

Noisy prior model of SPE by learners of monolingual and contact Spanish

```
#DO NOT USE plyr library. It will interfere with group by and summarise operations in code chunks 'prior'  
library(tidyverse)
```

This is a revised model of how learners calculate the posterior $P(\text{pronoun}|\text{reference})$. Unlike the baseline optimal model, this model assumes a NOISY learner who inaccurately calculates the posterior by occasionally omitting the prior $P(\text{pronoun})$. Like the baseline optimal model, this model still assumes that the learner accurately learns both the prior $P(\text{pronoun})$ and the likelihood $P(\text{reference}|\text{pronoun})$ from the input.

We will look at the saem two populations.

The code below loads Mexican and Villa 21 data, coded for form, reference, and other attributes. It only needs to be run if the optimal model has not already been run in the same R session.

Assumptions about the input

Assumptions about what constitutes the input are the same as in the optimal model.

```
#put Mexico City and Buenos Aires together into a single aduchi dataset and format its variables accordingly  
aduchi <- read_csv("aduchi.csv")
```

```
## Parsed with column specification:  
## cols(  
##   file = col_character(),  
##   participant = col_character(),  
##   dyad = col_character(),  
##   SES = col_character(),  
##   stem = col_character(),  
##   nulloverrt = col_character(),  
##   animacy = col_character(),  
##   person = col_character(),  
##   number = col_character(),  
##   tma = col_character(),  
##   reference_utt = col_character(),  
##   reference_turn = col_character(),  
##   yr = col_double(),  
##   mo = col_double(),  
##   community = col_character(),  
##   group = col_character(),  
##   filetype = col_character()  
## )
```

```

aduchi <- aduchi %>%
  select(community, file, participant, dyad, SES, stem, animacy, nullover, reference_turn, person, number)
  filter(
    animacy == 1 &
    nullover %in% c(0, 1) &
    reference_turn %in% c("same", "switch") &
    #exclude usted(es), following Shin 2016
    person %in% c("1", "2", "3") &
    number %in% c("s", "p", "S", "P") &
    tma %in% c("cond", "fut=", "pas", "pres", "PRES", "pret", "PRET", "sub&pres")
  ) %>% mutate(
    #make the outcome variable numeric
    SPE = as.numeric(paste(nullover)),
    #rename the reference variable
    ref = reference_turn,
    #create personnum factor following Shin 2016
    personnum = as.factor(paste0(person, tolower(number))),
    #create TMA factor following Shin 2016
    tma = recode(tma, "pres"="present", "PRES"="present", "pret"="preterite", "PRET"="preterite", "pas"="")
  ) %>% filter(
    #filter out a coding mistake
    personnum != "2p"
  ) %>% mutate(
    #relevel personnum and tma
    personnum = fct_relevel(personnum, "1s", "2s", "3s", "1p", "3p"),
    tma = fct_relevel(tma, "other", "imperfect", "preterite", "present")
  ) %>% mutate(
    #Add child ages and age groups.
    child_age_mo = 12*yr + mo,
    child_age_yr = child_age_mo / 12,
    #Note: use 2 age groups - must be split at 56 to prevent missing values
    child_age_group = cut(child_age_mo, breaks = c(-Inf, 56, Inf), labels=c("younger", "older"), right = FALSE)
  ) %>% mutate(
    #split primary dataset into input and output
    inputoutput = ifelse(participant == "CHI", "output", "input")
  )

```

```

## # A tibble: 2 x 3
##   speech_type tokens proportion
##   <chr>      <int>      <dbl>
## 1 ps_input    2357      0.611
## 2 rps_input   1498      0.389

```

The model

This model is a mixture between:

- the optimal model, which uses both likelihood and prior to accurately calculate the posterior:
 $P_{OPTIMAL}(overt|reference) \propto P(reference|overt) \times P(overt)$
- a model that omits the prior: $P_{NOISY}(overt|reference) \propto P(reference|overt)$

The degree of omission is determined by how much the optimal and noisy models each contributes to the posterior, as controlled by the parameter $0 \leq \beta_{PRIOR} \leq 1$. The larger β_{PRIOR} , the more the learner uses the prior.:

$$P(overt|reference) = \beta_{PRIOR} * P_{OPTIMAL}(overt|reference) + (1 - \beta_{PRIOR}) * P_{NOISY}$$

Priors

As with the optimal model, priors are learned accurately from the input that the child receives. For Villa21, more Rioplatense input will result in a lower prior and more Paraguayan input will result in a higher prior. The last column is the prior probability of null (SPE=0) and overt (SPE=1) in each community.

```
#Form a tibble with the prior of each form in each community
priors <- aduchi %>%
  filter(inputoutput == "input") %>%
  group_by(community, SPE) %>%
  dplyr::summarise(n = n()) %>%
  mutate(
    freq = n/sum(n)
  )
print(priors)
```

```
## # A tibble: 4 x 4
## # Groups:   community [2]
##   community    SPE      n freq
##   <chr>      <dbl> <int> <dbl>
## 1 Buenos_Aires    0   3352 0.870
## 2 Buenos_Aires    1    503 0.130
## 3 Mexico_City     0   2589 0.890
## 4 Mexico_City     1    319 0.110
```

