9.19/9.190: Computational Psycholinguistics, Pset 1
due 22 September 2021

6 September 2021

Incremental inference about possessor animacy

English has two CONSTRUCTIONS for grammatically expressing possession within a noun phrase, as exemplified in (1)–(2) below:

(1) the queen's crown (PRENOMINAL or 'S GENITIVE: possessor comes before the possessed noun)

(2) the crown of the queen (POSTNOMINAL or *of* GENITIVE: possessor comes after the possessed noun)

There is a correlation between the ANIMACY of the possessor and the preferred construction: animate possessors, as above, tend to be preferred prenominally relative to inanimate possessors, as in (3)– (4) below (Futrell & Levy, 2019; Rosenbach, 2005):

(3) the book's cover (Prenominal)

(4) the cover of the book (Postnominal)

Here is a set of probabilities that reflects this correlation:

$$\begin{aligned} P(\text{Possessor is } \textbf{animate}) &= 0.4 \\ P(\text{Possessor is } \textbf{prenominal} | \text{Possessor is } \textbf{animate}) &= 0.9 \\ P(\text{Possessor is } \textbf{prenominal} | \text{Possessor is } \textbf{inanimate}) &= 0.25 \end{aligned}$$

Now consider the cognitive state of a language comprehender mid-sentence who has just heard the start of a noun phrase involving a noun that they don't know:

the sneg of the...

Task: Based on the knowledge encoded in the probabilities above, what probability should the comprehender assign to the upcoming possessor being animate? Show your work in carrying out this computation.

Phoneme categorization

The questions in this section relate to ideal probabilistic categorization of instances of the sound categories /b/ and /p/, as covered in Lecture 1 (related readings include Clayards et al., 2008, Feldman et al., 2009).

Assume that a single informative cue (VOT) distinguishes between these categories, and that the distributions of VOT values for these categories can be approximated by Gaussian distributions with means of $\mu_b = 0$ and $\mu_p = 50$. Imagine a context in which the prior probabilities of the two categories differ, $p(/\mathrm{b}/) = 0.75$ and $p(/\mathrm{p}/) = 0.25$.

For a given VOT value x, we can calculate the posterior distribution on the category c that token came from $p(c|x)$ using Bayes rule:

$$p(c|x) = \frac{p(x|c)p(c)}{p(x)} \tag{1}$$

$$= \frac{p(x|c)p(c)}{\Sigma_{c'} p(x|c')p(c')} \tag{2}$$

where the prior $p(c)$ is as given above, the likelihood $p(x|c)$ is given by the Gaussian probability density function

$$p(x|c) = \frac{1}{\sigma_c\sqrt{2\pi}} \exp\left[-\frac{(x-\mu_c)^2}{2\sigma_c^2}\right] \tag{3}$$

and the normalizing constant in the denominator is evaluated by summing across all possible hypotheses $c' \in \{/\mathrm{b}/, /\mathrm{p}/\}$:

$$p(c|x) = \frac{p(x|c)p(c)}{p(x|/\mathrm{b}/)p(/\mathrm{b}/) + p(x|/\mathrm{p}/)p(/\mathrm{p}/)} \tag{4}$$

1. Imagine that both categories had equal variances $\sigma_b^2 = \sigma_p^2 = 144$. Under this assumption, calculate the posterior probability of the category /p/ for a VOT value of 25 ms, i.e., $p(c = /\mathrm{p}/|x = 25\mathrm{ms})$. *Want an additional challenge? Also calculate the posterior for VOT values of* -25*, 0, 50, and 75 ms, and plot these values against VOT.*

2. In fact, VOTs for voiceless stops such as /p/ are more variable than those for voiced stops such as /b/. This means that the Gaussian approximations of these categories should have different variances, such as $\sigma_b^2 = 64$ and $\sigma_p^2 = 144$. Assuming these values, calculate the posterior for a VOT value of 25 ms again. How does the categorization preference change? Why? *Want an additional challenge? Again, calculate the posterior for all VOT values mentioned in the additional-challenge part of (1), and make another plot.*

3. Continuing to assume the unequal-variance parameters as in (2), calculate the posterior for the very low VOT of -200ms. There is some counter-intuitive behavior: what is it? What does this counter-intuitive behavior tell us about the limitations of the model

we've been using? *Want an additional challenge? Extend your graph from (2) with VOT values of* -50*,* -100*,* -150*, and* -200 *ms to see how this counter-intuitive effect develops.*

Tuning an n-gram language model

This Colab notebook contains starter code for this problem.

