EXTRA SLIDES ON SMOOTHING

Notation: N_c = Frequency of frequency c

- N_c = the count of things we've seen c times
- Sam I am I am Sam I do not eat

```
I   3
sam 2
am  2
do  1
not 1
eat 1
```

$N_1 = 3$

$N_2 = 2$

$N_3 = 1$

Good-Turing Smoothing Intuition

- You are fishing (a scenario from Josh Goodman), and caught:
  - 10 carp, 3 perch, 2 whitefish, 1 trout, 1 salmon, 1 eel = 18 fish
- How likely is it that next species is trout?
  - 1/18
- How likely is it that next species is new (i.e. catfish or bass)
  - Let's use our estimate of things-we-saw-once to estimate the new things.
  - 3/18 (because N_1=3)
- Assuming so, how likely is it that next species is trout?
  - Must be less than 1/18
  - How to estimate?

Good-Turing Calculations

$$P^*_{GT}(\text{things with zero frequency}) = \frac{N_1}{N} \qquad c^* = \frac{(c+1)N_{c+1}}{N_c}$$

Unseen (bass or catfish)

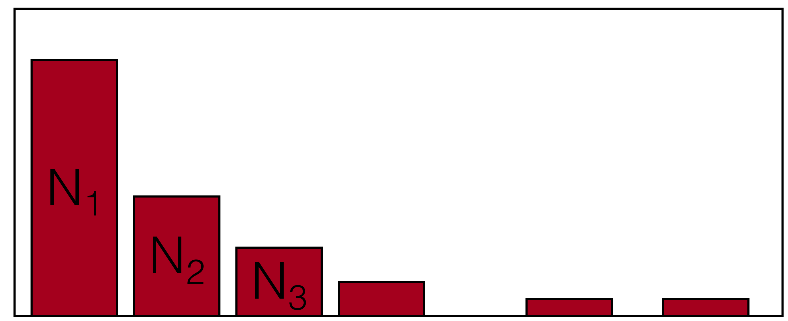
- $c = 0$:
- MLE $p = 0/18 = 0$
- P^*_{GT} (unseen) $= N_1/N = 3/18$

Seen once (trout)

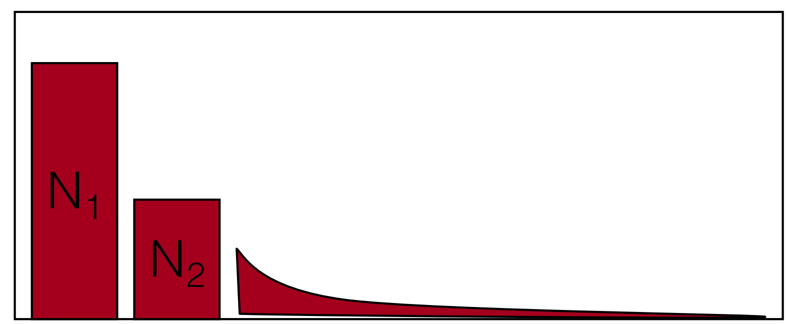
- $c = 1$
- MLE $p = 1/18$
- $C^*(\text{trout}) = 2 * N_2/N_1 = 2 * 1/3 = 2/3$
- $P^*_{GT}(\text{trout}) = 2/3 / 18 = 1/27$

Good-Turing Complications

- Problem: what about "the"? (say c=4417)
  - For small k, $N_k > N_k+1$
  - For large k, too jumpy, zeroes wreck estimates



  - Simple Good-Turing [Gale and Sampson]: replace empirical N_k with a best-fit power law once counts get unreliable



Good-Turing Numbers

- Numbers from Church and Gale (1991)
- 22 million words of AP Newswire

$$c^* = \frac{(c+1)N_{c+1}}{N_c}$$

- It sure looks like
  c* = (c - .75)

Count c	Good Turing c*
0	.0000270
1	0.446
2	1.26
3	2.24
4	3.24
5	4.22
6	5.19
7	6.21
8	7.24
9	8.25

Absolute Discounting

- Idea: observed n-grams occur more in training than they will later:

Count in 22M Words	Future c* (Next 22M)
1	0.448
2	1.25
3	2.24
4	3.23

- Absolute Discounting (Bigram case)
  - No need to actually have held-out data; just subtract 0.75 (or some d)

$$c^*(v,w) = c(v,w) - 0.75 \text{ and } q(w|v) = \frac{c^*(v,w)}{c(v)}$$

  - But, then we have "extra" probability mass

$$\alpha(v) = 1 - \sum_{w} \frac{c^*(v,w)}{c(v)}$$

  - Question: How to distribute α between the unseen words?