Likelihoods

As with the optimal model, likelihoods are also learned accurately from the input. For Villa21, more Rioplatense input will not change the likelihoods, since both dialects condition pronoun realization on reference *to the same degree*. The last column is the likelihood of each reference context, given an overt or null pronoun - calculated for each community.

```
ref_likelihoods <- aduchi %>%
  filter(inputoutput == "input") %>%
  group_by(community, SPE, ref) %>%
  dplyr::summarise(
    n = n()
  ) %>%
  mutate(
    freq = n/sum(n)
  )
print(ref_likelihoods)
```

```
## # A tibble: 8 x 5
## # Groups:   community, SPE [4]
```

##	community	SPE	ref	n	freq
##	<chr>	<dbl>	<chr>	<int>	<dbl>
## 1	Buenos_Aires	0	same	1530	0.456
## 2	Buenos_Aires	0	switch	1822	0.544
## 3	Buenos_Aires	1	same	130	0.258
## 4	Buenos_Aires	1	switch	373	0.742
## 5	Mexico_City	0	same	1149	0.444
## 6	Mexico_City	0	switch	1440	0.556
## 7	Mexico_City	1	same	68	0.213
## 8	Mexico_City	1	switch	251	0.787

Posteriors

Posteriors are calculated through a probabilistic mix of accurate and inaccurate use of the prior and likelihoods. Spelling out the equation from above more fully:

$$P(overt|reference) = \beta_{PRIOR} \sum_{pro=\{null,overt\}} \frac{P(reference|overt) \times P(overt)}{P(reference|pro) \times P(pro)} + (1 - \beta_{PRIOR}) \sum_{pro=\{null,overt\}} \frac{P(reference|overt)}{P(reference|pro)}$$

```
noisy_prior <- function(c, r, beta) {
  #priors
  prior_o <- priors$freq[priors$community == c & priors$SPE == 1]
  prior_n <- priors$freq[priors$community == c & priors$SPE == 0]

  #likelihoods
  lik_ref_o <- ref_likelihoods$freq[ref_likelihoods$community == c & ref_likelihoods$ref == r & ref_likelihoods$SPE == 1]
  lik_ref_n <- ref_likelihoods$freq[ref_likelihoods$community == c & ref_likelihoods$ref == r & ref_likelihoods$SPE == 0]

  #optimal calculation uses both priors and likelihoods
  posterior_opt <- (lik_ref_o * prior_o) /
    (lik_ref_o * prior_o +
     lik_ref_n * prior_n)

  #non-optimal calculation uses only likelihoods
  posterior_noisy <- (lik_ref_o) /
    (lik_ref_o +
     lik_ref_n * prior_n)

  #mixing them together, with beta = weight of the model WITH a prior
  posterior <- beta * posterior_opt + (1 - beta) * posterior_noisy
}
```

Predicted versus observed proportion overt

In order to find the best possible prediction of this model, we find the value of β_{PRIOR} that minimizes mean squared error between observed and predicted % overt SPE in each context.

```
#create all levels of beta_PRIOR to test
beta_PRIOR <- tibble("beta_PRIOR" = seq(from = 0, to = 1, by = 0.01))

#calculate observed % overt in each context
```

```

observed <- aduchi %>%
  filter(inputoutput == "output") %>%
  group_by(
    community, ref
  ) %>%
  summarise(
    overt = sum(SPE),
    tokens = n(),
    observed = overt/tokens
  )

#calculate predicted % overt for each combination of context and beta_PRIOR
predictions <- observed %>%
  merge(beta_PRIOR, all = TRUE) %>%
  arrange(beta_PRIOR, community, ref) %>%
  rowwise() %>%
  mutate(
    predicted = noisy_prior(community, ref, beta_PRIOR)
  )

#compare observed to predicted % overt for each value of beta_PRIOR
fit <- predictions %>% ungroup() %>%
  mutate(
    error = observed - predicted
  ) %>%
  group_by(beta_PRIOR, community) %>%
  summarise(
    MSE = mean(error*error)
  ) %>% arrange(community, MSE)

print(observed)

```

```

## # A tibble: 4 x 5
## # Groups:   community [2]
##   community    ref    overt tokens observed
##   <chr>        <chr> <dbl>  <int>   <dbl>
## 1 Buenos_Aires same     83    683   0.122
## 2 Buenos_Aires switch  128    598   0.214
## 3 Mexico_City  same     44    603   0.0730
## 4 Mexico_City  switch  115    720   0.160

