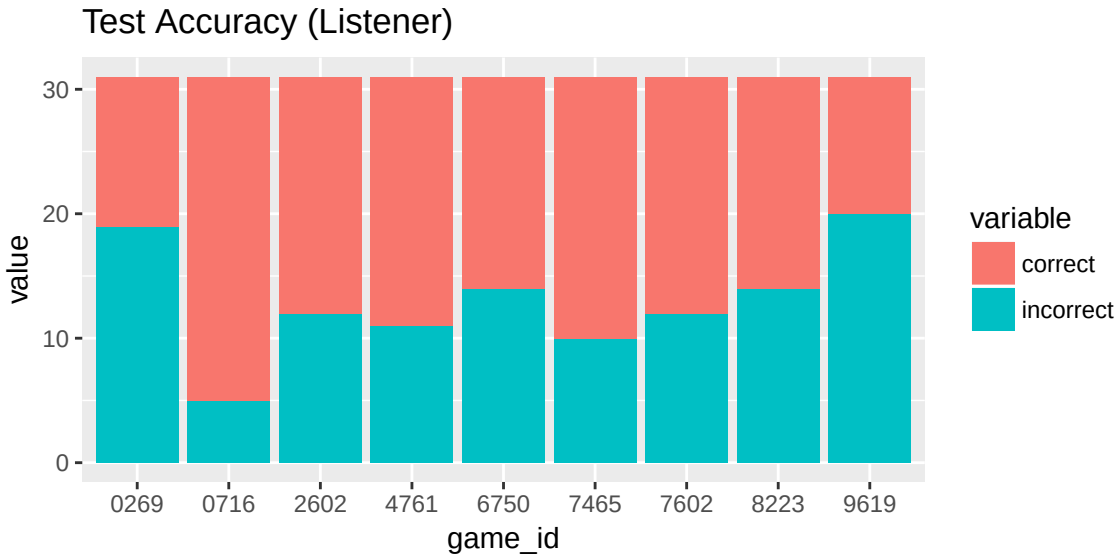
Test Accuracy (Teacher & Listener)

```
test_perf <- summary_data_reduced$testing_summary_stats %>%
  spread_values(
    hits = jnumber("hits"),
    misses = jnumber("misses"),
    correct_rejections = jnumber("correct_rejections"),
    false_alarms = jnumber("false_alarms"),
  )

test_acc <- summary_data_reduced %>%
  mutate(correct = test_perf$hits + test_perf$correct_rejections) %>%
  mutate(incorrect = test_perf$misses + test_perf$false_alarms) %>%
  select(correct, incorrect, game_id, role)

test_acc_teacher <- test_acc %>%
  filter(role == "explorer") %>%
  select(correct, incorrect, game_id)

test_acc_listener <- test_acc %>%
  filter(role == "student") %>%
  select(correct, incorrect, game_id)
```

```
ggplot(melt(test_acc_teacher, id=c("game_id"))) +
  geom_bar(aes(x=game_id, y=value, fill=variable), stat="identity") +
  labs(title = "Test Accuracy (Teacher)") +
  scale_x_discrete(label=abbreviate)
```




```
ggplot(melt(test_acc_listener, id=c("game_id"))) +
  geom_bar(aes(x=game_id, y=value, fill=variable), stat="identity") +
  labs(title = "Test Accuracy (Listener)") +
  scale_x_discrete(label=abbreviate)
```



tldr;

Incorrect teachers lead to the worst performance amongst listeners – with performance comparable between teachers and listeners. Partially informative but unsderspecific teachers lead to slightly better performances amongst listeners and better performance than the respective teachers, because of a greater rate of correct rejections. Teachers with the best performance lead to the best performing listeners, wherel listeners tend to slightly worse than their respective teachers.

The teachers with the best performance on the test set were those in games 0716 (100% accuracy) and 7602 (~84%).

Accordingly the student with the best performance, was the one from game 0716 (~84%). Examining the dialogue between teacher and student – it's evident that the teacher completely understood the concept and conveyed it succinctly to the student, without enumerating every possible "wudsy" critter:

L: Hi, what did you learn?

T: There are 3 different creatures. A bird, a fish, and a kind of bug.

L: Got it. thanks

L: what else (if anything)

T: whether they are wudsy or not depends on their color. For the fish if their MAIN color (not stripes) is green or orange they are wudsy. For birds if their body color is blue they are wudsy. For the bugs if their wings are blue they are wudsy

L: got it thanks

L: anything else?

T: write that down and you will be set!

L: done and done!

