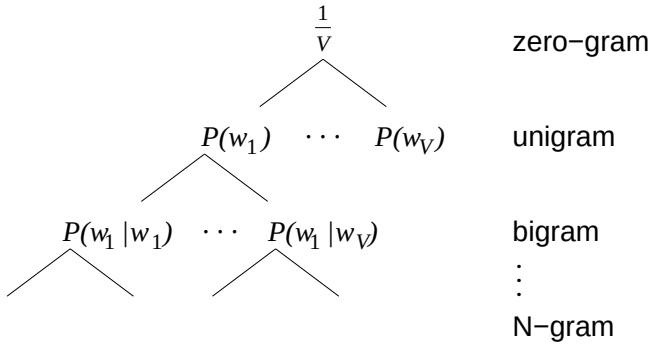
UNIVERSITY OF SURREY
Centre for Vision Speech & Signal Processing
University of Surrey, Guildford GU2 7XH.

# Language modelling

Dr Philip Jackson

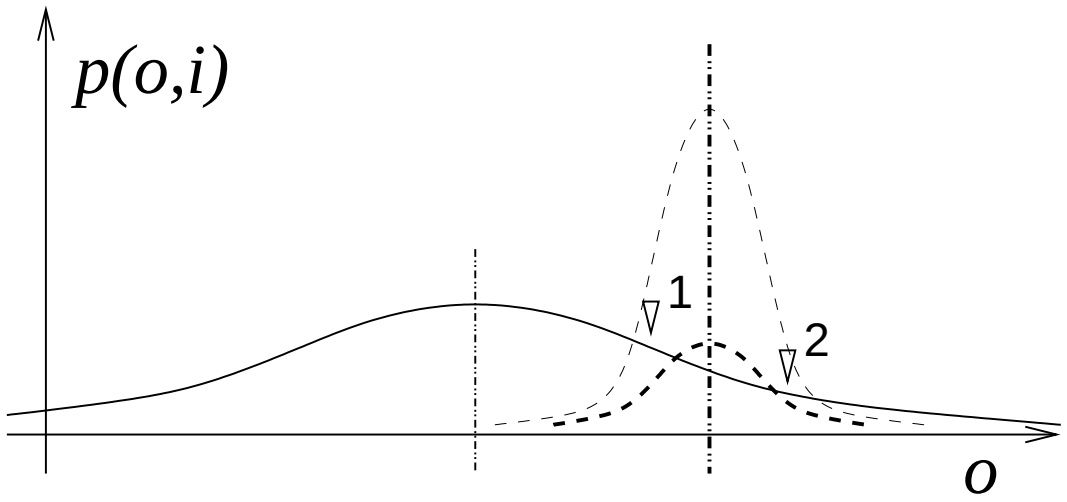
- Bayes and prior probabilities
- Language models
  - Bigram language model
  - $N$-grams
  - Smoothing & backoff



# Bayesian inference

## Value of prior knowledge

Let us suppose there are two conditional pdfs, as follows:



## Interpretation of Bayes' theorem

Consider a word $w$ and set of observations $\mathcal{O}$,

$$P(w|\mathcal{O}) = \frac{P(\mathcal{O}|w)\,P(w)}{P(\mathcal{O})}, \tag{1}$$

where $\mathcal{O}$ denotes a series of measured or observed data, and $w$ describes a model of the word.

$P(w|\mathcal{O})$ is the **posterior probability**

$P(\mathcal{O}|w)$ is the **likelihood**

$P(w)$ is the **prior probability**

$P(\mathcal{O})$ is the **evidence**

# Language modelling

## Language model definition

For the task of recognizing a word sequence $W = \boldsymbol{w}_1^n$, we seek

$$\hat{W} = \arg\max_W \underbrace{P(\mathcal{O}|W)}_{\substack{\text{acoustic} \\ \text{model}}} \underbrace{P(W)}_{\substack{\text{language} \\ \text{model}}} \tag{2}$$

where $P(W)$ is referred to as the **language model**. Now, using the chain rule, gives

$$\begin{aligned} P(\boldsymbol{w}_1^n) &= P(w_1)P(w_2|w_1)P(w_3|\boldsymbol{w}_1^2)\ldots P(w_n|\boldsymbol{w}_1^{n-1}) \\ &= \prod_{k=1}^{n} P\left(w_k|\boldsymbol{w}_1^{k-1}\right). \end{aligned} \tag{3}$$

We could just assume $P(w_k|\boldsymbol{w}_1^{k-1}) \approx P(w_k)$ to simplify the expression, but we would lose all information about word order and phrasing.

