MD1 - CDL data analysis

February 21, 2021

Preregistration

Hypothesis

ManyDogs 1 aims to investigate how dogs perceive human pointing gestures. Our specific question is: Do dogs understand and act on human pointing as though it is a social communicative gesture? While studies of dogs following pointing are numerous, there is ongoing debate in the field of Canine Science about whether dogs perceive pointing gestures as peremptory and follow them much like they would a verbal command, or as socially communicative cues. We will address this question and investigate whether dogs understand and act on pointing as a social communicative gesture by comparing a condition in which pointing is prefaced by ostensive cues (intentional gaze, name calling verbal cues) to a condition in which it is not (gaze avoidance, no verbal cues).

Condition predictions

i) We predict that if dogs perceive pointing gestures as commands, as a group they would follow the gesture above chance in both experimental conditions. Moreover, we would expect there to be no significant differences between point-following by dogs in the two experimental conditions.
ii) We predict that if dogs perceive pointing gestures as socially communicative cues, they would perform significantly better in the ostensive condition than in the non-ostensive condition.
iii) We predict that in the odor control task that dogs will not differ significantly from the hypothetical chance level of 0.5.

Additional predictors about demographics, research task experience, training history, along with site-specific variables will be evaluated as well, but we do not have specific predictions about how they may influence performance in this task.

Data analysis

Read data

Aggregating data

One-sample t-test to compare against chance level

Preregistration: T-tests: Proportion of correct choices different from chance level (0.5) per condition (ostensive, non-ostensive, odor control). We will conduct this analysis also separately for each lab. For this analysis, performance is aggregated per individual.

```r
tt.ost <- t.test(agg.data$mean.resp[agg.data$condition == "ost"], mu = 0.5, alternative = "two.sided")
tt.non <- t.test(agg.data$mean.resp[agg.data$condition == "non"], mu = 0.5, alternative = "two.sided")
tt.oc <- t.test(agg.data$mean.resp[agg.data$condition == "odour"], mu = 0.5, alternative = "two.sided")

tt.ost
```

```
##
##  One Sample t-test
##
## data:  agg.data$mean.resp[agg.data$condition == "ost"]
## t = 4.4063, df = 60, p-value = 4.414e-05
## alternative hypothesis: true mean is not equal to 0.5
## 95 percent confidence interval:
##  0.5548270 0.6459927
## sample estimates:
## mean of x
## 0.6004098
```

```r
tt.non
```

```
##
##  One Sample t-test
##
## data:  agg.data$mean.resp[agg.data$condition == "non"]
## t = 1.4735, df = 60, p-value = 0.1458
## alternative hypothesis: true mean is not equal to 0.5
## 95 percent confidence interval:
##  0.4897447 0.5676323
## sample estimates:
## mean of x
## 0.5286885
```

```r
tt.oc
```

```
##
##  One Sample t-test
##
## data:  agg.data$mean.resp[agg.data$condition == "odour"]
## t = -1.4543, df = 60, p-value = 0.1511
## alternative hypothesis: true mean is not equal to 0.5
## 95 percent confidence interval:
##  0.4123802 0.5138494
## sample estimates:
## mean of x
## 0.4631148
```

The dogs (N = 61) performed significantly better than expected by chance in the ostensive condition (mean = 0.6, 95% CI [0.55, 0.65], t(60)=4.41, p=0) but not in the non-ostensive condition (mean = 0.53, 95% CI [0.49, 0.57], t(60)=1.47, p=0.146) or the odor control condition (mean = 0.46, 95% CI [0.41, 0.51], t(60)=-1.45, p=0.151).