Interestingly, the second best student was from game 7465, where the teacher had the poorest performance. Examining the dialogue for this specicifc pair:

T:It seems like there is one color combo for each creature that is a wudsy

T:blue and yellow fish for example

T:red and blue bird

L: Ok perfect

The former is incorrect according to the rule, while the latter is a single example of a wudsy creature; so it is not clear why the listener performed so well here, while the teacher struggled. Examining the plots below, examining the performance of the teachers and listeners in terms of hits/misses and correct rejections/false alarms – we see that this teacher deemed many critters "wudsy", when they were not - i.e. many false alarms. The terseness of the exchange provided the speaker with an incomplete understanding of the wudsy concept – but this may have allowed them to correctly reject many of the critters for which the teacher registered false alarms.

The third best student was from game 0716, where the teacher was the second best in performance amongst the teachers. Running this experiment, I would have expected a priori that listeners would perform worse than their teachers. This holds for games 0716, 7602, 8223, and 9619. Technically, this also holds for 0269, but examining the conversation reveals that the explorer was clueless as to the wudsy definition and simply wished the listener the best of luck. In the four other games, the listener somehow out performed the speaker.

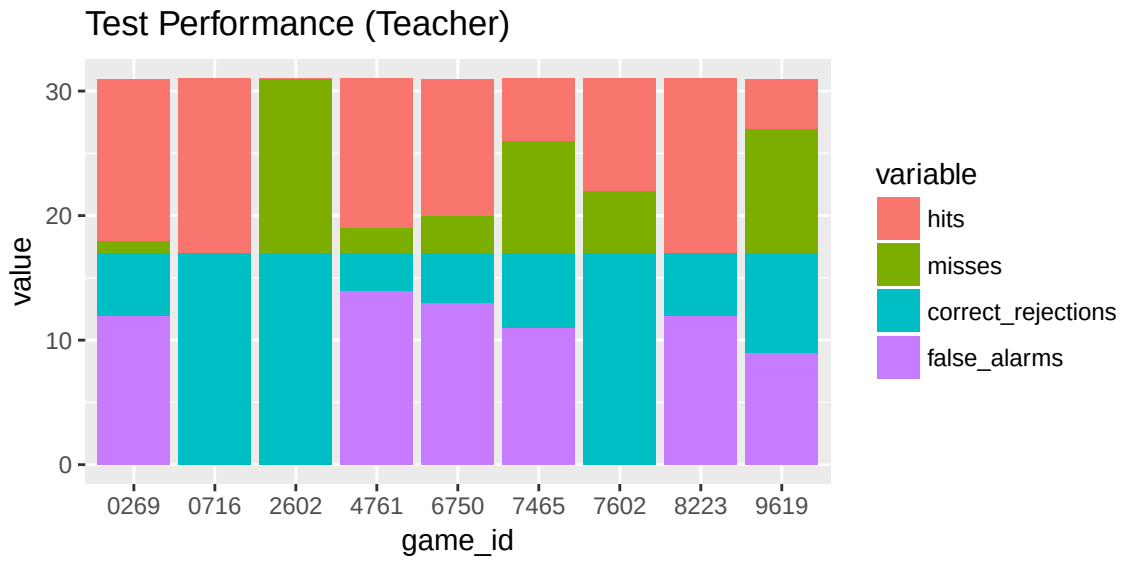
We examined one of these cases with game 7465 – where we saw that underspecification of the wudsy class allowed the listener to correctly reject critters that were not discussed. Now we turn to the other cases were listeners out performed the speakers:

In 2602, we see that the teacher incorrectly describes the wudsy critters as "skinny and little eyes". Eye size as invarible and the size of the critters had nothing to do with the rule. Thus, this is uninformative and wrong. If the teacher actually believed this, they applied this rule by rejecting every critter in the test set. Meanwhile, the listener having uncertainty over this rule and the meaning of this rules, slightly outperformed the teacher.

