The Second Data Analysis for the User Study in STFT-LDA

anonymous

4/23/2019

This document presents the data analysis performed in the evaluation section of the submitted paper.

Library requirements

```
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(ggplot2)
library(purrr)
library(stringr)
library(tidyr)
library(boot)
```

Data preparation

Our experiment tracks additional information which we drop in this analysis for the sake of anonymity. These lines are used internally to convert the full input files to the anonymized dataset we provide with the submission.

```
file_list_v2 = lapply(Sys.glob("data_2/*.csv"), read.csv)
study_data_v2 = do.call(rbind, file_list_v2) %>%
  filter(!is.na(corr_res)) %>%
  mutate(is_heamap = str_detect(image1,"^signals")) %>%
  mutate(vis_type = ifelse(str_detect(image1,"^signals"), "heatmap", "topic")) %>%
  mutate(test_result = ifelse(corr_res == user_res, "correct", "incorrect")) %>%
  mutate(accurate = corr_res == user_res) %>% select(-user_name);
write.csv(x=study_data_v2, file="user_study_data.csv")
```

Instead, we load the file that was generated with the above code:

```
study_data = read.csv("user_study_data_2.csv")
```

Our experiment includes a trivial test we use to rule out participants that clearly were not attempting to answer the question correctly, by including a judgment task with a perfectly identical target image; this is encoded by level `0` in the factor `rule`. Our experiment was designed to drop the data from participants who respond incorrectly to any of the trivial task; however, no participants were caught in this check:

```
trivial_data = study_data %>% filter(rule == 0)
all(trivial_data$accurate)
```