Generalised linear mixed model

Preregistration: Generalized linear mixed model (GLMM) with binomial error structure and logit link function (only including the ostensive and non-ostensive condition): Correct choice ~ condition + order_condition + trial_within_condition + sex + age + dog_training_experience + (condition + trial_within_condition + | Subject ID) + (condition+ order_condition + trial_within_condition + sex + age + dog_training_experience | Lab ID), family = binomial

All covariates will be z-transformed. The random slope components of the factors will be centered (to ensure that the results are not conditional on the choice of the reference category). In order to rule out collinearity issues, we will determine Variance Inflation Factors using the function vif (R package car). As inference criterion, we will use p-values below .05 (two-tailed) for one sample t-tests. For the GLMMs, likelihood ratio tests (R function drop1 with argument 'test' set to "Chisq") with p-values smaller than .05 will be used as criterion to make inferences about fixed effects.

```
# centering variables for modeling

model.data <- prereg.data  %>%
  filter(condition == "ost" | condition == "non") %>%
  mutate(
    z.age = as.numeric(scale(age, scale = T, center = T)),
    sex.c = as.numeric(scale(as.numeric(as.factor(sex)), scale = F, center = T)),
    condition.c = as.numeric(scale(as.numeric(as.factor(condition)), scale = F, center = T)),
    condition_order.c = as.numeric(scale(as.numeric(as.factor(condition_order)), scale = F, center = T)
    z.trial_num = as.numeric(scale(trial_num, scale = T, center = T)),
    z.training_experience = as.numeric(scale(dog_training_experience, scale = T, center = T))
  )

#view(dfSummary(model.data))
```

```
contr <- glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 10000000))

mm1 <- glmer(response ~ condition + condition_order + z.trial_num + sex + z.age + z.training_experience
  family = binomial, data = model.data, control = contr
)
```

```
## boundary (singular) fit: see ?isSingular
```

Likelihood ratio test

```
drop1.mm1<-drop1(mm1, test = "Chisq", control = contr)
```

```
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
```

```
round(drop1.mm1, 3)
```

```
## Single term deletions
```

```
##
## Model:
## response ~ condition + condition_order + z.trial_num + sex +
##     z.age + z.training_experience + (condition.c + z.trial_num |
##     subject_ID)
##                         npar    AIC   LRT Pr(Chi)
## <none>                       1352.1
## condition                  1 1355.2 5.107   0.024 *
## condition_order            1 1350.3 0.185   0.667
## z.trial_num                1 1352.4 2.322   0.128
## sex                        1 1350.5 0.371   0.543
## z.age                      1 1350.2 0.033   0.855
## z.training_experience      1 1352.1 1.957   0.162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The dogs were significantly more likely to choose the baited cup in the ostensive condition compared to the non-ostensive condition ($\chi^2(1)$=5.107, p = 0.024). The control predictors order of condition, trial number (within condition), sex, age, or the C-BARQ trainability score had no significant effect on dogs' choices.

Estimates of the fixed effects

```
round(summary(mm1)$coefficients, 3)
```

```
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)                0.122      0.128   0.955    0.340
## conditionost               0.298      0.132   2.253    0.024
## condition_orderost         0.057      0.133   0.431    0.667
## z.trial_num               -0.100      0.066  -1.519    0.129
## sexM                      -0.084      0.137  -0.611    0.541
## z.age                     -0.012      0.067  -0.183    0.855
## z.training_experience      0.095      0.067   1.405    0.160
```

Random effects

```
summary(mm1)$varcor
```

```
##  Groups     Name        Std.Dev. Corr
##  subject_ID (Intercept) 0.099209
##             condition.c 0.158625  1.000
##             z.trial_num 0.063051 -1.000 -1.000
```

Calculating confidence intervals with 1000 bootstraps

```
#perform bootstraps for all predictors
mm1.boot.ci <- boot.glmm.pred(model.res = mm1, excl.warnings = T, nboots = 1000, para = T, n.cores="all
```

