Laplace Smoothing Method

(*Part 2 of Modelling Textual Data Series*)

Rey R. Cuenca*

May 2022

Abstract

This paper gives a quick introduction about the Laplace smoothing method on how to deal with zero counts in language model estimation.

1 Introduction

Recall from the previous lesson that rarity of particular token sequence lead to zero counts during the estimation of probabilities. As shown this is very common for n-gram models that are of longer lengths like the following:

$$C(\texttt{in}, \texttt{the}, \texttt{garden}, \texttt{just}, \texttt{outside}, \texttt{city}) = 0$$

Such counts lead to underestimation of the true probabilities. Thus, we discuss in this article a collection of methods called *smoothing* to address this issue.

2 Dictionaries and OOVs

Let $\mathcal{T}$ be all tokens the in the training set with $|\mathcal{T}| = N$. Futhermore, let $\mathcal{V}$ be the collection of all unique words in $\mathcal{T}$ such that $|\mathcal{V}| = V$. We call $\mathcal{V}$ as the *dictionary* or *vocabulary*. Tokens that our found in the test but does not belong in $\mathcal{V}$ are what we call *unkown words* or *out of vocabulary* (OOV) words.

*Department of Mathematics and Statistics, MSU-Iligan Institute of Technology

3 Laplace Smoothing

Recall that we estimate the n-gram probabilities $P(w_k|\mathbf{w}_{(k-n+1):(k-1)})$ as follows

$$P(w_k|\mathbf{w}_{(k-n+1):(k-1)}) = \frac{C(\mathbf{w}_{(k-n+1:k)})}{C(\mathbf{w}_{(k-n+1):(k-1)})} \tag{1}$$

The common problem of zero counts is when the numerator of the estimator above is zero, i.e., $C(\mathbf{w}_{(k-n+1):k}) = 0$. One way to solve this is by adding 1 to the numerator count, i.e.,

$$C(\mathbf{w}_{(k-n+1):k}) \longrightarrow C(\mathbf{w}_{(k-n+1):k}) + 1$$

However, we also need to adjust the denominator in order for new P_{LP} to be still a probability. To do this, note that

$$P(w_k|\mathbf{w}_{(k-n+1):(k-1)}) = \frac{C(\mathbf{w}_{(k-n+1):k)})}{C(\mathbf{w}_{(k-n+1):(k-1)})} = \frac{C(\mathbf{w}_{(k-n+1):k})}{\sum_{w\in\mathcal{V}} C(\mathbf{w}_{(k-n+1):(k-1)}w)}$$

Thus,

$$\begin{aligned} P_{\text{LP}}(w_k|\mathbf{w}_{(k-n+1):(k-1)}) &= \frac{C(\mathbf{w}_{(k-n+1):k}) + 1}{\sum_{w\in\mathcal{V}} \left[C(\mathbf{w}_{(k-n+1):k}) + 1\right]} \\ &= \frac{C(\mathbf{w}_{(k-n+1):k}) + 1}{\sum_{w\in\mathcal{V}} C(\mathbf{w}_{(k-n+1):k}) + \sum_{w\in\mathcal{V}} 1} \\ &= \frac{C(\mathbf{w}_{(k-n+1):k}) + 1}{C(\mathbf{w}_{(k-n+1):(k-1)}) + V} \end{aligned}$$

Therefore, *Laplace smoothed* estimate of the probability is

$$P_{\text{LP}}(w_k|\mathbf{w}_{(k-n+1):(k-1)}) = \frac{C(\mathbf{w}_{(k-n+1):k}) + 1}{C(\mathbf{w}_{(k-n+1):(k-1)}) + V}.$$

Example 3.1. Now let us consider some examples illustrating the Laplace smoothing method. We'll use data from the now-defunct Berkeley Restaurant Project, a dialogue system from the last century that answered questions about a database of restaurants in Berkeley, California (Dan Jurafsky et al., 1994). For the current illustration, we use only 8 tokens from the corpus with unigram and bigram counts tabulated respectively in Tables 1 and 2. This tables are saved as csv files in `/data/berkeley-unigram.csv` and `/data/berkeley-bigram.csv`.

Table 1: Unigram counts for eight of the wordsin the Berkeley Restaurant Project corpus of 9332 sentences

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

Table 2: Bigram counts for eight of the words in the Berkeley Restaurant Project corpus of 9332 sentences

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Our goal is to fill in these zeros in the bigram counts. First we note that we are restricting our vocabularies to 8 words thus

$$\mathcal{V} = \{\texttt{i}, \texttt{want}, \texttt{to}, \texttt{eat}, \texttt{chinese}, \texttt{food}, \texttt{lunch}, \texttt{spend}\}.$$

and $|\mathcal{V}| = V = 8$. For simplicity, let us denote as matrix W the bigram counts in Table 2, i.e.,

$$W = \begin{bmatrix} 5 & 827 & - & 9 & - & - & - & 2 \\ 2 & - & 608 & 1 & 6 & 6 & 5 & 1 \\ 2 & - & 4 & 686 & 2 & - & 6 & 211 \\ - & - & 2 & - & 16 & 2 & 42 & - \\ 1 & - & - & - & - & 82 & 1 & - \\ 15 & - & 15 & - & 1 & 4 & - & - \\ 2 & - & - & - & - & 1 & - & - \\ 1 & - & 1 & - & - & - & - & - \end{bmatrix}$$

where "$-$" denotes zero counts and $W = (C(w_i, w_j))$ with $w_i, w_j \in \mathcal{V}$. Without smoothing the bigram probabilities are derived by dividing element-wise the values of Table 1 to each of the rows of W to arrive with the following matrix P_b.