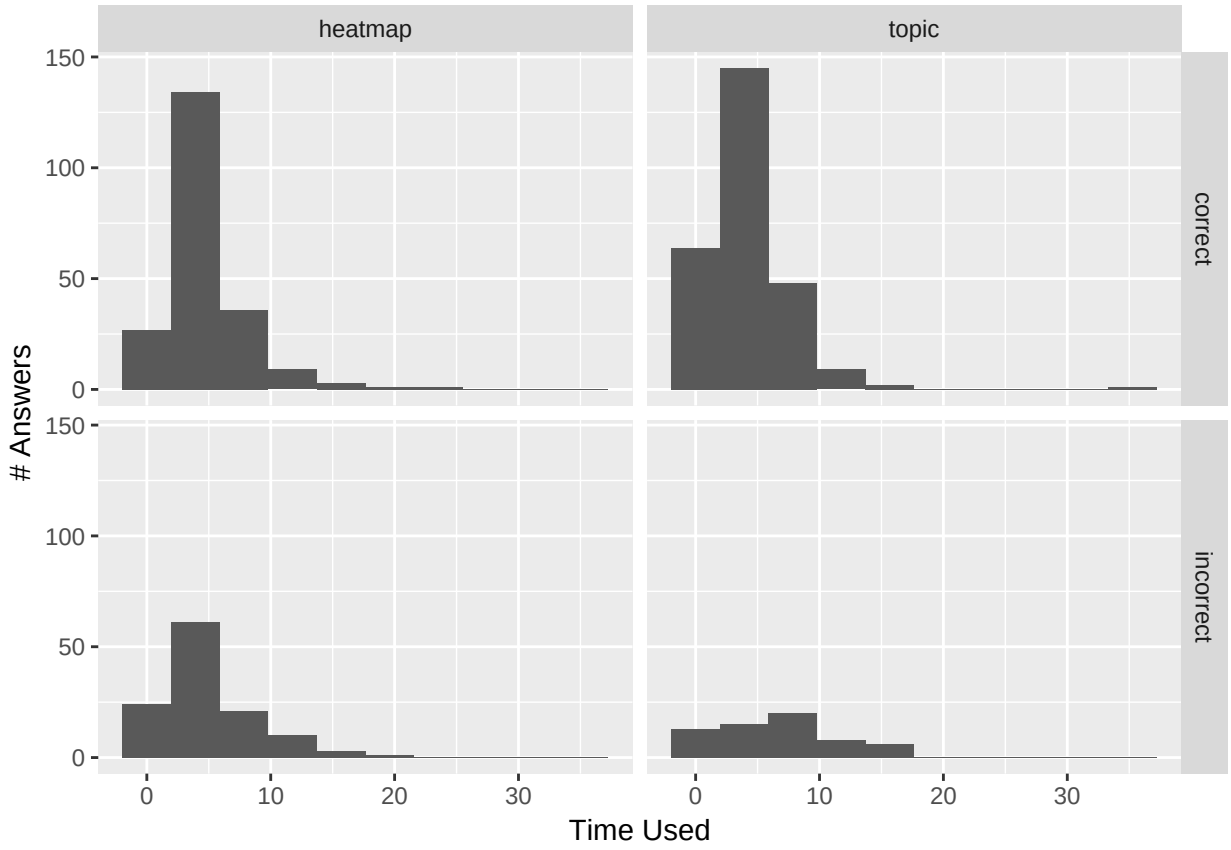
```
## [1] FALSE
```

Nevertheless, because this task is trivial, we drop its results from the analysis:

```
study_data = study_data %>% filter(rule != 0)
```

The average time per question spent on the user study was 4.6372961 seconds.

```
ggplot(study_data) +
  aes(x = time_used) +
  geom_histogram(bins=10) +
  facet_grid(test_result ~ vis_type) +
  xlab("Time Used") + ylab("# Answers")
```



We can test directly whether participants took longer to answer the tasks using topic or signal views:

```
t.test(time_used ~ is_heatmap, study_data)
```

```
##
##  Welch Two Sample t-test
##
## data:  time_used by is_heatmap
## t = -0.079454, df = 654.99, p-value = 0.9367
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.560106  0.516541
```

```
## sample estimates:
## mean in group FALSE  mean in group TRUE
##            4.626405            4.648187
```

In other words, there's no statistical evidence that participants took longer to answer the tasks using either the topic or the signal-based views.

```
contingency_table = study_data %>%
  group_by(test_result, vis_type) %>%
  summarise(count=n()) %>% mutate(entry = str_c(test_result, ", ", vis_type))
```

Because our question of interest is whether people perform better using the topic-model visualization rather than the signal-based visualization, our hypothesis is simply:

Hypothesis: participants will incur fewer errors on the topic-model visualization compared to signal-based visualization.

The null hypothesis is then that the proportion of correct answers is the same for signal and topic visualizations, and we can test this hypothesis using the one-sided version of Fisher's exact test:

```
test.result <- fisher.test(matrix(contingency_table$count, nrow=2), alternative="less")
test.result
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  matrix(contingency_table$count, nrow = 2)
## p-value = 2.977e-07
## alternative hypothesis: true odds ratio is less than 1
## 95 percent confidence interval:
##  0.0000000 0.5545258
## sample estimates:
## odds ratio
##  0.4058255
```

As we can see, we can reject the null hypothesis, since $p = 3.36 \times 10^{-5}$, and that at the 95% confidence interval, the odds ratio for the test is 0.6. Roughly speaking, we can say that the test indicates 95% confidence that participants are $((1/0.6) - 1) = 66\%$ better at the task using the topic-based visualization [^1].

Additional visualizations and analyses

The result is robust under the bootstrap (using 1000 samples for efficiency, but you're welcome to crank it up to 10000 and verify that the result doesn't change):