```
round(mm1.boot.ci$ci.estimates, 3)
```

```
##                           orig  X2.5. X97.5.
## (Intercept)              0.122 -0.134  0.372
## conditionost             0.298  0.034  0.575
## condition_orderost     0.057 -0.216  0.324
## z.trial_num           -0.100 -0.227  0.032
## sexM                  -0.084 -0.361  0.184
## z.age                 -0.012 -0.140  0.121
## z.training_experience  0.095 -0.034  0.236
```

Check assumptions

```
# check for colliniarity
xx <- lm(response ~ condition + condition_order + z.trial_num + sex + z.age,
  data = model.data
)
vif(xx)
```

```
##       condition condition_order     z.trial_num             sex           z.age
##        1.000000        1.000291        1.000000        1.021891        1.021822
```

Collinearity was no issue (maximum variance inflation factor: 1.02)

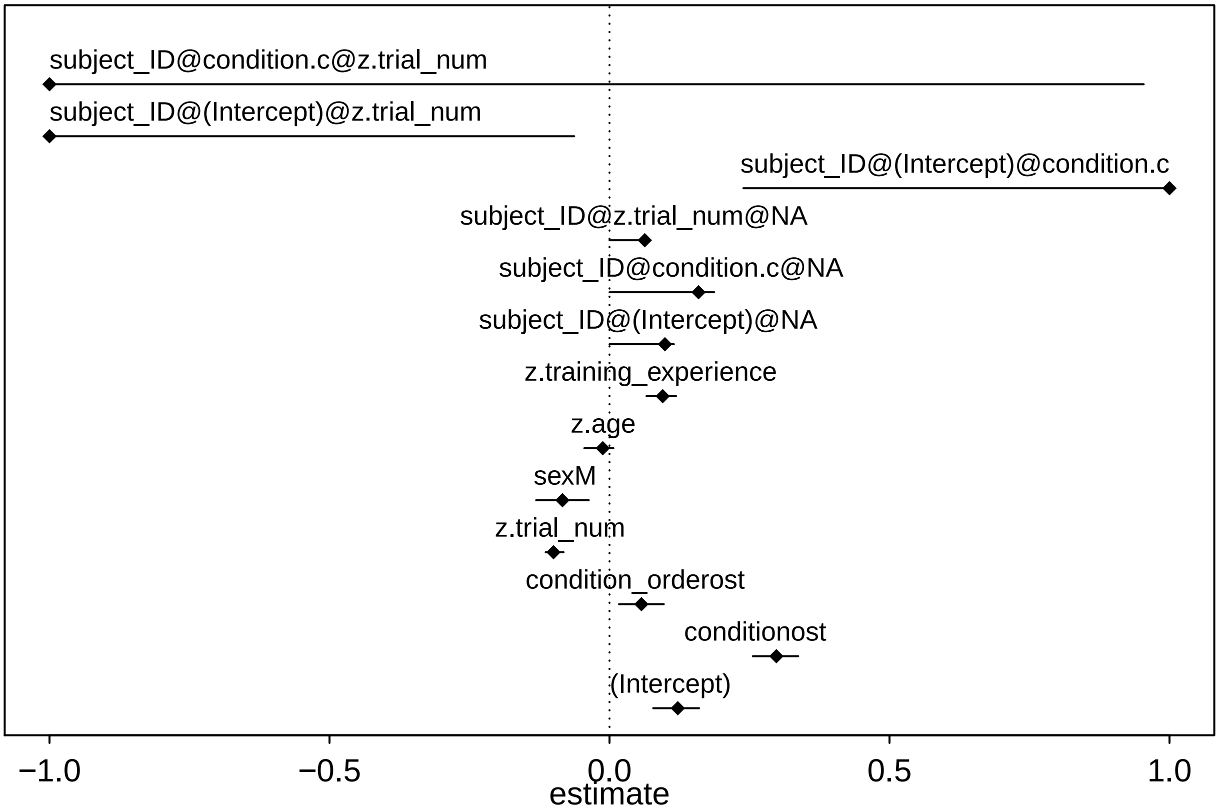
Model stability

One subject at a time excluded to assess the impact of outliers.

