Computing Dynamic Meanings: Building Integrated Competence-Performance Theories for Semantics

Day 3, part 1: Introduction to Bayesian estimation for linguists

Jakub Dotlačil & Adrian Brasoveanu

ESSLLI 2018, August 8 2018

Plan: the basics of Bayesian statistical modeling

- Bayesian methods are not specific to ACT-R, or to cognitive modeling
- a general framework for doing plausible inference over data – both categorical ('symbolic') and numerical ('subsymbolic') data

Why a Bayesian 'detour'?

- Main goal: integrated, fully formalized theories of competence and performance
- That is, theories that formally / explicitly link:
  - theoretical constructs postulated by generative linguists
  - experimental data generated by widely used psycholinguistics methodologies
- The ACT-R cognitive architecture provides the bridge between ling. theory and exp. data
- ACT-R's performance / subsymbolic components come with a good number of numerical parameters / 'knobs'
- the 'knobs' need to be dialed in to specific settings based on (numerical) experimental data
- Bayesian methods do the 'dialing in' + extra useful stuff

  information about ranges of 'reasonable' values (credible intervals), quantitative comparison of alternative qualitative theories etc.

The Python libraries we need

- `numpy`: fast numerical and vectorial operations
- `matplotlib` and `seaborn`: plotting facilities
- `pandas`: data frames, i.e., data structures well suited for data analysis

  Excel sheets on steroids; similar to R data frames
- finally, `pymc3`: the library for Bayesian modeling – Monte Carlo (MC) methods for `Python3`

Loading the libraries

```
>>> import numpy as np

>>> import matplotlib as mpl
>>> import matplotlib.pyplot as plt
>>> plt.style.use('seaborn')
>>> import seaborn as sns

>>> import pandas as pd

>>> import pymc3 as pm
```

The data

- very simple data set from chapter 3 of Johnson (2008)
- download here: `http://media.wiley.com/product_ancillary/46/14051442/DOWNLOAD/3phonetics.zip`
- unpack the zip file, the file containing the data set is `cherokeeVOT.txt`
- load data (values separated by a tab):

```
>>> VOT_data = pd.read_csv("cherokeeVOT.txt",\
...                        sep="\t")
>>> VOT_data["year"] = \
...     VOT_data["year"].astype('category')
```

The data (ctd.)

Examine the data set:

```
>>> VOT_data.shape
(44, 3)
>>> VOT_data.head(n=3)
   VOT  year Consonant
0   67  1971         k
1  127  1971         k
2   79  1971         k
>>> VOT_data.iloc[[0, 8, 18, 31], :]
    VOT  year Consonant
0    67  1971         k
8   109  1971         t
18   84  2001         k
31   79  2001         t
```

The data (ctd.)

- voice onset times (VOTs) for the same speaker for:
  - 2 different years: 1971 and 2001
  - 2 consonants: [t] and [k]
- VOT is the point at which voicing/vocal fold vibration begins after the initial time of consonantal articulation
  - simple unaspirated voiceless stops like [t] in [k^hɪt] (kit) or [k] in [t^hɪk] (tic) have a VOT near 0: the voicing of a subsequent sonorant begins as soon as the stop is released.
  - aspirated stops like [k^h] in [k^hɪt] or [t^h] in [t^hɪk] have a larger VOT: the voicing of the following vowel [ɪ] does not start as soon as the stop is released.
  - the longer the VOT (the longer the vocal folds don't vibrate), the stronger the aspiration.

Main question about this data set

We can ask several questions about this data set; we focus on:

Did the VOT of the speaker change from 1971 to 2001?

(aggregating over the 2 consonants)

Formalizing the main question

- so, we want to model VOT as a function of year
- one way: estimate the two means for the two years
- in a Bayesian framework, we estimate the means and our uncertainty about them – two full probability distributions, one for each of the means
- but: estimating mean VOTs will not give us a direct answer to our question: is there a difference in VOT between the two years?
- in a Bayesian framework, we could still answer the question given a two-mean model
- more straightforward (and closer to frequentist estimation) to estimate the difference between the two years directly

Formalizing the main question (ctd.)