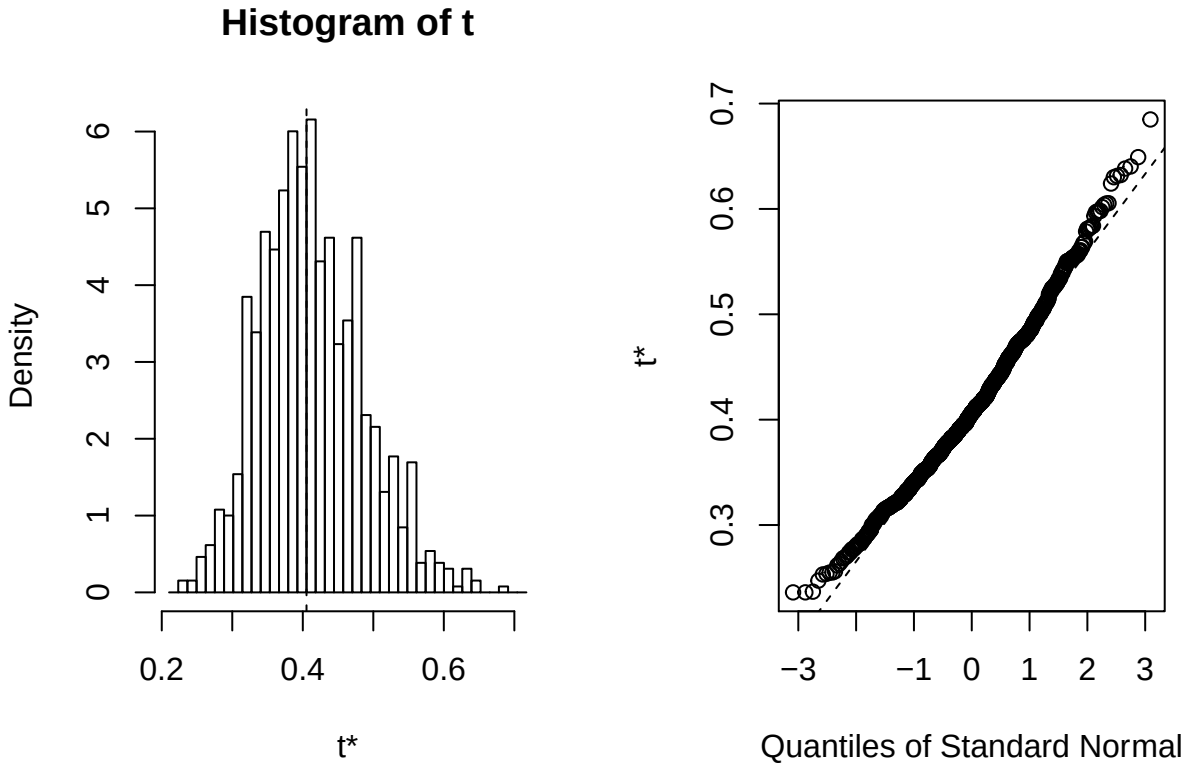
```
vis_odds_ratio = function(data, indices) {
  data = data[indices,] %>% group_by(vis_type, accurate) %>% summarize(count=n())

  incorrect_signal   = as.numeric(data[1,3])
  correct_signal     = as.numeric(data[2,3])
  incorrect_topic    = as.numeric(data[3,3])
  correct_topic      = as.numeric(data[4,3])

  topic_odds <- incorrect_topic / correct_topic
  signal_odds <- incorrect_signal / correct_signal

  return(topic_odds/signal_odds)
```

```
}
r<-boot(data= study_data, statistic = vis_odds_ratio, R=1000)
plot(r)
```



The result is also robust under different analyses (now using differences in the mean accuracy, and again bootstrapping), showing a 95% confidence interval value between 0.10 and 0.15 (that is: the difference in percentages is very likely to be above 10%):