```
m.stab.b <- glmm.model.stab(model.res = mm1, use = c("subject_ID"))
m.stab.b$detailed$warnings
xx <- as.data.frame(round(m.stab.b$summary[, -1], 3))

png("graphs/mm1_stability_plot.png")
m.stab.plot(round(m.stab.b$summary[, -1], 3))
dev.off()
```

```
m.stab.plot(round(m.stab.b$summary[, -1], 3))
```



The model appeared to be stable with respect to the fixed effects (see mm1_stability_plot).

Calculate CIs for plot

```
# Model with all predictor variables centered except for condition:
mm1.CI.con <- glmer(response ~ condition + condition_order.c + z.trial_num + sex.c + z.age + z.training
  family = binomial, data = model.data, control = contr
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
pred.con.ci=boot.glmm.pred(model.res=mm1.CI.con,  level=0.95, use="condition", n.cores="all-1", para=T)

pred.con.ci$ci.predicted

write.csv(pred.con.ci$ci.predicted, file = "data/mm1_predicted_ci_for_conditionMD1_Vienna.csv")
```

```
#save.image("md1_vienna_glmm_workspace.RData")
```

all data (including data that was collected prior to preregistration)

Read data

Aggregating data

One-sample t-test to compare against chance level

```r
tt.ost.all <- t.test(agg.data.all$mean.resp[agg.data.all$condition == "ost"], mu = 0.5, alternative = "
tt.non.all <- t.test(agg.data.all$mean.resp[agg.data.all$condition == "non"], mu = 0.5, alternative = "
tt.oc.all <- t.test(agg.data.all$mean.resp[agg.data.all$condition == "odour"], mu = 0.5, alternative =

tt.ost.all
```

```
##
##  One Sample t-test
##
## data:  agg.data.all$mean.resp[agg.data.all$condition == "ost"]
## t = 6.1177, df = 90, p-value = 2.417e-08
## alternative hypothesis: true mean is not equal to 0.5
## 95 percent confidence interval:
##  0.5779141 0.6528551
## sample estimates:
## mean of x
## 0.6153846
```

```r
tt.non.all
```

```
##
##  One Sample t-test
##
## data:  agg.data.all$mean.resp[agg.data.all$condition == "non"]
## t = 2.985, df = 90, p-value = 0.003651
## alternative hypothesis: true mean is not equal to 0.5
## 95 percent confidence interval:
##  0.5169979 0.5846504
## sample estimates:
## mean of x
## 0.5508242
```

```r
tt.oc.all
```

```
##
##  One Sample t-test
##
## data:  agg.data.all$mean.resp[agg.data.all$condition == "odour"]
## t = -1.9841, df = 90, p-value = 0.05029
## alternative hypothesis: true mean is not equal to 0.5
## 95 percent confidence interval:
##  0.4175288 0.5000536
## sample estimates:
## mean of x
## 0.4587912
```

The dogs (N = 91) performed significantly better than expected by chance in the ostensive condition (mean = 0.62, 95% CI [0.58, 0.65], t(90)=6.12, p=0) but not in the non-ostensive condition (mean = 0.55, 95% CI [0.52, 0.58], t(90)=2.98, p=0.004) or the odor control condition (mean = 0.46, 95% CI [0.42, 0.5], t(90)=-1.98, p=0.05).

Generalised linear mixed model

```
# centering variables for modeling

model.data.all <- all.data %>%
  filter(condition == "ost" | condition == "non") %>%
  mutate(
    z.age = as.numeric(scale(age, scale = T, center = T)),
    sex.c = as.numeric(scale(as.numeric(as.factor(sex)), scale = F, center = T)),
    condition.c = as.numeric(scale(as.numeric(as.factor(condition)), scale = F, center = T)),
    condition_order.c = as.numeric(scale(as.numeric(as.factor(condition_order)), scale = F, center = T)
    z.trial_num = as.numeric(scale(trial_num, scale = T, center = T)),
    z.training_experience = as.numeric(scale(dog_training_experience, scale = T, center = T))
  )

