Description of model & hypothesis testing

Carcassi, Fausto   Schouwstra, Marieke   Kirby, Simon

May 24, 2019

1 Overview

This is the README of the repository `https://github.com/thelogicalgrammar/model_fitting_fifth_experiment`.

2 Hierarchical Bayesian model

The data will be analyzed with a Bayesian hierarchical model coded with PyMC3, a Python implementation of Hamiltonian Monte Carlo algorithms and related tools for Bayesian statistics. The response variable is the number of times the participant judged that the presented stimulus belongs to the category. The predictor variables are the condition, the stimuli presented as belonging to the category that the participant should infer, and how many times the participant is asked to categorize the stimuli.

There are three levels in the hierarchical model. The bottom level encodes how many times a specific participant judged a specific stimulus as belonging or not to the category. The middle level clusters the single judgments into participants. The top level encodes information about clusters of participants.

At the bottom level, the participants judgments for the stimuli are distributed binomially, and they are coded as the number of times the participant judged that the presented stimulus belongs to the category. The binomial n parameter is the number of times that the participant was asked to categorize that stimulus. The p parameter for stimulus i and participant j is the sum of the probabilities that j attributes to all the categories that contain i (*hypothesis averaging* from Xu & Tenenbaum, 2007), plus an error applied that is estimated for j. The categories are the convex set of stimuli, i.e. any set of stimuli such that if two stimuli s and p belong to the set, then any stimulus q between s and p also belongs to the set.

The posterior probability over categories for each participant is calculated at the second level of the hierarchical model. For agent j, it is a function of three things: which stimuli are presented to j as belonging to the category, js prior preference for monotonic categories, and js prior preference for categories of certain sizes. The posterior of j for a category k is proportional to the product of the likelihood, i.e. the probability of the presented stimuli v given k, and the probability that j attributes a priori to k. The probability $p(v|k)$ is calculated following the *size principle* discussed in Xu & Tenenbaum (2007) (see the full specification of the model below).

The population level parameters are the following. (1) Population level distribution(s) of preferences for monotonic hypotheses. This is two distributions, one per condition. (2) A population level distribution of preference for large or small hypotheses. Same for both conditions. (3) A population level distribution of production error. Same for both conditions.

The hyperpriors can be read below from the full description of the model.

Full model:

$$\text{ACCEPT}_{j,i} \sim \text{Binomial}(\omega_{j,i}, \phi_{j,i})$$

$$\phi_{j,i} = g(\text{PE}_j, \sum_{h \in \mathbb{H}} p_j(h) \mathbb{1}_{i \in h})$$
$$p_j(h) = p(h|\text{PL}_j, \text{PM}_j, \text{OBS}_j) \quad \text{for } h \in \mathbb{H}$$
$$p(h|\text{PL}_j, \text{PM}_j, \text{OBS}_j) \propto p(\text{OBS}_j|h, \text{PL}_j, \text{PM}_j)p(h|\text{PM}_j, \text{PL}_j) = p(\text{OBS}_j|h)p(h|\text{PM}_j, \text{PL}_j)$$
$$p(\text{OBS}_j|h) = \mathbb{1}_{\text{OBS}_j \subseteq h} \frac{1}{|h|^{|\text{OBS}_j|}}$$
$$p(h|\text{PM}_j, \text{PL}_j) \propto p(h|\text{PM}_j)p(h|\text{PL}_j)$$
$$p(h|\text{PM}_j) = \text{PM}_j \mathbb{1}_{\text{mon}}(k) + (1 - \text{PM}_j)(1 - \mathbb{1}_{\text{mon}}(k))$$
$$p(h|\text{PL}_j) = e^{|h|\text{PL}_j}$$

$$\text{PL}_j \sim \mu_{\text{PL}} + \sigma_{\text{PL}} \text{Normal}(0,1)$$
$$\text{logit}(\text{PM}_j) \sim \mu_{\text{PM, CONDITION}_j} + \sigma_{\text{PM, CONDITION}_j} \text{Normal}(0,1)$$
$$\text{PE}_j \sim \text{HalfNormal}(\sigma_{\text{PE}})$$

$$\mu_{\text{PL}} \sim \text{Normal}(0,2)$$
$$\sigma_{\text{PL}} \sim \text{HalfNormal}(2)$$
$$\mu_{\text{PM, CONDITION}_j} \sim \text{Normal}(0,1.5)$$
$$\sigma_{\text{PM, CONDITION}_j} \sim \text{HalfNormal}(1)$$
$$\sigma_{\text{PE}} \sim \Gamma(1,2)$$

Where

1. $\text{ACCEPT}_{j,i}$ is the number of times participant j accepted stimulus i as belonging to the category.

2. $\omega_{j,i}$ is the number of times participant j was asked to categorize stimulus i.

3. $\phi_{j,i}$ is the probability of participant j accepting stimulus i.

4. $\mathbb{H}$ is the set of convex hypotheses.

5. h is used ambiguously as a set of stimuli, e.g. in $i \in h$, and the event that the hypothesis to infer is the set h, e.g. $p(h)$. Context should disambiguate how each instance is meant.

6. $\mathbb{1}_{i \in h}$ is the indicator function for whether hypothesis h contains stimulus i.

7. $g(x,y) = y + (\frac{1}{1+e^{-x}} - 0.5)(1 - 2y)$. For all values of y, $g(x,y) \to 0.5$ as $x \to \inf$ and $g(x,y) = y$ if $x = 0$. We use this function to model the error in j's production behaviour. As the error $\to \inf$, the choice of whether to accept or not a stimulus becomes uncorrelated with the participant's biases.

8. PE_j is j's production error.

9. PL_j is j's preference for large hypotheses.

10. PM_j is j's preference for monotonic hypotheses.

11. OBS_j is the set of stimuli that were presented to j as belonging to the category.

12. $CONDITION_j$ is the condition of participant j, coded as 0 (for distance condition) or 1 (for property condition).

13. We parameterize the Γ distribution with α and β.

The simpler model assumes that the participant's preferences for monotonicity are sampled from a single distribution, rather than difference distributions for the two conditions.

3   Hypothesis testing

Call x^s the value of model variable x at the sth posterior sample, out of S samples. We calculate the posterior distribution of differences between the means of the population-level distribution over preferences for monotonicity between two conditions $\{\mu^s_{\text{PM,CONDITION}_0} - \mu^s_{\text{PM,CONDITION}_1}\}^S_{i=1}$. We calculate the 95% HPD interval, and we accept the hypothesis that there is a difference between the two conditions with respect to the preference for monotonic hypotheses iff the HPD interval does not contain 0.

4   Practicalities

This repository contains the code to fit the model to the experimental data and analyze the results. Fitting the model is computationally intensive, and the LOO CV requires numerous fits. Therefore, we add code to run the model on the server. The code is written for Oracle Grid Engine (`https://en.wikipedia.org/wiki/Oracle_Grid_Engine`), but other implementations can be written based on the code below.

5   References

Xu, Fei, and Joshua B. Tenenbaum. 2007. "Word Learning as Bayesian Inference." Psychological Review 114 (2): 245–72. `https://doi.org/10.1037/0033-295X.114.2.245`.