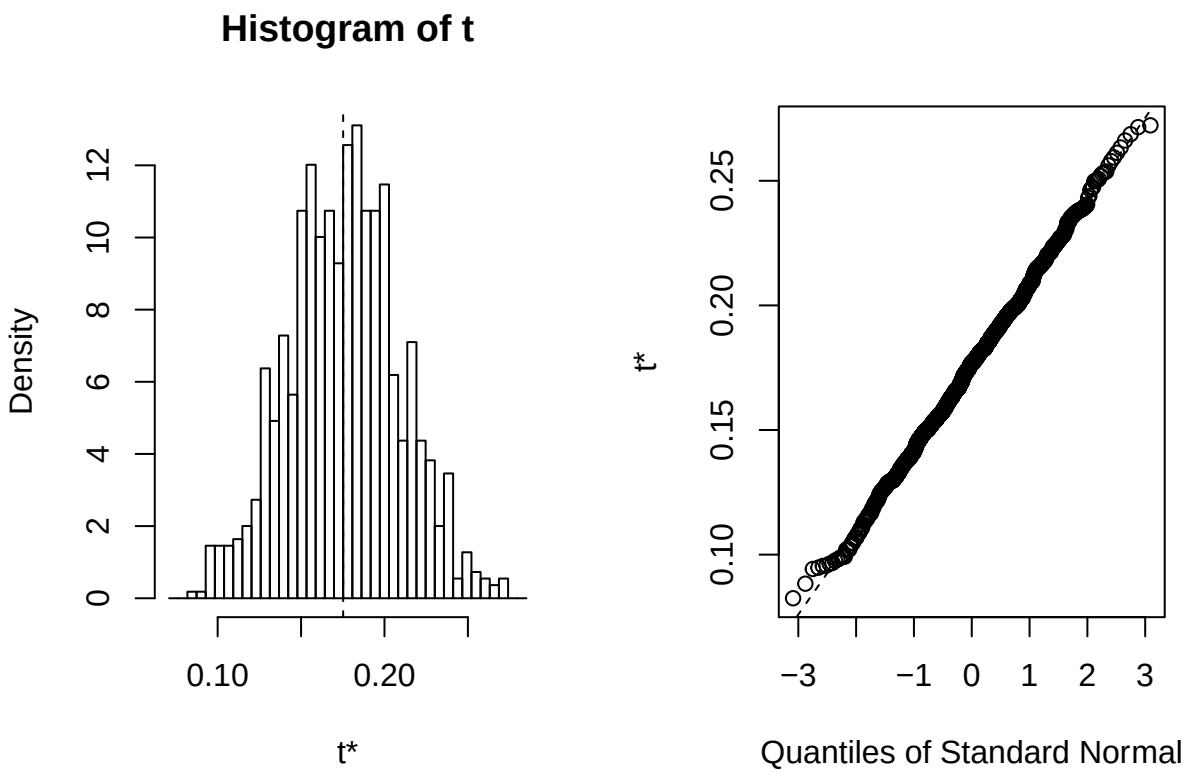
```
vis_mean_diff = function(data, indices) {
  data = data[indices,] %>% group_by(vis_type, accurate) %>% summarize(count=n())

  incorrect_signal = as.numeric(data[1,3])
  correct_signal   = as.numeric(data[2,3])
  incorrect_topic  = as.numeric(data[3,3])
  correct_topic    = as.numeric(data[4,3])

  topic_acc <- correct_topic / (correct_topic + incorrect_topic)
  signal_acc <- correct_signal / (correct_signal + incorrect_signal)

  return(topic_acc - signal_acc)
}

r<-boot(data= study_data, statistic = vis_mean_diff, R=1000)
plot(r)
```



