Lecture 11: The Good-Turing Estimate

Scribes: Ellis Weng, Andrew Owens

March 4, 2010

1 Introduction

In many language-related tasks, it would be extremely useful to know the probability that a sentence or word sequence will occur in a document. However, there is not enough data to account for all word sequences. Thus, n-gram models are used to approximate the probability of word sequences. Making an independence assumption between the n-grams reduces some of the problems with data sparsity, but even n-gram models can have sparsity problems. For example, the Google corpus has 1 trillion words of running English text. There are 13 million words that occur over 200 times, so there are at least 169 trillion potential bigrams - much more than the 1 trillion words in the corpus. Smoothing is a strategy used to account for this data sparsity. In this lecture, we will explore Good-Turing smoothing, a particular kind of smoothing.

2 Setup

Suppose we have the set of all possible item types: $X = \{x_1, ..., x_m\}$. These item types may be n-grams, but for simplicity, we will consider unigram item types. For example, $X = \{\text{the}, \text{bad}, \text{cat}, \text{dog}\}$.

We also have a sequence W of N independent samples: $W = w_1$, ..., w_n, where $w_k \in X$. We want to estimate $\theta[j]$, the probability that a future sample will be x_j. We can assume $\theta[j] > 0$ because we want to account for the possibility of a word occurring even if it does not appear in the corpus. This implies that the relative frequency estimate $\frac{\#(x_j)}{N}$, where $\#(x_j)$ is the count of x_j in W, is not desirable or accurate for small counts. Here we run into a problem: how can we estimate the probability of something we have never seen before?

In order to reduce the number of parameters, we introduce the idea of tying parameters based on observed events in W, a key idea in Good-Turing smoothing. We can reduce the number of parameters by making the following assumption: if $\#(x_j) = \#(x_{j'})$, then $\theta[j] = \theta[j']$. In other words, if two words appear the same number of times in the corpus, we assume that they have the same probability of occurring in general. This assumption is not entirely realistic; it may be a coincidence that these two items appeared the same number

of times. However, this assumption significantly reduces the number of the parameters.

With this assumption, we introduce the notation $\theta(r)$ to mean the probability of a word occurring given that it appeared r times in W. We also let N_r denote the number of item types that occur exactly r times in W. In other words, $N_r = |\{x_j : \#(x_j) = r\}|$. For example, if $W =$ the, bad, cat, the, cat then:

$N_0 = 1$ because dog does not appear in W,
$N_1 = 1$ because "bad" appears once in W,
$N_2 = 2$ because the words "cat" and "the" appear twice in W.

With these definitions, the following property holds:

$$N = \sum_r r N_r. \tag{1}$$

We now introduce the Good-Turing estimate for $\theta(r)$.

$$\hat{\theta}(r) = \frac{1}{N}(r+1)\frac{N_{r+1}}{N_r} \tag{2}$$

This estimate seems strange at this point, but we will present two derivations to justify it. As a sanity check, we verify that the sum of the word-occurrence probabilities is 1.

$$\begin{aligned} \sum_j \hat{\theta}[j] &= \sum_r \hat{\theta}(r) N_r & (3)\\ &= \frac{1}{N}\sum_r [(r+1)\frac{N_{r+1}}{N_r}] N_r & (4)\\ &= \frac{1}{N}\sum_r (r+1) N_{r+1}. & (5) \end{aligned}$$

We show $\sum_r (r+1)N_{r+1} = \sum_r r N_r$. All of the terms in the right hand side are also present in the left hand side (except for the term where $r = 0$, which contributes nothing). The only term that appears on the left hand side but not the right is $(r_m+1)N_{r_m+1}$ where r_m is the maximum number of times any word appears in the corpus. Since $N_{r_m+1} = 0$, this term also contributes nothing to the summation. Thus equality holds, and

$$\sum_j \hat{\theta}[j] = \frac{1}{N}\sum_r (r+1) N_{r+1} = \frac{1}{N}\sum_r r N_r = N/N = 1. \tag{6}$$