- so, we estimate:
  - mean VOT for 1971 (together with our uncertainty about it)
  - mean difference between the 1971 VOT and 2001 VOT (together with our uncertainty about it)
- can obtain mean VOT for 2001 by starting with mean for 1971 and adding the difference
- to answer main question (did VOT change from 1971 to 2001?), we examine probability distribution for VOT difference:
  - is ‘enough’, e.g., 95%, of that probability distribution away from 0? (or some small region around 0)
  - if so, we’re pretty confident the VOT changed

The structure of the statistical argument

- this type of argument is the opposite of what linguists are trying to do
- from very early on in our linguistic training:
  - we are presented with some data
  - we automatically assume there is a pattern in the data
  - we try to identify the pattern / generalization and build a theory to capture it
- as empirically-driven statistical modelers, we skeptically ask instead: is there really a pattern in the data?
- how sure are we that we're not hallucinating regularities in white noise / finding patterns in fleeting clouds?
- we're skeptical and quantify our (un)certainty about the presence of such patterns
- only if we are certain 'enough' that there is a pattern, we start building a theory for it

Formalizing the main question: final version

Our question about the VOT data set is unpacked as follows:

i. can we actually show with enough credibility that the VOT actually changed between the two years (1971 and 2001)?
ii. if we can, what is the magnitude of the change (in ms)?
iii. finally, what is our uncertainty about that magnitude?

We're looking for an answer of the form:

- there was a change of x_{mean} ms on average
- we're 95% certain that the actual value of the change is somewhere in the interval $(x_{\text{lower limit}}, x_{\text{upper limit}})$
- this interval excludes 0, which shows that change is actually credible

Now, let's specify the actual model.

officially, the model we are about to specify is called a t-test, or a linear regression with one binary categorical predictor

How does Bayesian estimation work?

- start with a prior belief about the quantities of interest (VOT for 1971, VOT difference between 2001 and 1971)
  - ‘prior’: these are our beliefs before we see the data
  - beliefs take the form of full probability distributions: we say what values are possible for the quantities of interest and which of them plausible (before we see the data)
- then, update prior beliefs with the data stored in the `"VOT"` and `"year"` columns of our data set
- result: we shift/update our prior beliefs in the direction of the data; 2 posterior probability distributions
  - posterior distribution of the mean 1971 VOT
  - posterior distribution of VOT difference

More about posterior probability distributions

- posteriors: weighted average of the priors and the data
- if priors very strong (not the case here; see next slide), posteriors reflect the data to smaller extent
- if a lot of data, and with low variability, posteriors reflect data to larger / overwhelming extent

Weak priors

We have weak prior beliefs about VOTs. We know:

- VOT has to be positive (we're dealing with voiceless stops here)
- a VOT cannot really be more than 500-600 ms: the average word-per-minute rate is more than 100, so it takes about half a second (500 ms) to say a full word in normal speech
- prior belief for 1971 VOT: half-normal (half-Gaussian) with a standard deviation of 200 ms
- that is, a normal (Gaussian) distribution centered at 0 and 'folded over' so that all the probability mass over negative values gets transferred to the positive values