#view(dfSummary(model.data.all))
```

```
contr <- glmerControl(optimizer = "bobyqa", optCtrl = list(maxfun = 10000000))

mm2 <- glmer(response ~ condition + condition_order + z.trial_num + sex + z.age + z.training_experience
  family = binomial, data = model.data.all, control = contr
)
```

```
## boundary (singular) fit: see ?isSingular
```

Likelihood ratio test

```
drop1.mm2<-drop1(mm2, test = "Chisq", control = contr)
```

```
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
```

```
round(drop1.mm2, 3)
```

```
## Single term deletions
##
## Model:
## response ~ condition + condition_order + z.trial_num + sex +
##     z.age + z.training_experience + (condition.c + z.trial_num |
##     subject_ID)
##                        npar    AIC   LRT Pr(Chi)
## <none>                      1992.0
## condition                 1 1996.4 6.358   0.012 *
## condition_order           1 1990.5 0.433   0.510
## z.trial_num               1 1991.1 1.080   0.299
## sex                       1 1991.1 1.025   0.311
## z.age                       1 1990.1 0.029   0.866
## z.training_experience       1 1991.7 1.602   0.206
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The dogs were significantly more likely to choose the baited cup in the ostensive condition compared to the non-ostensive condition ($\chi^2(1)$=6.358, p = 0.012). The control predictors order of condition, trial number (within condition), sex, age, or the C-BARQ trainability score had no significant effect on dogs' choices.

Estimates of the fixed effects

```
round(summary(mm2)$coefficients, 3)
```

```
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)                0.292      0.105   2.774    0.006
## conditionost               0.276      0.110   2.512    0.012
## condition_orderost        -0.074      0.112  -0.657    0.511
## z.trial_num               -0.058      0.056  -1.036    0.300
## sexM                      -0.116      0.114  -1.011    0.312
## z.age                      0.010      0.056   0.169    0.866
## z.training_experience      0.072      0.057   1.270    0.204
```

Random effects

```
summary(mm2)$varcor
```

```
##  Groups     Name        Std.Dev. Corr
##  subject_ID (Intercept) 0.16111
##             condition.c 0.19885   1.000
##             z.trial_num 0.12375  -1.000 -1.000
```

Calculating confidence intervals with 1000 bootstraps

```
# perform bootstraps for all predictors
boot.res.mm2 <- boot.glmm.pred(model.res = mm2, excl.warnings = T, nboots = 1000, para = T, level = 0.95
```

```
round(boot.res.mm2$ci.estimates, 3)
```

```
##                           orig  X2.5. X97.5.
## (Intercept)              0.292  0.088  0.508
## conditionost             0.276  0.067  0.481
## condition_orderost      -0.074 -0.296  0.144
## z.trial_num             -0.058 -0.170  0.058
## sexM                    -0.116 -0.343  0.120
## z.age                    0.010 -0.108  0.122
## z.training_experience    0.072 -0.042  0.185
```

Check assumptions

```
# check for colliniarity
xx2 <- lm(response ~ condition + condition_order + z.trial_num + sex + z.age,
  data = model.data
)
vif(xx2)
```