When we covered n-gram models, you learned about MAXIMUM-LIKELIHOOD ESTIMATION and ADDITIVE SMOOTHING for a bigram model. Maximum-likelihood estimation is equivalent to RELATIVE FREQUENCY ESTIMATION:

$$\widehat{P}_{RFE}(w_i|w_{i-1}) = \frac{\text{Count}(w_{i-1}w_i)}{\text{Count}(w_{i-1})}$$

In ADDITIVE SMOOTHING, a pseudo-count of α is added to each n-gram count, so that the bigram probability of w_i given w_{i-1} is estimated as

$$\widehat{P}_{\alpha}(w_i|w_{i-1}) = \frac{\text{Count}(w_{i-1}w_i) + \alpha}{\text{Count}(w_{i-1}) + \alpha V}$$

where V is the vocabulary size.

Task: consider the following toy datasets, building on the Lecture 2 handout:

Training set	**Validation set**	**Test set**
`dogs chase cats`	`the cats meow`	`cats meow`
`dogs bark`	`the dogs bark`	`dogs chase the birds`
`cats meow`		
`dogs chase birds`		
`cats chase birds`		
`dogs chase the cats`		
`the birds chirp`		

1. Implement a bigram language model with additive smoothing, trained on the training set and with the smoothing parameter α optimized to mimimize the held-out perplexity of the validation set. In choosing which possible values of α to consider, you can use any of a variety of methods; the simplest is grid search, but you can use another method if you prefer.

2. For the values of α that your implementation considers, plot the validation-set perplexity against the test-set perplexity. What value of α worked the best for the validation set? Was it the same that would have worked best for the test set?

Background note: in machine-learning terminology, choosing the value of α based on the held-out perplexity of the validation set is an example of HYPERPARAMETER TUNING. In general, you don't get to look at the test-set performance for all the different choices of the hyperparameters; you have to make your hyperparameter commitments on the basis of exploration against the validation set, and then look at performance on the test set at the end. In the present exercise, I'm asking you to look at both together strictly for pedagogical purposes.

Want to take this farther? Once you've done the above you've completed the assignment, but for further learning you can try tuning α for a larger-scale dataset, such as Wikitext-2 or Wikitext-103 (training sets of about 2 million words and 103 million words respectively, which are still small by contemporary NLP standards!), This will also give you an opportunity to see how well your implementation scales time- and memory-wise as the dataset gets larger, which is often very important for computational work on language.

Open versus closed vocabulary language modeling: if you try to scale up to a larger dataset like suggested above, you will have to deal with the question of how to define the vocabulary size V. In the general case, there may be words in your validation and/or test sets that do not appear in your training set, so you cannot just set V to the number of distinct words that appear in the training set, or you will wind up with an improper probability distribution (why?). A couple of alternative options include:

- "Peeking" at the validation and test sets and defining the vocabulary as including at least the union of all words that appear in training, validation, and test sets; this is what is known as CLOSED VOCABULARY language modeling.

- Defining a vocabulary that does not necessarily include all words that appear in the validation and/or test sets, together with an "unkification" function that maps words that do not appear in this vocabulary to one or more UNKNOWN WORD categories (when there is only one unknown word category, it's traditionally denoted with `<UNK>`). You then convert your training, validation, and test datasets using this unkification function, and all perplexity evaluations involve these converted dataset. This is OPEN VOCABULARY language modeling.

These issues are covered in some more detail in SLP3 section 3.3.1. Note that you don't have to worry about any of this for the toy dataset for the present problem, because every word appearing in the validation and test sets also appears in the training set.

References

Clayards, M., Tanenhaus, M. K., Aslin, R. N., & Jacobs, R. A. (2008). Perception of speech reflects optimal use of probabilistic speech cues. *Cognition*, *108*, 804–809.

Feldman, N. H., Griffiths, T. L., & Morgan, J. L. (2009). The influence of categories on perception: Explaining the perceptual magnet effect as optimal statistical inference. *Psychological Review*, *116*(4), 752–782.

Futrell, R., & Levy, R. P. (2019). Do RNNs learn human-like abstract word order preferences? In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*.

Rosenbach, A. (2005). Animacy versus weight as determinants of grammatical variation in English. *Language*, *81*(3), 613–644.