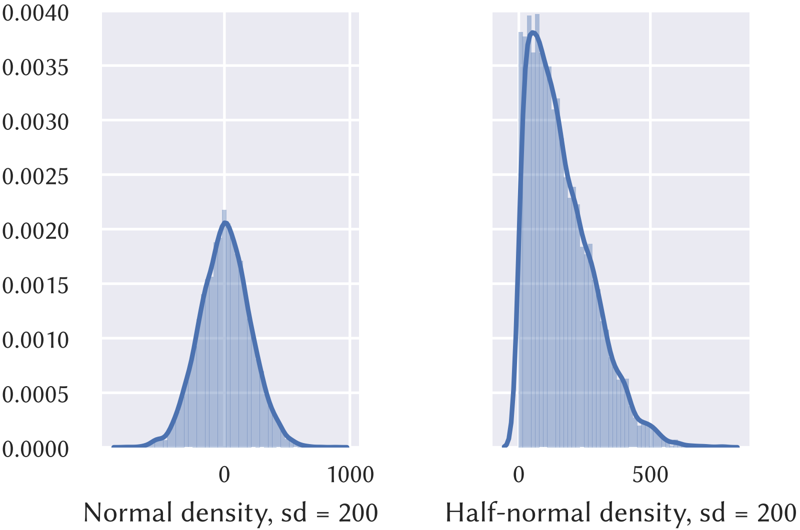
Weak prior for 1971 VOT

- plot a normal and half-normal dist. with sd = 200
- in the process, introduce basics of pymc3 models

```
>>> from pymc3.backends import SQLite
>>> from pymc3.backends.sqlite import load
>>> VOT_model = pm.Model()
>>> with VOT_model:
...     norm = pm.Normal('norm', mu=0, sd=200)
...     half_norm = pm.HalfNormal('half_norm',\
...                              sd=200)
...     #db = SQLite('half_normal_trace.sqlite')
...     #trace = pm.sample(draws=5000, trace=db,\
...                        #n_init=500)
...     # load results / trace of previous run
...     trace = load('half_normal_trace.sqlite')
...
```

```
>>> def generate_half_normal_prior_figure():
...     fig, (ax1, ax2) = plt.subplots(ncols=2,\
...                                    nrows=1, sharey=True)
...     fig.set_size_inches(4.6, 2.9)
...     sns.distplot(trace['norm'], hist=True,\
...                  ax=ax1)
...     ax1.set_xlabel('Normal density, sd = 200')
...     sns.distplot(trace['half_norm'], hist=True,\
...                  ax=ax2)
...     ax2.set_xlabel('Half-normal density,\
...                    sd = 200')
...     plt.tight_layout(pad=0.5, w_pad=0.2,
...                      h_pad=0.7)
...     plt.savefig('half_normal_prior.pgf')
...     plt.savefig('half_normal_prior.pdf')
...
>>> generate_half_normal_prior_figure()
```

Figure: A normal and a half-normal probability density



Weak priors for 1971 VOT and VOT difference

- we use the half-normal density in the right panel of the figure as our prior for the 1971 VOT
  - very weak, low information prior with very mild constraints:
  - we know the VOT is positive
  - we think it is somewhere in the $(0, 600)$ ms interval, with a (reasonable) preference for the $(0, 400)$ ms interval
- we use the normal density in the left panel of the figure as our prior for the VOT difference
  - the prior allows for a positive difference (2001 VOT $>$ 1971 VOT), a negative difference (2001 VOT $<$ 1971 VOT), or 0 difference (2001 VOT $=$ 1971 VOT)
  - difference cannot be larger than 600 ms in absolute value since both VOTs are positive and at most about 600 ms

Model for the data: the likelihood function

- let's specify the model for how (we think) nature generated the data
- need to estimate 2 quantities:
  - the mean VOT for 1971: VOT_{1971}
  - the mean difference between the 1971 and the 2001 VOTs: $VOT_{2001-1971}$
- need to mathematically specify how VOT is a function of year with these 2 quantities
- rewrite the *year* variable as taking either a value of 0 (VOT from 1971) or a value of 1 (VOT from 2001) – 'dummy coding' / 'one-hot encoding'

```
>>> VOT_data["dummy_year"] =\
...     (VOT_data["year"] == 2001).astype("int")
```

Model for the data: the likelihood function

VOT as a function of year:

$$\boxed{VOT = VOT_{1971} + year \cdot VOT_{2001-1971} + noise}$$

- if *VOT* comes from 1971, our dummy-coding for *year* says that $year = 0$
  $VOT = VOT_{1971} + 0 \cdot VOT_{2001-1971} + noise = VOT_{1971} + noise$
- if *VOT* comes from 2001, our dummy-coding for *year* says that $year = 1$
  $VOT = VOT_{1971} + 1 \cdot VOT_{2001-1971} + noise = VOT_{2001} + noise$

Posterior beliefs

- we now implement the model and ask `pymc3` to give us the posterior distributions for the quantities of interest
  - `mean_VOT_1971`
  - `mean_VOT_diff`

Model implementation: priors

```
>>> year = np.array(VOT_data["dummy_year"])
>>> VOT = np.array(VOT_data["VOT"])
>>> VOT_model = pm.Model()
>>> with VOT_model:
...     # priors
...     mean_VOT_1971 =\
...         pm.HalfNormal('mean_VOT_1971', sd=200)
...     mean_VOT_diff =\
...         pm.Normal('mean_VOT_diff', mu=0,
...                   sd=200)
...     sigma = pm.HalfNormal('sigma', sd=200)
...
```

Model implementation: likelihood and posteriors

```
>>> with VOT_model:
...     # likelihood
...     observed_VOT =\
...         pm.Normal('observed_VOT',
...             mu=mean_VOT_1971 + \
...                year*mean_VOT_diff,
...             sd=sigma, observed=VOT)
...     # sample posteriors
...     #db = SQLite('VOT_model_trace.sqlite')
...     #trace = pm.sample(draws=5000, trace=db,\
...                   #n_init=50000, njobs=4)
...     # we use a previous run
...     trace = load('VOT_model_trace.sqlite')
...
```

More about model likelihood

- each observed VOT: an imperfect, noisy reflection of the mean VOT for the year in which VOT was collected
- add normally distributed noise to that mean to obtain actual VOT
- this normal distribution for the noise has a standard deviation `sigma`
- we do not know how big the noise is, so specify weak, low information prior (half-normal because `sigma` has to be positive)
- likelihood for observed VOTs: normal distribution around the year mean with a `sigma` standard deviation