Katz Backoff

- Absolute discounting, with backoff to unigram estimates

$$c^*(v,w) = c(v,w) - \beta \qquad \alpha(v) = 1 - \sum_w \frac{c^*(v,w)}{c(v)}$$

- Define seen and unseen bigrams:

$$\mathcal{A}(v) = \{w : c(v,w) > 0\} \quad \mathcal{B}(v) = \{w : c(v,w) = 0\}$$

- Now, backoff to maximum likelihood unigram estimates for unseen bigrams

$$q_{BO}(w|v) = \begin{cases} \frac{c^*(v,w)}{c(v)} & \text{If } w \in \mathcal{A}(v) \\ \alpha(v) \times \frac{q_{ML}(w)}{\sum_{w' \in \mathcal{B}(v)} q_{ML}(w')} & \text{If } w \in \mathcal{B}(v) \end{cases}$$

- Can consider hierarchical formulations: trigram is recursively backed off to Katz bigram estimate, etc
- Can also have multiple count thresholds (instead of just 0 and >0)
- Problem?
  - Unigram estimates are bad predictors

Kneser-Ney Smoothing

- Better estimate for probabilities of lower-order unigrams!
  - Shannon game: I can't see without my reading___*Francisco* *glasses*___?
  - "Francisco" is more common than "glasses"
  - … but "Francisco" always follows "San"
- Instead of P(w): "How likely is w"
- $P_{continuation}(w)$: "How likely is w to appear as a novel continuation?
  - For each word, count the number of bigram types it completes
  - Every bigram type was a novel continuation the first time it was seen

$$P_{CONTINUATION}(w) \propto \left|\{w_{i-1} : c(w_{i-1}, w) > 0\}\right|$$

Kneser-Ney Smoothing

- How many times does w appear as a novel continuation:

$$P_{CONTINUATION}(w) \propto \left|\{w_{i-1} : c(w_{i-1}, w) > 0\}\right|$$

- Normalized by the total number of word bigram types

$$\left|\{(w_{j-1}, w_j) : c(w_{j-1}, w_j) > 0\}\right|$$

$$P_{CONTINUATION}(w) = \frac{\left|\{w_{i-1} : c(w_{i-1}, w) > 0\}\right|}{\left|\{(w_{j-1}, w_j) : c(w_{j-1}, w_j) > 0\}\right|}$$

Kneser-Ney Smoothing

- A frequent word (Francisco) occurring in only one context (San) will have a low continuation probability
- Replace unigram in discounting:

$$P_{KN}(w_i \mid w_{i-1}) = \frac{\max(c(w_{i-1}, w_i) - d, 0)}{c(w_{i-1})} + \lambda(w_{i-1}) P_{CONTINUATION}(w_i)$$

λ is a normalizing constant; the probability mass we've discounted

$$\lambda(w_{i-1}) = \frac{d}{c(w_{i-1})} \left| \{ w : c(w_{i-1}, w) > 0 \} \right|$$

the normalized discount

The number of word types that can follow w_{i-1}
= # of word types we discounted
= # of times we applied normalized discount

Kneser-Ney Smoothing: Recursive Formulation

$$P_{KN}(w_i \mid w_{i-n+1}^{i-1}) = \frac{\max(c_{KN}(w_{i-n+1}^{i}) - d, 0)}{c_{KN}(w_{i-n+1}^{i-1})} + \lambda(w_{i-n+1}^{i-1}) P_{KN}(w_i \mid w_{i-n+2}^{i-1})$$

$$c_{KN}(\bullet) = \begin{cases} count(\bullet) & \text{for the highest order} \\ continuationcount(\bullet) & \text{for lower order} \end{cases}$$

Continuation count = Number of unique single word contexts for •

Smoothing at Web-scale

- "Stupid backoff" (Brants *et al*. 2007)
- No discounting, just use relative frequencies

$$S(w_i \mid w_{i-k+1}^{i-1}) = \begin{cases} \dfrac{\text{count}(w_{i-k+1}^{i})}{\text{count}(w_{i-k+1}^{i-1})} & \text{if } \text{count}(w_{i-k+1}^{i}) > 0 \\ 0.4 S(w_i \mid w_{i-k+2}^{i-1}) & \text{otherwise} \end{cases}$$

$$S(w_i) = \frac{\text{count}(w_i)}{N}$$

[1]The name originated at a time when we thought that such a simple scheme cannot possibly be good. Our view of the scheme changed, but the name stuck.