Parameter estimation

- For a trigram model:

\[ P(w_n|w_{n-2}, w_{n-1}) = \frac{P(w_{n-2}, w_{n-1}, w_n)}{P(w_{n-2}, w_{n-1})} \]

So, we need to estimate the marginal bigram and trigram probabilities.

- If we want to estimate the parameter vector \( \theta \) from a training corpus \( c \), then find:

\[
\hat{\theta} = \underset{\theta}{\text{argmax}} P(\theta|c)
= \underset{\theta}{\text{argmax}} \frac{P(c|\theta) P(\theta)}{P(c)}
= \underset{\theta}{\text{argmax}} P(c|\theta) P(\theta)
\]
Parameter estimation

- If we assume $P(\theta)$ is uniform, then we get the maximum likelihood estimate:
  \[
  \hat{\theta} = \arg\max_{\theta} P(c|\theta)
  \]

- For trigram probabilities, the MLE is:
  \[
  P(w_1, w_2, w_3) = \frac{C(w_1, w_2, w_3)}{N}
  \]

- **Maximum** likelihood sounds good, but how well does it work?
Parameter estimation

- What’s the probability assigned to a word in a context in which it did not occur in the training data?

\[
P(w_n | w_{n-2}, w_{n-1}) = \frac{P(w_{n-2}, w_{n-1}, w_n)}{P(w_{n-2}, w_{n-1})} = \frac{C(w_{n-2}, w_{n-1}, w_n)}{C(w_{n-2}, w_{n-1})} = 0
\]

- The likelihood of a corpus which contains this will then be 0, and the cross entropy (and perplexity) is \(\infty\)!

- The MLE uses only information from the training text – how can we do better?
Laplace’s Law of Succession

• Laplace (c. 1775) was interested in inductive reasoning: what can we conclude from a sequence of observations?

• Suppose we’ve got a set of coins labeled 0, . . . , N such that each come up heads with probability \( \frac{i}{N} \)

• Given that you’ve seen one of these coins come up heads \( n \) times in a row, which coin are you looking at?

• Averaging over all the coins, the probability of getting \( n \) heads in a row is:

\[
P(n) = \frac{1}{N + 1} \sum_{i=0}^{N} \left( \frac{i}{N} \right)^n
\]
Laplace’s Law of Succession

The probability of getting heads on the next toss is:

\[ P(n+1|n) = \frac{P(n+1)}{P(n)} \]

If we take the limit as \( N \to \infty \), then

\[
\lim_{N \to \infty} P(n) \approx \int_0^1 x^n \, dx = \frac{1}{n+1}
\]

And:

\[
\lim_{N \to \infty} P(n+1|n) = \frac{n+1}{n+2}
\]
Laplace’s Law of Succession

• If we’ve seen 10 heads, the probability that the next toss will be heads is \( \frac{10 + 1}{10 + 2} = \frac{11}{12} \) (cf. MLE)

• Laplace’s Law was ridiculed even at the time, and has been at the center of the frequentist/Bayesian controversy ever since

• The sun has risen every day for the last 5,000 years. The probability that it won’t rise tomorrow is:

\[
P(\text{no sun}) = 1 - P(\text{sun})
\]

\[
= 1 - \frac{5,000 \times 365.25 + 1}{5,000 \times 365.25 + 2}
\]

\[
= \frac{1}{1,826,252}
\]
Laplace’s Law of Succession

- The probability that a 10-year-old will be alive next year is \( \frac{11}{12} \approx 0.92 \), but the probability that a 70-year-old will be alive is \( \frac{71}{72} \approx 0.99 \).

- If cold fusion is achieved in the lab once, the probability that it will be achieved when the experiment is repeated exactly is \( \frac{2}{3} \).

- Cold fusion has never been achieved in the lab, so the probability that it will be achieved on the first try is \( \frac{1}{2} \).

- These are all misapplications of Laplace’s Law!
Laplace’s Law of Succession

- If we pick dice instead of coins, then we can generalize this to estimating multinomial probabilities (de Morgan):

\[
P(w_1, \ldots, w_n) = \frac{C(w_1, \ldots, w_n) + 1}{N + B}
\]

- For zero counts, this gives a probability of \( \frac{1}{N+B} \), so is an improvement over MLE

- But, it depends on knowing how many things we didn’t see (to calculate \( B \)), and...

- generally the number of unseen \( n \)-grams \( B \) is much larger than the size of the corpus \( N \), so most of the probability mass gets assigned to unseen events!
Laplace’s Law of Succession

- The Brown corpus is 1,170,811 words long
- It includes 53,849 different word types, so the number of possible bigram types is $53,849 \times 53,849 = 2,899,714,801$
- Only 450,852 bigram types are actually observed, so the total probability mass assigned to unseen bigrams is:

$$P(\text{unseen}) = \frac{0 + 1}{1,170,811 + 2,899,714,801} \times (2,899,714,801 - 450,852) = 0.9994$$

- The best we can say is that Laplace probability estimates increase monotonically with observed frequency
Laplace’s Law of Succession

• Laplace’s law depends on the assumption of a uniform prior distribution over models

• But, what about Zipf’s law?

• A more general prior is the *Dirichlet distribution*:
  – includes the uniform distribution as a special case
  – also includes lumpy power law distributions
  – conjugate prior for multinomial distribution
Lidstone’s Law of Succession

- A more general alternative based on a Dirichlet prior, proposed by Hardy (1882) and Lidstone (1920):

\[ P(w_1, \ldots, w_n) = \frac{C(w_1, \ldots, w_n) + \lambda}{N + B\lambda} \]

- Johnson (1932) showed that if we set \( \mu = N/(N + B\lambda) \), then we can rewrite this as:

\[ P(w_1, \ldots, w_n) = \mu \frac{C(w_1, \ldots, w_n)}{N} + (1 - \mu) \frac{1}{B} \]

- This is a *linear interpretation* between the ML and uniform estimates, with parameter \( \mu \) representing our relative confidence in the two estimates.
Lidstone’s Law of Succession

- What value of $\lambda$ should we use?

- A common guess is $\lambda = 1/2$ (known as the Jeffreys-Perks Law, the Krichevsky-Trofimov estimator, or Expected Likelihood Estimation)

- Another good one is $\lambda = 1/B$ (Schurmann-Grassberger Law)

- Useful values vary widely, so $\lambda$ could be estimated using held-out data

- Still suffers from the general problem that for a small sample, the prior determines the solution (nuisance parameters, hyperparameters)
Cross-validation methods

• Divide training data into two parts a and b

• Suppose an $n$-gram $w$ occurs $r$ times in part a

• Let $N_r^a$ be the total number of $n$-grams that occurred $r$ times in part a, and $T_r^b$ be the total number of times an $n$-gram which occurred $r$ times in part $a$ occurs in part $b$

• Held out estimation:

$$P(w) = \frac{T_r^b}{N_r^a N}$$

• Deleted interpolation:

$$P(w) = \frac{T_r^b + T_r^a}{N(N_r^a + N_r^b)}$$
Good-Turing theorem

- Another approach to the problem was taken by Alan Turing and I.J. Good at Bletchley Park (Good 1953)
- Resurrected and simplified by William Gale (AT&T) and colleagues in 1991
- Used in population biology catch-and-release studies to estimate the number of animals you didn’t see
- Applied to language modeling by IBM speech group
- Related to leave one cross validation