3 First Derivation

For the first derivation, we will make up a "generative" story for W. Start by assuming that we have access to $\theta[j]$ (remember that we're trying to derive $\hat{\theta}(r)$

and the problem is that the $\theta[j]'s$ for different terms that occur exactly r times might be different). Draw j (hence $\theta[j]$) uniformly at random from $\{1, 2, ..., m\}$. Then flip a coin N times, with $\theta[j]$ being the probability of success (e.g. Yes, Yes, No, ..., No, Yes). The number of successes is the number of times the word x_j is generated by our model. If x_j appears exactly r times, then throw $\theta[j]$ into the average for $\hat{\theta}(r)$. At the end of this process, $\hat{\theta}(r)$ will (approximately) be the average of the $\theta[j]$ for which $\#(x_j) = r$. More precisely, we set

$$\hat{\theta}(r) = \mathbb{E}[\theta[j] \mid \#(x_j) = r] = \sum_j \theta[j] \Pr(\theta[j] \mid \#(x_j) = r). \tag{7}$$

We want a generative model, so we would like to condition on the "generator," $\theta[j]$. We do this by applying the Bayes flip.

$$\sum_j \frac{\theta[j] \Pr(\#(x_j) = r \mid \theta[j]) \Pr(\theta[j])}{\sum_{j'} \Pr(\#(x_{j'}) = r \mid \theta[j']) \Pr(\theta[j'])} \tag{8}$$

We are assuming a uniform prior on $\theta[j]$ (i.e. $P(\theta[j]) = 1/m$), so the $\Pr(\theta[j])$ and $\Pr(\theta[j'])$ terms cancel.

$$\sum_j \frac{\theta[j] \Pr(\#(x_j) = r \mid \theta[j])}{\sum_{j'} \Pr(\#(x_{j'}) = r \mid \theta[j'])} \tag{9}$$

Now we rewrite the numerator and denominator in terms of the probability mass function for the binomial distribution.

$$\sum_j \frac{\theta[j] \binom{N}{r} \theta[j]^r (1 - \theta[j])^{N-r}}{\sum_{j'} \binom{N}{r} \theta[j']^r (1 - \theta[j'])^{N-r}} \tag{10}$$

Let $\mathbb{E}_{\text{in } N}[N_r]$ be the expected value of N_r given that we flipped N coins at each step of our experiment. Then we can rewrite the equation as

$$\frac{1}{\mathbb{E}_{\text{in } N}[N_r]} \sum_j \theta[j] \binom{N}{r} \theta[j]^r (1 - \theta[j])^{N-r}. \tag{11}$$

We cannot immediately rewrite the numerator in terms of $\mathbb{E}_{\text{in } N}[N_r]$ because of the extra $\theta[j]$ term. However, it is possible to write it in terms of $\mathbb{E}_{\text{in } N+1}[N_{r+1}]$. Observe that

$$\theta[j]\binom{N}{r}\theta[j]^r(1-\theta[j])^{N-r} \tag{12}$$

$$= \frac{N!}{(N-r)!r!}\theta[j]^{r+1}(1-\theta[j])^{N-r} \tag{13}$$

$$= \frac{r+1}{N+1}\frac{N+1}{r+1}\frac{N!}{(N-r)!r!}\theta[j]^{r+1}(1-\theta[j])^{(N+1)-(r+1)} \tag{14}$$

$$= \frac{r+1}{N+1}\frac{(N+1)!}{(N-r)!(r+1)!}\theta[j]^{r+1}(1-\theta[j])^{(N+1)-(r+1)} \tag{15}$$

$$= \frac{r+1}{N+1}\binom{N+1}{r+1}\theta[j]^{r+1}(1-\theta[j])^{(N+1)-(r-1)} \tag{16}$$

$$= \frac{r+1}{N+1}\mathbb{E}_{\text{in } N+1}[N_{r+1}]. \tag{17}$$

Therefore

$$\hat{\theta}(r) = \frac{r+1}{N+1}\frac{\mathbb{E}_{\text{in } N+1}[N_{r+1}]}{\mathbb{E}_{\text{in } N}[N_r]} \tag{18}$$

Now we plug in our observed values for $\mathbb{E}_{\text{in } N}[N_r]$ and $\mathbb{E}_{\text{in } N+1}[N_{r+1}]$. These are N_r and N_{r+1} respectively. This yields

$$\hat{\theta}(r) = \frac{1}{N+1}(r+1)\frac{N_{r+1}}{N_r}. \tag{19}$$

For large N, $\frac{1}{N+1} \approx \frac{1}{N}$, so finally we can rewrite the above equation as

$$\hat{\theta}(r) = \frac{1}{N}(r+1)\frac{N_{r+1}}{N_r}. \tag{20}$$

We will explore this approximation and an alternate explanation more in exercise 4.

