Informatics 1 Cognitive Science – Tutorial 4

Frank Keller, Carina Silberer, Frank Mollica

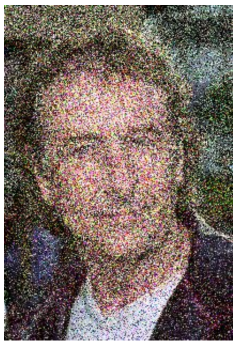
Week 5

1 Bayesian Modeling

Last week, we discussed Bayesian Modeling as a way of capturing human reasoning and decision making. In this exercise, we will look at an example of how we can formalize a cognitive process in Bayesian terms.

Exercise 1 In an experiment on face recognition, participants are presented with images of people they know, and asked to identify them. The images are presented for a very short period of time so that participants may not have time to see the details of the entire face, but are likely to get a general impression of things like hair color and style, overall shape, skin color, etc. In this question we will consider how to formulate the face recognition problem as a probabilistic inference model.

1. What is the hypothesis space in this problem? Is it continuous or discrete? Finite or infinite?
2. What constitutes the observed data d and what kinds of values can it take on?
3. Write down an equation that expresses the inference problem that the participants must solve to identify each face. Describe what each term in the equation represents.
4. What factors might influence the prior in this situation?
5. Suppose one group of participants sees clear images, such as the one on the left below, and another group sees noisy images, such as the one on the right below. Which term(s) in your equation will be different for the noisy group compared to the clear group?



6. What does the model predict about participants' performance with noisy images compared to clear images?

2 Word Learning

In the lectures, we discussed Piantadosi et al.'s program induction model for learning number words. The model considers several possible hypotheses for the definition of number words and uses Bayes Rule to infer the most likely hypothesis as a learner sees data. Recall Bayes rule:

$$P(h|d) = \frac{P(d|h)P(h)}{\sum_i P(d|h_i)P(h_i)}. \tag{1}$$

Exercise 2 We will attempt to replicate Piantadosi et al.'s results. Instead of searching a vast hypothesis space at multiple data amounts, let's focus on the four hypotheses in Figure 1 and the two datasets in Table 1.

One-knower

```
λ S . (if (singleton? S)
          "one"
          undef)
```

Two-knower

```
λ S . (if (singleton? S)
          "one"
          (if (doubleton? S)
              "two"
              undef))
```

Three-knower

```
λ S . (if (singleton? S)
          "one"
          (if (doubleton? S)
              "two"
              (if (tripleton? S)
                  "three"
                  undef))
```

CP-knower

```
λ S . (if (singleton? S)
          "one"
          (next (L (set-difference S
                                   (select S)))))
```

Figure 1: The four hypotheses for number word meanings that we are considering.

Dataset	(one, ·)	(two, :)	(three, :·)	(four, ::)	(five, ::·)	(six, :::)	(seven, :::·)	(eight, ::::)
($N = 30$)	20	4	2	2	0	1	0	1
($N = 60$)	41	10	5	2	0	1	0	1

Table 1: The datasets we are considering. Each column denotes a type of data. Each row contains the number of times that type of data has been seen.

1. In the original model, they used a simplicity prior. Let's try using a prior based on the length of the hypotheses. For each hypothesis, write down its length (include parentheses and punctuation).

2. Let's say the prior probability of a hypothesis h is inversely proportional to its length L_h. So longer lengths are less probable a priori:

$$P(h) = \frac{{L_h}^{-1}}{\sum_i {L_{h_i}}^{-1}} \tag{2}$$

   Fill in the prior in Tables 2 and 3.

3. In the original model, the authors used a noisy size principle likelihood, which considers three possible ways the data might have been generated: (i) undefined, i.e., we get no response (ii) correct according to the hypothesis and (iii) correct by random guessing. In the lecture, we formulated the noisy size principle as:

$$P(w|s,h) = \begin{cases} \frac{1}{N} & \text{if } h(s) = \text{undef} \\ \alpha + (1-\alpha)\frac{1}{N} & \text{if } h(s) = w \\ (1-\alpha)\frac{1}{N} & \text{else} \end{cases} \tag{3}$$

Here, let's simplify things and leave out the first case (undefined answers) and assume that the number of words is constant with $N = 10$. We get:

$$P(w|s,h) = \begin{cases} \alpha + \frac{1-\alpha}{10} & \text{if } h(s) = w \\ \frac{1-\alpha}{10} & \text{else} \end{cases} \tag{4}$$

The parameter α reflects how reliably the data comes from the hypothesis. Let's consider $\alpha = 0.9$ for this exercise. Fill in the rest of the likelihoods in Table 3.

4. Now use Bayes Rule to calculate the posterior beliefs over knower levels. In the real world, we often deal with probabilities of events that are really small. To make the calculations easier we can work in logarithms. Here is the log of Bayes rule:

$$\log P(h|d) = \log P(d|h) + \log P(h) - \log P(d), \tag{5}$$

where $P(d) = \sum_h \exp(\log P(d|h) + \log P(h))$.

Most programming languages have a function that computes this `LogSumExp` operation. To make your life easier, the attached python script walks through this computation. You should be able to plug in the above information and it will return the posterior. You can run it locally or copy it on notable.

5. Take a look at the results from the original model (in the slides). Did we replicate their results? Was hypothesis length an appropriate prior? Take a guess on how we might have to change the prior.

Knower Level	**Prior**	**Likelihood**$_{N=30}$	**Posterior**
1-knower		$(0.91)^{20}(0.01)^{10}$	
2-knower		$(0.91)^{24}(0.01)^{6}$	
3-knower		$(0.91)^{26}(0.01)^{4}$	
CP-knower		$(0.91)^{30}(0.01)^{0}$	

Table 2: Use this table to write down the components of Bayes Rule for the first dataset $N = 30$.

Knower Level	**Prior**	**Likelihood**$_{N=60}$	**Posterior**
1-knower			
2-knower			
3-knower			
CP-knower			

Table 3: Use this table to write down the components of Bayes Rule for the second dataset $N = 60$.