```

```
print(predictions)
```

```

## Source: local data frame [404 x 7]
## Groups: <by row>
##
## # A tibble: 404 x 7
##   community    ref    overt tokens observed beta_PRIOR predicted
##   <chr>        <chr> <dbl>  <int>   <dbl>   <dbl>   <dbl>
## 1 Buenos_Aires same     83    683   0.122     0    0.394
## 2 Buenos_Aires switch  128    598   0.214     0    0.611
## 3 Mexico_City  same     44    603   0.0730     0    0.350

```

```
## 4 Mexico_City switch 115 720 0.160 0 0.614
## 5 Buenos_Aires same 83 683 0.122 0.01 0.391
## 6 Buenos_Aires switch 128 598 0.214 0.01 0.606
## 7 Mexico_City same 44 603 0.0730 0.01 0.347
## 8 Mexico_City switch 115 720 0.160 0.01 0.609
## 9 Buenos_Aires same 83 683 0.122 0.02 0.388
## 10 Buenos_Aires switch 128 598 0.214 0.02 0.602
## # ... with 394 more rows
```

```
print(fit)
```

```
## # A tibble: 202 x 3
## # Groups:   beta_PRIOR [101]
##   beta_PRIOR community      MSE
##   <dbl> <chr> <dbl>
## 1 0.89 Buenos_Aires 0.0000452
## 2 0.88 Buenos_Aires 0.0000525
## 3 0.9 Buenos_Aires 0.0000673
## 4 0.87 Buenos_Aires 0.0000892
## 5 0.91 Buenos_Aires 0.000119
## 6 0.86 Buenos_Aires 0.000155
## 7 0.92 Buenos_Aires 0.000200
## 8 0.85 Buenos_Aires 0.000251
## 9 0.93 Buenos_Aires 0.000310
## 10 0.84 Buenos_Aires 0.000376
## # ... with 192 more rows
```

Model fit: start editing here

The mean squared error of the optimal model for each community is:

```
# Pick the beta_PRIOR with the lowest MSE in each community
# fit %>% group_by(community) %>%
#   filter(MSE == min(MSE))
```

The model's likelihood given the data is calculated by first assuming that the model is true and then calculating the probability of observing the data that we have observed in our sample. If the model actually is true (or at least, a better hypothesis than other models), this probability should be high, but if it is untrue (or at least, a worse hypothesis than other models) the probability should be low.

Assuming that the optimal model is true, the probability of observing N_{overt} overt pronouns and N_{null} null pronouns in any given context is equal to $P(N_{overt}|P_{predicted}(overt)) \times P(N_{null}|P_{predicted}(null))$. For example, in same-reference contexts, the predicted probability of overt and null pronouns for Mexico City kids is $P_{predicted}(overt) = 0.056$ and $P_{predicted}(null) = 1 - 0.056 = 0.944$, respectively, and the observed count of overt and null pronouns is $N_{overt} = 44$ and $N_{null} = 603 - 44 = 559$, respectively. Thus, the probability of the observed data is

$$P(N_{overt} = 44|P_{predicted}(overt) = .056) \times P(N_{null} = 559|P_{predicted}(null) = .944) = 0.056^{44} \times 0.944^{559} = 8.504 \times 10^{-70}$$

To get the probability of the whole dataset, we then calculate $P(N_{overt}|P_{predicted}(overt)) \times P(N_{null}|P_{predicted}(null))$ in switch-reference environments and multiply the two together. Since these probabilities get very small, especially for datasets with many observations, we typically take the logarithm.

```

# loglik <- predictions %>%
#   transmute(
#     community = community,
#     ref = ref,
#     N_overt = overt,
#     N_null = tokens - overt,
#     predicted_overt = predicted,
#     predicted_null = 1 - predicted,
#     loglik = log(predicted_overt)*N_overt + log(predicted_null)*N_null
#   ) %>%
#   group_by(community) %>%
#   summarise(
#     loglik = sum(loglik)
#   )
#
# print(loglik)

```

Finally, let's visualize the observed and predicted rates of overt SPE, to see where our model is over- or under-predicting pronoun rates.

```

# obs_v_opt <- predictions %>%
#   select(-c(error, overt, tokens)) %>%
#   gather(key = "predicted_vs_observed", value = "prop_overt", observed, predicted) %>%
#   arrange(community, ref)
#
# ggplot(obs_v_opt,
#   aes(x = ref, y = prop_overt, fill = predicted_vs_observed)) +
#   facet_grid(community~.) +
#   geom_col(position = "dodge")

```

This model seems to be a better fit for the Mexico City cohort, with a lower MSE, a higher (=less negative) log likelihood and a visually better fit to the data in both same- and swith-reference contexts. Specifically, while the model under-predicts the frequency of overt SPE in both Mexico City and Villa 21, Buenos Aires, it is more severely under the mark for the latter group.