This estimate has the nice property that

$$N_0\hat{\theta}(0) = N_0\frac{1}{N}\frac{N_1}{N_0} = \frac{N_1}{N}. \tag{21}$$

In other words, the total probability mass assigned to unseen events is the same as the relative occurrence of words that appear just once! This makes sense, because appearing zero times is not so different from appearing once in a relatively small sample.

One potential problem with this estimate is that it does not assign enough probability mass to events that occur a large number of times. For example, if r_M is the maximum number of times any word was observed, then

$$\hat{\theta}(r_M) = \frac{1}{N}(r_M+1)\frac{N_{r_M+1}}{N_{r_M}} = 0, \tag{22}$$

because $N_{r_M+1} = 0$ (i.e. there is no word that appeared $r_M + 1$ times).

4 Second Derivation

We will also examine another way to derive the Good-Turing estimation based on the concept of "deleted etimation" proposed by [3] (also see [4]). The idea behind this derivation is to divide W into two sets: the "train" set and the "heldout" set. The train set will be used to determine which terms occur r times, while the heldout set is used to estimate $\theta(r)$.

Let Heldcounts(r) be the number of times r-count items occur in the heldout set.

For example, let $X = \{\text{the}, \text{bad}, \text{dog}, \text{cat}\}$, W = the, cat, the, cat, the, dog, the, cat, the, dog, cat. The train set and the heldout set are partitioned in the following manner:
Train: the, cat, the cat
Heldout: the, dog, the, cat, the, dog, cat

In this scenario the Heldcounts are as follows:
Heldcounts$(0) = 2$. The 0-count items are "dog" and "bad"; "dog" occurs twice in the heldout set.
Heldcounts$(1) = 0$. There are no 1-count items in the train set.
Heldcounts$(2) = 5$. The 2-count items are "the" and "cat"; there are 5 of these items in the heldout set.

In order to estimate $\theta(r)$ for a given heldout set H, we can take $\hat{\theta}(r)$ values to be the max-likelihood estimates.

We introduce the non-normalized likelihood for a multinomial distribution

$$F(H) = C \prod_j \theta[j]^{\#_{ho}(x_j)}, \tag{23}$$

where C is the multinomial coefficient. Note that C is a constant, so we can remove it to get an equation that is equivalent under ranking. We will maximize this equation subject to the constraint

$$\sum_j \hat{\theta}[j] = 1 \tag{24}$$

With our definition of Heldcounts, we can rewrite the likelihood as

$$F(H) = \prod_r \theta(r)^{\text{Heldcounts}(r)} \tag{25}$$

and the constraint as

$$\sum_r \hat{\theta}(r) N_r = 1. \tag{26}$$

We will continue this derivation in the next lecture.

5 Exercises

1. This exercise is to test your understanding of the basic notation and concepts used in Good-Turing smoothing. Suppose we have the following

set of possible item types: $X = \{\text{apple}, \text{banana}, \text{carrots}, \text{dates}, \text{eggs}, \text{frogs}, \text{grapes}\}$. And suppose we have a sequence of N independent samples: $W =$ apple apple apple banana banana dates dates eggs eggs eggs frogs grapes grapes

(a) Calculate the empirical (observed relative-frequency) probabilities, $\theta_e(r)$.

(b) Calculate the Good-Turing probability estimates, $\hat{\theta}(r)$, based on W.

(c) Verify that $\sum_r \hat{\theta}(r) N_r = 1$.

2. (a) What would the Good-Turing estimates be for the following observed values: $N_0 = 1, N_1 = 0, N_2 = 1, N_3 = 0, N_4 = 1$?

   (b) What problems do you run into when you try to calculate these estimates? How might you correct these problems?

3. Show that $\hat{\theta}(0) = \hat{\theta}(1) = ... = \hat{\theta}(m) \propto 1/N$ if $N_r = s\frac{e^{-\lambda}}{r!}\lambda^r$ (i.e. N_r has Poisson form), where s is a positive constant. Note that $\frac{e^{-\lambda}}{r!}\lambda^r \leq 1$ because it is the density function of the Poisson distribution, so the s term acts as a scale factor that expands the range of N_r to the interval $[0, s]$. This exercise is based on a fact in [1].

4. In equation (20), we replaced the "normalization" term $\frac{1}{N+1}$ with $\frac{1}{N}$ to get

$$\hat{\theta}(r) = \frac{1}{N}(r+1)\frac{N_{r+1}}{N_r}.$$

   (a) Argue that $\frac{1}{N+1}$ is *not* the correct normalization for $(r+1)\frac{N_{r+1}}{N_r}$ by showing that if we use $\frac{1}{N+1}$ as the normalization, then we get an invalid probability distribution for the resulting word occurrence probability distribution.