## Bigram language model

As with the acoustic model, we can simplify probability calculation by assuming that the current word depends only on the previous one:

$$P\left(w_k|\boldsymbol{w}_1^{k-1}\right) \approx P\left(w_k|w_{k-1}\right), \tag{4}$$

e.g.,

$$P(\text{rabbit}|\text{Just the other day I saw a}) \approx P(\text{rabbit}|\text{a}).$$

This yields the probability of a word sequence in terms of **bigrams**:

$$P(\boldsymbol{w}_1^n) = \prod_{k=1}^{n} P\left(w_k|w_{k-1}\right), \tag{5}$$

and we define $w_0 = \texttt{START}$ to begin an utterance.

## N-gram language models

Rather than taking two words at a time, the amount of context can be increased by considering three words to give the **trigram** probability,

$$P\left(w_k|\boldsymbol{w}_1^{k-1}\right) \approx P\left(w_k|\boldsymbol{w}_{k-2}^{k-1}\right), \tag{6}$$

four words to give **quadrigrams**,

$$P\left(w_k|\boldsymbol{w}_1^{k-1}\right) \approx P\left(w_k|\boldsymbol{w}_{k-3}^{k-1}\right), \tag{7}$$

or a sequence of $N$ words to give the **N-gram**:

$$P\left(w_k|\boldsymbol{w}_1^{k-1}\right) \approx P\left(w_k|\boldsymbol{w}_{k-N+1}^{k-1}\right). \tag{8}$$

## Language model training

Bigram training is based on the relative frequency of each pair of words in the vocabulary $V$:

$$P\left(w_k|w_{k-1}\right) = \frac{C\left(w_{k-1}w_k\right)}{\sum_{\kappa\in V} C\left(w_{k-1}w_\kappa\right)}$$

$$= \frac{C\left(w_{k-1}w_k\right)}{C\left(w_{k-1}\right)} \tag{9}$$

More generally, we can write a similar expression for the $N$-gram, considering a sequence of $N$ words:

$$P\left(w_k|\boldsymbol{w}_{k-N+1}^{k-1}\right) = \frac{C\left(\boldsymbol{w}_{k-N+1}^{k}\right)}{C\left(\boldsymbol{w}_{k-N+1}^{k-1}\right)} \tag{10}$$

## Language model smoothing

- Add-one smoothing

$$C^*(\boldsymbol{w}) = C(\boldsymbol{w}) + 1$$

- Witten-Bell discounting

$$P^*(\boldsymbol{w}_{\text{unseen}}) \approx \frac{T}{N} = \frac{\#\text{ types}}{\#\text{ words}}$$

- Good-Turing discounting, from examples seen once:

$$P_0^* = \frac{n_1}{N}$$

$$P_r^* = (1 - P_0^*)\frac{k_r}{N^*} \qquad \text{where } k_r = (r+1)\frac{n_{r+1}}{n_r}$$