$$P_b = \begin{bmatrix} 0.002 & 0.33 & - & 0.0036 & - & - & - & 0.00079 \\ 0.0022 & - & 0.66 & 0.0011 & 0.0065 & 0.0065 & 0.0054 & 0.0011 \\ 0.00083 & - & 0.0017 & 0.28 & 0.00083 & - & 0.0025 & 0.087 \\ - & - & 0.0027 & - & 0.021 & 0.0027 & 0.056 & - \\ 0.0063 & - & - & - & - & 0.52 & 0.0063 & - \\ 0.014 & - & 0.014 & - & 0.00091 & 0.0037 & - & - \\ 0.0059 & - & - & - & - & 0.0029 & - & - \\ 0.0036 & - & 0.0036 & - & - & - & - & - \end{bmatrix}$$

Note that $P_b = (P_{ij})$ where (i,j)-th entry is $P_{ij} = C(w_i, w_j)/C(w_i) = P(w_j|w_i)$. For example, from matrix P_b, we can see the following probabilities

$$P(\texttt{i}|\texttt{i}) = 0.002 \quad \text{and} \quad P(\texttt{want}|\texttt{i}) = 0.33.$$

Now, let us try to smooth matrix P_b using Laplace smoothing. To do this, first we add 1 to each of the entries in W and denote the resulting matrix as W^*,

$$W^* = W + 1 = \begin{bmatrix} 6 & 828 & 1 & 10 & 1 & 1 & 1 & 3 \\ 3 & 1 & 609 & 2 & 7 & 7 & 6 & 2 \\ 3 & 1 & 5 & 687 & 3 & 1 & 7 & 212 \\ 1 & 1 & 3 & 1 & 17 & 3 & 43 & 1 \\ 2 & 1 & 1 & 1 & 1 & 83 & 2 & 1 \\ 16 & 1 & 16 & 1 & 2 & 5 & 1 & 1 \\ 3 & 1 & 1 & 1 & 1 & 2 & 1 & 1 \\ 2 & 1 & 2 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Next is we divide element-wise the values of Table 1 to each of the rows of W^* but this time with additional term in the divisor $V = 8$. If we denote the smoothed bigram probability matrix as $P_{\text{LP}} = (P^*_{ij})$, then

$$P^*_{ij} = \frac{C(w_i, w_j) + 1}{C(w_i) + 8}$$

Thus, matrix P_{LP} of Laplace-smoothed probabilities is

$$P_b = \begin{bmatrix} 0.0024 & 0.33 & 0.00039 & 0.0039 & 0.00039 & 0.00039 & 0.00039 & 0.0012 \\ 0.0032 & 0.0011 & 0.65 & 0.0021 & 0.0075 & 0.0075 & 0.0064 & 0.0021 \\ 0.0012 & 0.00041 & 0.0021 & 0.28 & 0.0012 & 0.00041 & 0.0029 & 0.087 \\ 0.0013 & 0.0013 & 0.004 & 0.0013 & 0.023 & 0.004 & 0.057 & 0.0013 \\ 0.012 & 0.006 & 0.006 & 0.006 & 0.006 & 0.5 & 0.012 & 0.006 \\ 0.015 & 0.00091 & 0.015 & 0.00091 & 0.0018 & 0.0045 & 0.00091 & 0.00091 \\ 0.0086 & 0.0029 & 0.0029 & 0.0029 & 0.0029 & 0.0057 & 0.0029 & 0.0029 \\ 0.007 & 0.0035 & 0.007 & 0.0035 & 0.0035 & 0.0035 & 0.0035 & 0.0035 \end{bmatrix}$$

Indicating the tokens gives us Table 3.

Table 3: New bigram probabilities smoothed using Laplace method.

	i	want	to	eat	chinese	food	lunch	spend
i	0.0024	0.33	0.00039	0.0039	0.00039	0.00039	0.00039	0.0012
want	0.0032	0.0011	0.65	0.0021	0.0075	0.0075	0.0064	0.0021
to	0.0012	0.00041	0.0021	0.28	0.0012	0.00041	0.0029	0.087
eat	0.0013	0.0013	0.004	0.0013	0.023	0.004	0.057	0.0013
chinese	0.012	0.006	0.006	0.006	0.006	0.5	0.012	0.006
food	0.015	0.00091	0.015	0.00091	0.0018	0.0045	0.00091	0.00091
lunch	0.0086	0.0029	0.0029	0.0029	0.0029	0.0057	0.0029	0.0029
spend	0.007	0.0035	0.007	0.0035	0.0035	0.0035	0.0035	0.0035

With smoothed probabilities,

$$P(\texttt{i}|\texttt{i}) = 0.0024 \quad \text{and} \quad P(\texttt{to}|\texttt{i}) = 0.00039.$$

4 Other Methods

There are many other smoothing methods not included in this discussion paper. For other methods, you can refer to (Daniel Jurafsky & Martin, 2021, p. 42).

References

Jurafsky, Daniel, & Martin, J. H. (2021). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition* (3ed draft.). Retrieved May 31, 2013, from https://web.stanford.edu/~jurafsky/slp3/

Jurafsky, Dan, Wooters, C., Tajchman, G. N., Segal, J., Stolcke, A., Fosler-Lussier, E., & Morgan, N. (1994). The berkeley restaurant project. *ICSLP*.