   (b) What went wrong? How did our derivation produce the wrong normalization for $\hat{\theta(r)}$?

6 Solutions

1. $N_0 = 1$ (carrots)
   $N_1 = 1$ (frogs)
   $N_2 = 3$ (banana, dates, grapes)
   $N_3 = 2$ (apple, eggs)

   (a) $\theta_e(0) = 0/13$
   $\theta_e(1) = 1/13$
   $\theta_e(2) = 3/13$
   $\theta_e(3) = 2/13$

(b) $\hat{\theta}(0) = \frac{1}{13}(1)\frac{1}{1} = \frac{1}{13}$
$\hat{\theta}(1) = \frac{1}{13}(2)\frac{3}{1} = \frac{6}{13}$
$\hat{\theta}(2) = \frac{1}{13}(3)\frac{2}{3} = \frac{6}{39}$
$\hat{\theta}(3) = \frac{1}{13}(4)\frac{0}{2} = 0$

(c) $\hat{\theta}(0) + \hat{\theta}(1) + 3(\hat{\theta}(2)) + 2(\hat{\theta}(3)) = \frac{1}{13} + \frac{6}{13} + \frac{6}{13} + 0 = \frac{13}{13} = 1$

2. (a) $\hat{\theta}(0) = \frac{1}{6}(1)\frac{0}{1} = \frac{0}{6}$
$\hat{\theta}(1) = \frac{1}{6}(2)\frac{1}{0} =$ undefined
$\hat{\theta}(2) = \frac{1}{6}(3)\frac{0}{1} = \frac{0}{6}$
$\hat{\theta}(3) = \frac{1}{6}(4)\frac{1}{0} =$ undefined
$\hat{\theta}(4) = \frac{1}{6}(5)\frac{0}{1} = \frac{0}{6}$

(b) There are at least two problems with these estimates. First of all, there are undefined values if $r = 1$ or $r = 3$. This might not be a problem because one can argue that if there are no items that appear once, or if there are no items that appear 3 times in W, then there should be no probability associated with these values of r. However, there is another potential problem: the probabilities do not sum to 1. In real data samples, we can expect that there are some N_r values that are zero, so this could be a problem in practice.

This example suggests that there are problems that arise when using the Good-Turing estimation with a dataset that has some N_r values equal to 0. One way to fix this problem is to smooth the N_r counts so that they are all nonzero. For example, we can use linear regression to interpolate values for unobserved $\theta(r)$, as in [2].

3. We have

$$\hat{\theta}(r) = \frac{r+1}{N}\frac{se^{-\lambda}\lambda^{r+1}}{(r+1)!}\frac{r!}{se^{-\lambda}\lambda^{r}} \quad (27)$$

$$= \lambda/N. \quad (28)$$

Note that λ cannot be a free parameter, since there is only one value of λ that normalizes the probability distribution.

4. (a) Let $\hat{\theta}'(r)$ be the new value for $\hat{\theta}(r)$ that we get under this normalization scheme and similarly let $\hat{\theta}'[j]$ be the new value for $\hat{\theta}[j]$. Note that $\hat{\theta}'(r) = \frac{N}{N+1}\hat{\theta}(r)$ and thus $\hat{\theta}'[j] = \frac{N}{N+1}\hat{\theta}[j]$. We showed previously that if we use the Good-Turing estimate for $\hat{\theta}(r)$, then $\sum_j \hat{\theta}[j] = 1$. Therefore $\sum_j \hat{\theta}'[j] = \frac{N}{N+1}$ and thus θ' is not a valid probability distribution.

(b) We used N_{r+1} as an estimate for $\mathbb{E}_{\text{in } N+1}[N_{r+1}]$. However, N_{r+1} was based on observing N items rather than $N+1$, so really it is an approximation for $\mathbb{E}_{\text{in } N}[N_{r+1}]$. Therefore we need a correction.

7 References

1. I. J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika* 40: 237-264 (1953).

2. W. A. Gale. Good-Turing Smoothing Without Tears. *Journal of Quantitative Linguistics* 2: 217-237 (1995).

3. Frederick Jelink and Robert Mercer. Probability distribution estimation from sparse data. IBM Technical Disclosure Bulletin 28: 2591-2594 (1985).

4. Arthur Nádas. On Turing's formula for word probabilities. *IEEE Transactions on Acoustics, Speech and Signal Processing* ASSP-33(6):1414-1416, 1985.