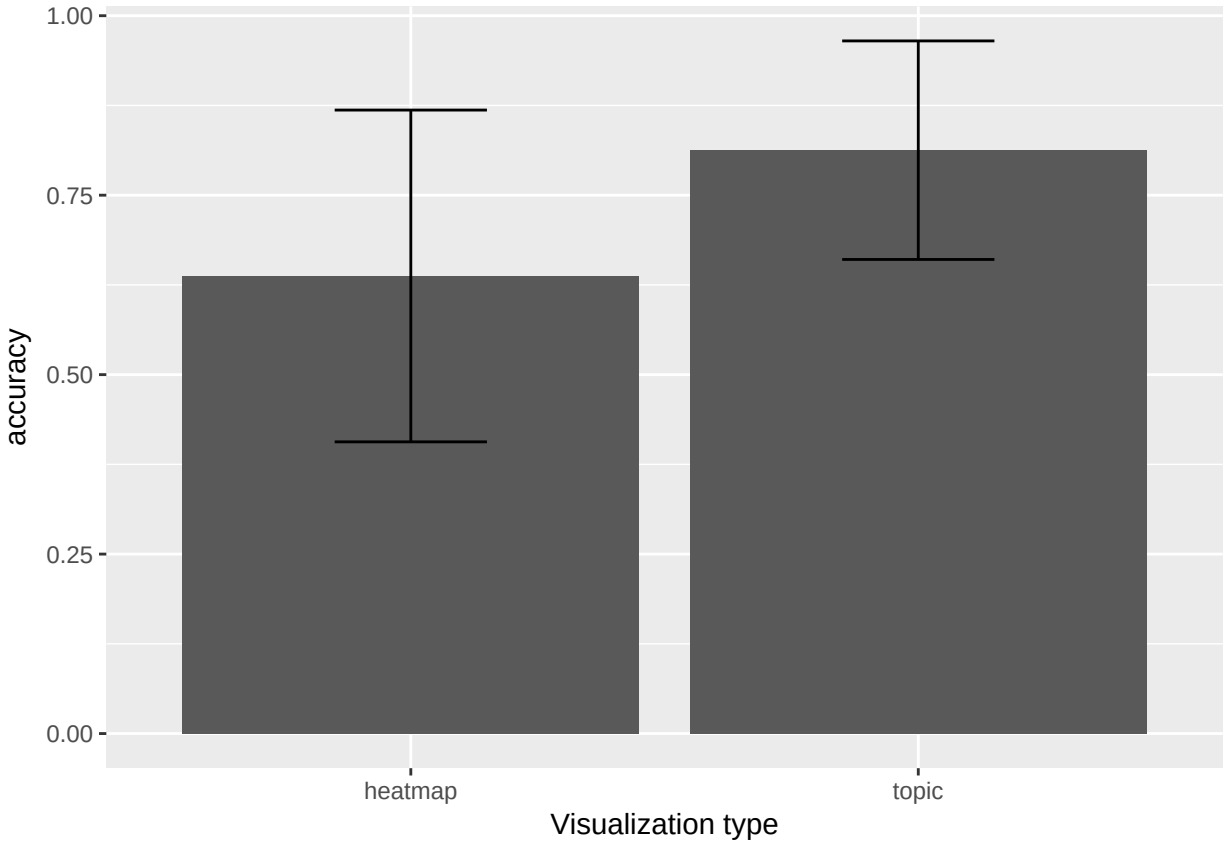
Here is the accuracy plot for the average accuracies over the entire dataset (using the binomial approximation for standard deviations):

```
# These indices change (compared to above) because the order is determined
# by the surrounding columns, which are different between the bootstrap samples
# and the full dataset

incorrect_signal   = as.numeric(contingency_table[3,3])
correct_signal     = as.numeric(contingency_table[1,3])
incorrect_topic    = as.numeric(contingency_table[4,3])
correct_topic      = as.numeric(contingency_table[2,3])

summary_results = data.frame(
  n = c((correct_signal + incorrect_signal),
        (correct_topic + incorrect_topic)),
  accuracy=c(correct_signal / (correct_signal + incorrect_signal),
             correct_topic / (correct_topic + incorrect_topic)),
  vis_type = c("heatmap", "topic")) %>% mutate(stdev = accuracy * (1 - accuracy))

ggplot(summary_results) +
  aes(x=vis_type, y=accuracy) +
  geom_col() +
  geom_errorbar(aes(ymin=accuracy-stdev, ymax=accuracy+stdev), width=0.3) +
  xlab("Visualization type")
```


1.00
0.75
0.50
0.25
0.00
accuracy
heatmap
topic
Visualization type