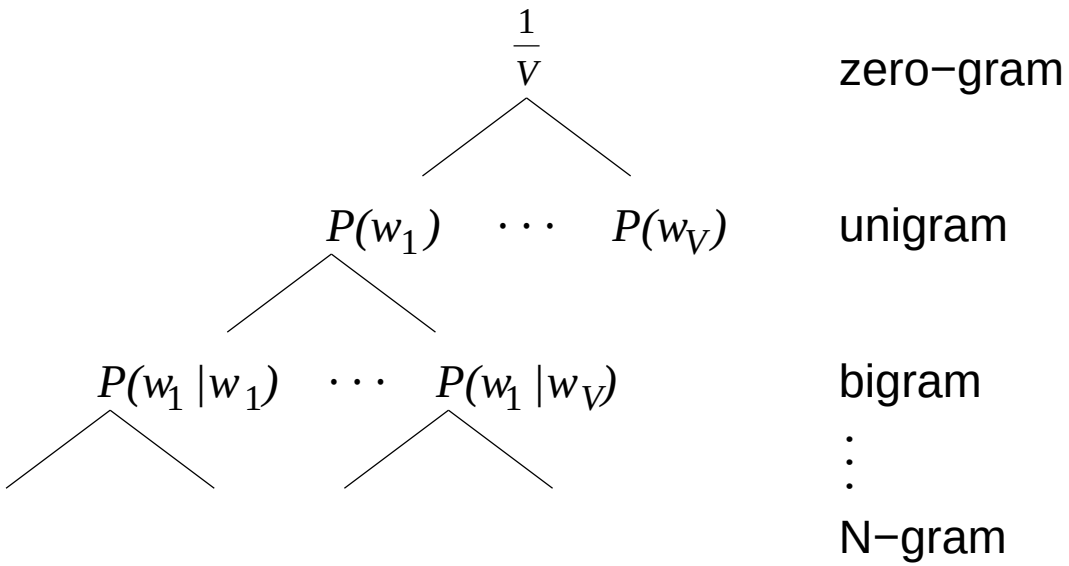
# Language model backoff

- Backoff

$$P^*(w_3|w_1, w_2) \approx \hat{P}(w_3|w_2)$$

- Deleted interpolation

$$P^*(w_3|w_1, w_2) = a_3\hat{P}(w_3|w_1, w_2) + a_2\hat{P}(w_3|w_2) + a_1\hat{P}(w_3) + a_0\frac{1}{V}$$



## Evaluating language models

Perplexity is used to compare the ability of language models to represent the likelihoods of a given word sequence, and it is based on the idea of **entropy**, measured in bits:

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \log_2 P(x) \tag{11}$$

It can be shown that the cross entropy between a model $m$ and a test sequence $X$ is

$$H(X, m) = \lim_{n \to \infty} -\frac{1}{n} \log_2 P(\boldsymbol{w}_1^n | m) \tag{12}$$

The related quantity **perplexity** is expressed directly:

$$J(X) = 2^{H(X)} \tag{13}$$

# Example of bigram LM for connected digits

Using phone number counts from my personal contacts:

| | START | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| START | .00 | .43 | .08 | .09 | .05 | .06 | .06 | .06 | .05 | .06 | .05 |
| 0 | .00 | .04 | .28 | .13 | .06 | .03 | .05 | .04 | .26 | .07 | .04 |
| 1 | .00 | .12 | .12 | .11 | .10 | .10 | .09 | .08 | .10 | .09 | .09 |
| 2 | .00 | .12 | .12 | .11 | .10 | .10 | .09 | .08 | .10 | .09 | .09 |
| 3 | .00 | .12 | .12 | .11 | .10 | .10 | .09 | .08 | .10 | .09 | .09 |
| 4 | .00 | .12 | .12 | .11 | .10 | .10 | .09 | .08 | .10 | .09 | .09 |
| 5 | .00 | .12 | .12 | .11 | .10 | .10 | .09 | .08 | .10 | .09 | .09 |
| 6 | .00 | .12 | .12 | .11 | .10 | .10 | .09 | .08 | .10 | .09 | .09 |
| 7 | .00 | .12 | .12 | .11 | .10 | .10 | .09 | .08 | .10 | .09 | .09 |
| 8 | .00 | .12 | .12 | .11 | .10 | .10 | .09 | .08 | .10 | .09 | .09 |
| 9 | .00 | .12 | .12 | .11 | .10 | .10 | .09 | .08 | .10 | .09 | .09 |

Bigram: $$P(w_2|w_1) = \frac{C(w_1, w_2)}{C(w_1)}$$

Back-off: $$P(w_2|w_1) \approx \frac{C(w_2)}{N}$$

# Language modelling summary

- Framework for priors based on a language model
- Language models
  - Bigrams, trigrams and $N$-grams
  - Smoothing & backoff
  - Evaluating perplexity of a LM
- Applications of $N$-grams
  - speech recognition
  - statistical machine translation
  - spell checking
  - entity detection
  - data mining and search engines