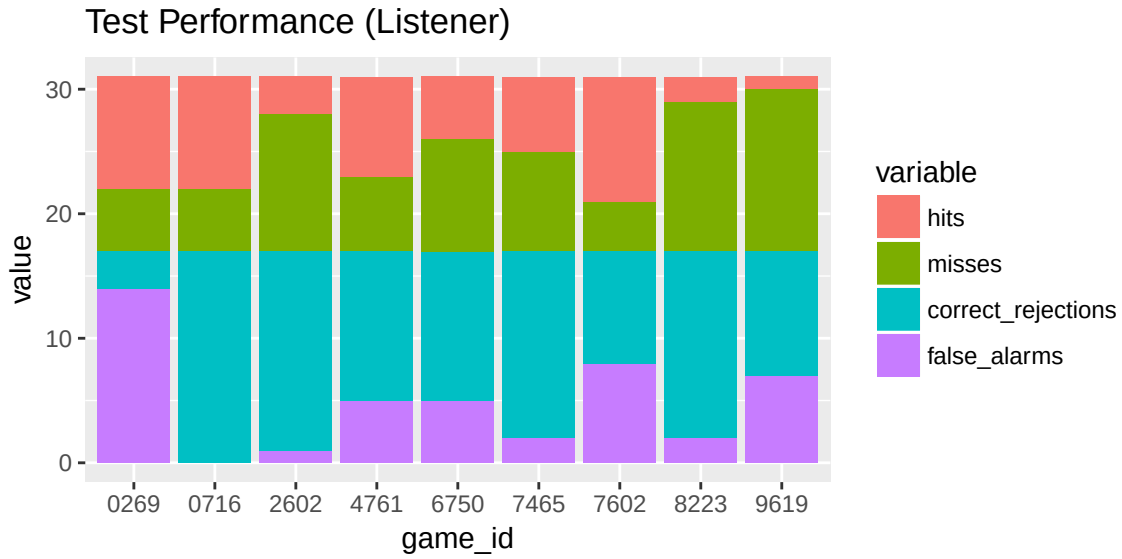
In 4761, the teacher similarly underspecified the wudsy critters by enumearating a handful of combinations of properties that yielded "wudsiness", without capturing all the variance. Here we see the same rsult as in 7465 – where the underspecification leads the listener to have significantly more "correct rejections" than the speaker. This also holds for 6750 where the teacher underspecified the wudsy creatures as "most of the fish".

Test Performance (Teacher & Listener)

```
test_perf <- summary_data_reduced %>%
  mutate(hits=test_perf$hits) %>%
  mutate(misses=test_perf$misses) %>%
  mutate(correct_rejections=test_perf$correct_rejections) %>%
  mutate(false_alarms=test_perf$false_alarms)

test_perf_teacher <- test_perf %>%
  filter(role == "explorer") %>%
  select(game_id, hits, misses, correct_rejections, false_alarms)

test_perf_listener <- test_perf %>%
  filter(role == "student") %>%
  select(game_id, hits, misses, correct_rejections, false_alarms)

ggplot(melt(test_perf_teacher, id.vars="game_id")) +
  geom_bar(aes(x=game_id, y=value, fill=variable), stat="identity") +
  labs(title = "Test Performance (Teacher)") +
  scale_x_discrete(label=abbreviate)
```




```
ggplot(melt(test_perf_listener, id.vars="game_id")) +
  geom_bar(aes(x=game_id, y=value, fill=variable), stat="identity") +
  labs(title = "Test Performance (Listener)") +
  scale_x_discrete(label=abbreviate)
```



Qualitative Language Analysis

It seems like the participants gravitated to one half of the XOR statement, usually towards color; so they captured a portion of the rule. This is a hard concept, so this is relatively expected.

Rational Rules

Before we examine the ability of rational rueles to model human performance on the task, we wanted to see if rational rules, with it's current CFG was powerful enough to learn the correct rule with which to classify the data, i.e. learn the "Body Color == Blue XOR Critter Type == Fish" rule.

RR: Ground Truth Labels

Posterior Predictives

First we examine the posterior predictions produced by rational rules, when using ground truth labels as the given labels. Note that the critter description is written as "belongsToConcept_Critter_BodyColor_SecondaryColor_Size"

Table 1: Posterior on Predictions (Rational Rules with Ground Truth Labels)

critter	prob
false_bird_green_purple_large	0
false_bird_green_red_large	0
false_bird_green_red_medium	0
false_bird_green_yellow_large	0
false_bird_orange_purple_large	0
false_bird_orange_red_large	0
false_bird_orange_red_medium	0
false_bird_orange_red_small	0
false_bird_orange_yellow_large	0
false_bug_green_purple_large	0
false_bug_green_red_large	0
false_bug_green_red_medium	0
false_bug_green_yellow_large	0
false_bug_orange_purple_large	0
false_bug_orange_red_large	0
false_bug_orange_red_medium	0
false_bug_orange_red_small	0
false_bug_orange_yellow_large	0
false_bird_green_yellow_small	0.0015
false_bird_orange_purple_small	0.0015
false_bird_orange_yellow_small	0.0015
false_bug_green_yellow_small	0.0015
false_bug_orange_yellow_small	0.0015
false_bird_green_purple_small	0.002
false_bird_green_red_small	0.002
false_bug_green_red_small	0.002
false_bird_green_yellow_medium	0.0055
false_bird_orange_yellow_medium	0.0055
false_bug_green_yellow_medium	0.0055
false_bug_orange_yellow_medium	0.0055
false_bug_orange_purple_small	0.0065
false_bug_green_purple_small	0.007
false_bird_orange_purple_medium	0.022
false_bug_orange_purple_medium	0.022
false_bird_green_purple_medium	0.025
false_bug_green_purple_medium	0.034
false_fish_blue_purple_large	0.999999999999995
false_fish_blue_purple_medium	0.999999999999995
false_fish_blue_purple_small	0.999999999999995
false_fish_blue_red_large	0.999999999999995
false_fish_blue_red_medium	0.999999999999995
false_fish_blue_red_small	0.999999999999995