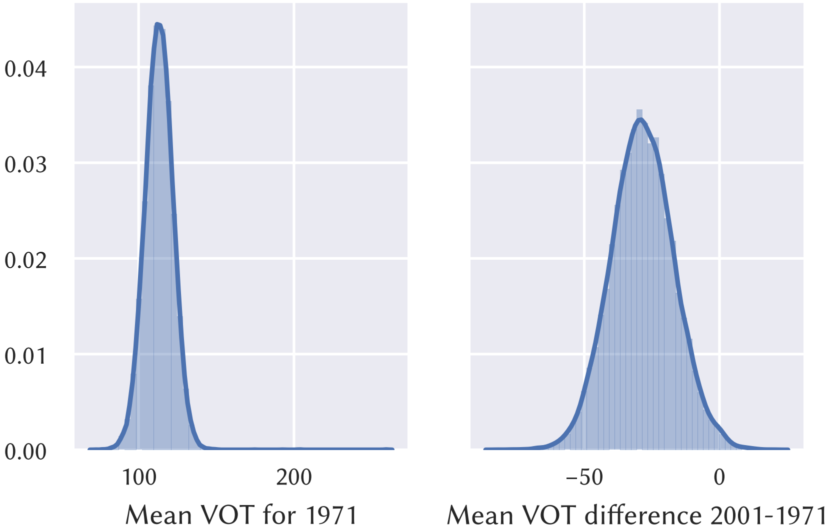
Estimated model parameters

(*i*) the mean VOT for 1971 (`mean_VOT_1971`)

(*ii*) the mean difference in VOT between 2001 and 1971 (`mean_VOT_diff`)

(*iii*) the magnitude of the noise / dispersion of the actual VOTs around the two mean VOTs for years 1971 and 2001 (`sigma`)

Figure: VOT model: posterior distributions



Answering our theoretical question

To answer the theoretical question of interest, we examine the 95% credible interval (CRI) for the VOT difference:

(the 95% highest posterior density CRI; the central 95% CRI also OK)

```
>>> mean_VOT_difference = trace['mean_VOT_diff']
>>> pm.hpd(mean_VOT_difference).round(2)
array([-50.92,  -5.39])
```

We are 95% certain that the difference in VOT between 2001 and 1971 is:

- negative
- between the values listed above

Other quantities of interest

We can find out information about other quantities of interest:

```
>>> mean_VOT_difference.mean().round(2)
-28.42
>>> mean_VOT_difference.std().round(2)
11.63
>>> mean_VOT_1971 = trace['mean_VOT_1971']
>>> mean_VOT_1971.mean().round(2)
113.13
>>> mean_VOT_1971.std().round(2)
9.02
>>> noise = trace['sigma']
>>> noise.mean().round(2)
37.19
>>> noise.std().round(2)
4.64
```

Quick comparison with frequentist estimation

Means & sd.s ≈ frequentist ones, e.g., using `lm()` in R:

```
VOT_data = read.delim("cherokeeVOT.txt", sep="\t")
VOT_data$year = factor(VOT_data$year)
summary(lm(VOT ~ year, data=VOT_data))
[...]
                       Estimate Std. Error [...]
[VOT_1971]               113.50       8.49 [...]
[VOT_difference]         -28.85      11.05 [...]
[...]
```

Summary

We've shown how to:

- formulate a Bayesian model for a problem of interest
- estimate the model parameters
- use the estimates to answer the theoretical question

Advantages of Bayesian methods for data analysis and cognitive modeling:

- mathematically encode the common-sense idea that
  - we have beliefs about what is plausible and (un)likely to happen
  - we learn from experience and update these beliefs
- access to a very powerful and flexible way of empirically evaluating linguistic theories
- theories faithfully and directly encoded in specific structures for the priors and for the way we think the data is generated (the likelihood)

Where we're going next

- taking mathematically specified cognitive models and embedding them in a Bayesian model for empirical evaluation – essential when we start introducing the performance / subsymbolic components of ACT-R
  - subsymbolic components of ACT-R: a good number of real-valued parameters / 'knobs'
  - Bayesian inference enables us to learn the best settings for these parameters from the data
- also, embedding rich cognitive theories in Bayesian models enables us to do quantitative comparison for qualitative theories

Johnson, K. 2008. Quantitative methods in linguistics. Blackwell Pub.