

## Outline

- Follows Manning and Schutze: chapter 6.
- Equivalence classes for prediction.
- Good (and less good) estimators.
- (Combining models).

### Equivalence classes

- In order to count things you need a protocol about which things count as equivalent.
- Count the number of white and of red butterflies in a suburban garden (maybe collapse across species).
- Count the number of species visiting the garden (maybe want to know how many species in East Lothian).
- Count the number of words/lemmas/senses bigrams in your corpus (maybe really want to know about Business English).

## Prediction

- Question: I want to be healthy. Should I give up smoking?
- Answer 1: How can I possibly tell? I've studied a thousand cases, but none with your hair colour.
- Answer 2: I think hair colour is irrelevant. I'll assume you are equivalent to all the others and predict that your outcome will be the same as the others. Stop!
- We are interested in using a limited set of *features* to predict the outcome as well as we can.
- In Statistical NLP we want to do this with linguistic things. There are limits on how much data we can get. especially if (costly) human judgement is involved.

### **Pitfalls of prediction**

- If equivalence classes too coarse, can't make interesting predictions.
- If equivalence classes too fine, there will be many occasions where we should say "Don't know". If forced to a choice we may make lots of mistakes.
- Unigram word-frequency figures are (coarse) predictions of what mix of words will come next.
- Bigrams are more delicate predictors, which is both good and bad.

**Word frequency predictors**

## Maximum Likelihood and Discounting

- Maximum Likelihood Estimator:

$$p_{MLE}(W_i) = \frac{|W_i|}{\sum_j |W_j|}$$

- for later use define:
  - $N = \sum_j |W_j|$  *number of word tokens*
  - $B = \sum_j 1$  *number of word types*
- Not robust. Breaks for unseen words
- Discount observations. Reallocate some of probability of seen words to unseen counterparts.
- Many possible ways to do this.

## Laplace

- Laplace Estimator:

$$p_{Lap}(W_i) = \frac{|W_i| + 1}{N + B}$$

- Let  $\mu = N/(N + B)$  then:

$$p_{Lap}(W_i) = \mu \frac{|W_i|}{N} + (1 - \mu) \frac{1}{B}$$

- This is a *mixture* of MLE and a uniform distribution.
- For bigrams, Church and Gale show that under this regime 46.5% of the probability is allocated to unseen bigrams. Which is far too much.

**Lidstone**

- Don't add one, add  $\lambda$  (usually  $\lambda < 1$ ),
- Hardy - Lidstone Estimator

$$p_{Lid}(W_i) = \frac{|W_i| + \lambda}{N + B\lambda}$$

- Let  $\mu = N/(N + B\lambda)$  then:

$$p_{Lid}(W_i) = \mu \frac{|W_i|}{N} + (1 - \mu) \frac{1}{B}$$

- Still a mixture of MLE and uniform but we can now choose the value of  $\lambda$  as we see fit.
- If  $\lambda = \frac{1}{2}$  It is called the Jeffreys-Perks law.
- How to get correct  $\lambda$ ?
- $p_{Lid}$  is a linear function of MLE. Bad fit for low frequencies.

### Held out estimation

- Idea 1: keep back some of the text, see how often events in first part occur in second part.
  - $f_1(\text{item})$  frequency in first part
  - $f_2(\text{item})$  frequency in held-out part

- Idea 2: Pool all items having frequency  $r$  in first half.
- Measure counts of these items in held out part

$$C_r = \sum_{\text{Items with } f_1=r} f_2$$

- Divide by the number of items in the pool  $N_r$ , getting the average frequency of these items.
- Estimate the probability of items in the pool as

$$\frac{C_r}{N_r N}$$

- Hold out gives a value even to items whose count in the first part of the corpus is zero.
- If you let this method see the test data, it has an unfair advantage over the other methods. This is the best any method can do. Measure how close the others get without seeing the test data.
- That's how we find out that that the Laplace M-estimate was wrong to give 46.5% to unseen events.
- If used *without* peeking at the test data, must split training data, to get hold out, which seems a waste.

### Deleted Estimation

- There's a neat extension to held out estimation called *deleted estimation* which essentially does held out twice, swapping the role of main and held out portions, then averages the results in an appropriate way.
- Dividing corpus in half at middle is risky, better to do something like even and odd sentences.
- Deleted estimation works much better than Laplace, but still overestimates the probability of unseen events, and takes too much away from once-only (*hapax*) events.

## Good-Turing Estimation

- Different idea: hypothesise that items are binomially distributed.
- If that is true, there will on average, be a particular relation between the number of items having frequency  $r$  and the number of items having frequency  $r + 1$ .
- This fact can be converted into an expression for an adjusted frequency (see Church and Gale, *Computer Speech and Language* 5,19–54, 1991 for a proof).
- The expression is

$$r^* = (r + 1) \frac{\text{Estimate of number of items with frequency } r + 1}{\text{Estimate of number of items with frequency } r}$$

- The probability given to all unseen items is  $\frac{N_1}{N}$ . If we know the number of such items (we do with bigrams) can share it out either evenly or otherwise.

### Estimating $N_r$ for Good-Turing

- We still have the task of making sure that our estimates for the  $N_r$  above are reasonable.
- We can't use raw counts would break down at high frequencies
  - $p(\text{topword}) = 0!$
  - Words with measured frequency 32 and 35 might exist, but none with frequency 33 or 34.
- So don't use GT above some threshold frequency.
- Alternative is to fit a smooth function (e.g. a line) to the raw  $N_r$  data, then use them to form adjusted frequencies.
- This does really well for big corpora and large spaces of possible items.
- See Church and Gale for Enhanced Good-Turing, which frees you from the need of giving the same estimate to all items having same frequency in training data.