critter	prob
false_fish_blue_yellow_large	0.999999999999995
false_fish_blue_yellow_medium	0.999999999999995
false_fish_blue_yellow_small	0.999999999999995
true_bird_blue_purple_large	0.999999999999995
true_bird_blue_purple_medium	0.999999999999995
true_bird_blue_purple_small	0.999999999999995
true_bird_blue_red_large	0.999999999999995
true_bird_blue_red_medium	0.999999999999995
true_bird_blue_red_small	0.999999999999995
true_bird_blue_yellow_large	0.999999999999995
true_bird_blue_yellow_medium	0.999999999999995
true_bird_blue_yellow_small	0.999999999999995
true_bug_blue_purple_large	0.999999999999995
true_bug_blue_purple_medium	0.999999999999995
true_bug_blue_purple_small	0.999999999999995
true_bug_blue_red_large	0.999999999999995
true_bug_blue_red_medium	0.999999999999995
true_bug_blue_red_small	0.999999999999995
true_bug_blue_yellow_large	0.999999999999995
true_bug_blue_yellow_medium	0.999999999999995
true_bug_blue_yellow_small	0.999999999999995
true_fish_green_purple_large	0.999999999999995
true_fish_green_purple_medium	0.999999999999995
true_fish_green_purple_small	0.999999999999995
true_fish_green_red_large	0.999999999999995
true_fish_green_red_medium	0.999999999999995
true_fish_green_red_small	0.999999999999995
true_fish_green_yellow_large	0.999999999999995
true_fish_green_yellow_medium	0.999999999999995
true_fish_green_yellow_small	0.999999999999995
true_fish_orange_purple_large	0.999999999999995
true_fish_orange_purple_medium	0.999999999999995
true_fish_orange_purple_small	0.999999999999995
true_fish_orange_red_large	0.999999999999995
true_fish_orange_red_medium	0.999999999999995
true_fish_orange_red_small	0.999999999999995
true_fish_orange_yellow_large	0.999999999999995
true_fish_orange_yellow_medium	0.999999999999995
true_fish_orange_yellow_small	0.999999999999995

It seems like the model gets most of the classifications correct but fails on blue bodied fish. Next let's examine the posterior over rules.

Posterior on Rules

Table 2: Posterior on Rules (Rational Rules with Ground Truth Labels)

logProb	rule
-0.0720332	((critter==2)OR(body_color==0))
-3.8397023	((critter==2)OR((body_color==0)AND(body_color==0)))

logProb	rule
-4.1997051	((critter==2)OR((body_color==0)OR(critter==2)))
-4.3050656	(((critter==2)AND(critter==2))OR(body_color==0))
-5.2983174	((critter==2)OR((body_color==0)OR(((critter==1)AND(critter==0))AND(critter==0))))
-5.8091430	((critter==2)OR((body_color==0)OR((((((critter==1)AND(secondary_color==1))AND(secondary_color==
-6.5022902	((critter==2)OR((body_color==0)OR((((critter==0)AND(secondary_color==0))AND(body_color==2))AN
-6.5022902	((critter==2)OR((body_color==0)OR((((critter==1)AND(size==0))AND(critter==0))OR((secondary_colo
-6.5022902	((critter==2)OR((body_color==0)OR((((secondary_color==2)AND(body_color==1))AND(size==2))AND(
-6.5022902	((critter==2)OR((body_color==0)OR((secondary_color==1)AND(critter==2))))

Examining the results of this experiment, we see that the rule with the greatest probability, by far is ((Body Color == 0) OR (Critter == 2)). This is ((Body Color == BLUE) OR (Critter == FISH)). It seems like the model is getting close to the true answer, but is still incorrect.

(A == 1 XOR B == 1) can be expressed as (A == 1 && not B == 1) OR (Not A == 1 && B == 1). This short definition, requires the inclusion of a negation operand, which our CFG lacks currently. Our CFG can still express XOR, but it takes a lot longer of a formula: (A == 1 && (B == 0 OR B == 2)) OR ((A == 0 OR A == 2) && B == 1).

The longer the formula, the harder it is to generate, so our CFG may never be generating the proper rule to be applying as a proposal, when running MCMC to develop a distribution over possible rules. We might want to add negation to our CFG.

Teachers