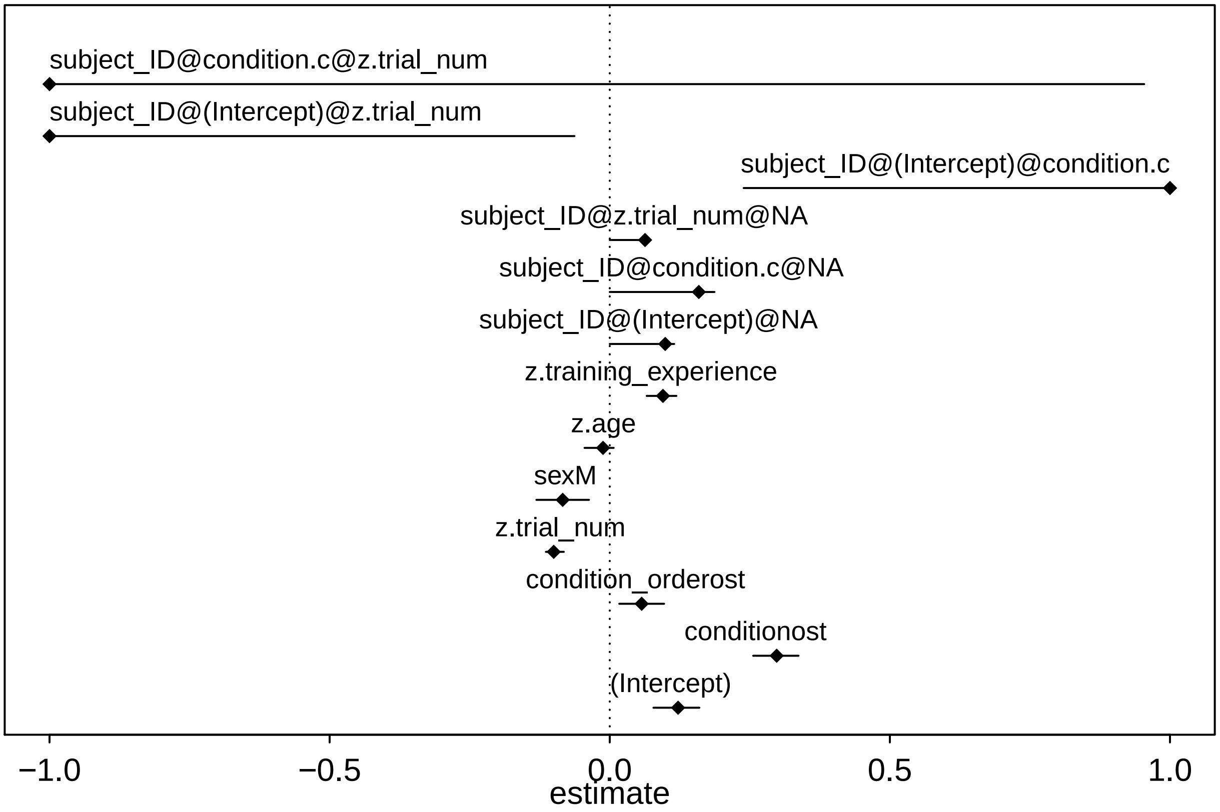
```
##       condition condition_order     z.trial_num             sex           z.age
##        1.000000        1.000291        1.000000        1.021891        1.021822
```

Model stability

One subject at a time excluded to assess the impact of outliers.

```
m.stab.b.mm2 <- glmm.model.stab(model.res = mm1, use = c("subject_ID"))
m.stab.b.mm2$detailed$warnings
xx2 <- as.data.frame(round(m.stab.b$summary[, -1], 3))
png("graphs/mm2_stability_plot.png")
m.stab.plot(round(m.stab.b.mm2$summary[, -1], 3))
dev.off()
```

```
m.stab.plot(round(m.stab.b.mm2$summary[, -1], 3))
```



Calculate CIs for plot

```
# Model with all predictor variables centered except for condition:
mm2.CI.con <- glmer(response ~ condition + condition_order.c + z.trial_num + sex.c + z.age + z.training
  family = binomial, data = model.data, control = contr
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
mm2.pred.con.ci=boot.glmm.pred(model.res=mm2.CI.con,  level=0.95, use="condition", n.cores="all-1", para

mm2.pred.con.ci$ci.predicted

write.csv(mm2.pred.con.ci$ci.predicted, file = "data/mm2_predicted_ci_for_condition_MD1_Vienna.csv")
```

```
#save.image("md1_vienna_glmm_workspace.RData")
```