Watch a video where Prof. Jurafsky (Stanford) explains Good-Turing smoothing (between 02:00 – 08:45)
Click on:
https://youtu.be/GwP8gKa-ij8

Or search for:"Good Turing video Jurafsky"
Answer briefly these 3 questions in a single file or text field

1. Estimate the prob. of catching next any new fish species, if you already got: 5 perch, 2 pike, 1 trout, 1 zander and 1 salmon?
Hint: estimate the prob of unseen things using the prob of things seen only once N1 / N

Number of catch fish N = 5 + 2 +1 + 1 + 1 = 10
We have N_c = frequency of frequency of things occurring c times
There are three fish occurring once => N1 = 3
There are one fish occurring twice => N2 = 1
There are one fish occurring 5 times => N5 = 1

The Good Turing probability of catching next any new fish species is:
P*GT(unseen) = N1/N = 3/10 = 0.3 (answer)

2. Estimate the prob. of catching a salmon next?
Hint: The counts must be smoothed.
The new count for things seen once is (c + 1)* N2 / N1

We know that salmon occurs once => c = 1

The smooth count of salmon is
C*(salmon) = (c + 1) * N_{c+1} / N_c = (1 + 1) * N2 / N1 = 2 * 1/3 = 2/3

The Good Turing probability of catching a salmon next is
P*GT(salmon) = C*(salmon)/N = (2/3)/10 = 2/30 = 0.067 (answer)

3. What may cause practical problems when applying Good-Turing smoothing for rare words in large text corpora?
What if N_c = 0 for some c?

Good-Turing smoothing can face issues when dealing with rare words in large text corpora. A significant challenge arises when Nc=0 for some count c. This situation occurs when there are no words or items observed exactly c times in the data, making it difficult to estimate the smoothed count for such frequencies. Without certain frequencies in the data, we cannot correctly estimate probabilities for rare or unseen items, especially in large and diverse datasets where rare words are more common.

Slides from the video

Reminder: Add-1 (Laplace) Smoothing

$$P_{Add-1}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) + 1}{c(w_{i-1}) + V}$$

Unigram prior smoothing

$$P_{Add-k}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) + m(\frac{1}{V})}{c(w_{i-1}) + m}$$

$$P_{\text{UnigramPrior}}(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i) + mP(w_i)}{c(w_{i-1}) + m}$$

Dan Jurafsky

Advanced smoothing algorithms

- Intuition used by many smoothing algorithms
  - Good-Turing
  - Kneser-Ney
  - Witten-Bell
- Use the count of things we've **seen once**
  - to help estimate the count of things we've **never seen**

Dan Jurafsky

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

$N_1 = 3$

$N_2 = 2$

$N_3 = 1$

Dan Jurafsky

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?

Dan Jurafsky

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*(trout) = 2 * N2/N1
    = 2 * 1/3
    = 2/3
  - P^*_{GT}(trout) = 2/3 / 18 = 1/27