

# Statistical Linguistics

Why to count, what to count and how  
to count it

# Applications of statistical thinking

Language Models: predict the next word

Applications: act on limited information

Linguistics<sub>1</sub>: prefer economical theories

Linguistics<sub>2</sub>: explain natural language phenomena.

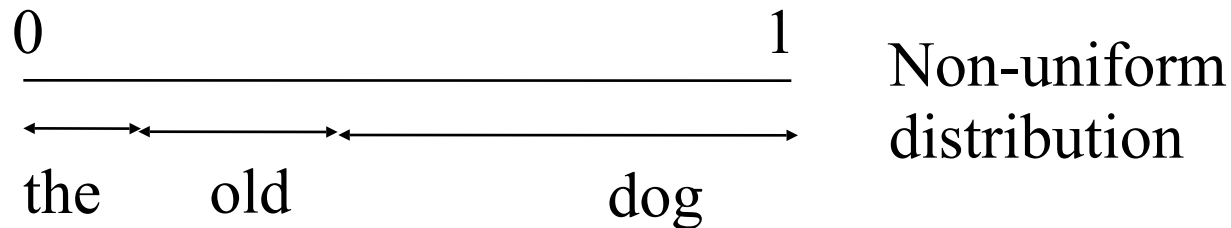
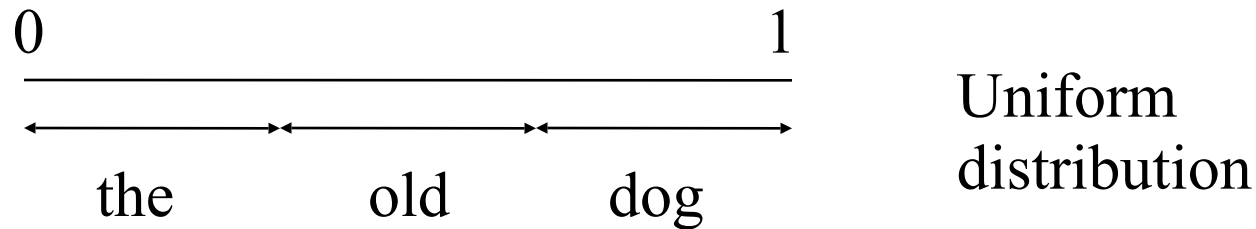
Linguistics<sub>3</sub>: explain language acquisition.

# Probability and language models

- Language Models: predict next word on basis of knowledge or assumptions.
- Probability theory developed as a way of reasoning about uncertainty and (especially) games of chance.

# Probability distributions

- Suppose there are just three words: “the”, “old” and “dog”.



# Events, Trials and Outcomes

- We sometimes say that an **event** (Heads), is the outcome of a **trial** (Tossing a coin).
- Now imagine a **series** of trials (repeatedly tossing a coin). Plausible that the outcome of trial  $n+1$  will be unaffected by that of trial  $n$ . Earlier events in the sequence do not affect later events.

# Propositional variables

- *Variables* in propositional logic formalise the idea of a truth value.
- They acquire their meaning from their relationships with other variables
- These relationships are expressed using connectives with evocative names like AND, OR and NOT
- Or, as Charles Peirce knew, we can just use a NAND gate and have the full expressive power of Boolean logic

# Propositional variables

- For convenience, we often choose to use a wider range of connectives than just NAND, because words like “or”, “and” and “implies” are more familiar.
- In essence, what we are able to do is to make statements about the relationships between truths, formalizing things like “If rain would have made the grass wet, and the grass isn’t wet, then it didn’t rain”

# Random variables

- *Random variables* formalise the idea of a trial.
- Random variables represent what you know about the trial before you have seen its outcome.
- Before you draw a card from the deck, you know that there are fifty-two cards, four suits, two colours, twelve face cards, ...

# Notation for probabilities

- $P(X=x_i)$  a probability (number) for  $x_i$
- $P(x_i)$  an abbreviation for the above
- $P(X)$ , or either of above to mean the function that assigns a value to each  $x_i$
- $|X=x_i|$  for the number of times  $x_i$  occurs.

# Conditional probabilities

- Conan-Doyle does not play dice
  - “Holmes” follows every use of “Sherlock”.
- The formal statement of this fact is that the conditional probability of the  $n$ th word being “Holmes” if the  $n-1$ th is “Sherlock” appears to be 1 for Conan-Doyle stories.
- $P(W_n = \text{holmes} \mid W_{n-1} = \text{sherlock}) = 1$

# Joint events

- *Joint event* of the  $n-1$ th word being “Sherlock” and the  $n$ th “Holmes”.
- $P(W_{n-1} = \text{sherlock}, W_n = \text{holmes})$
- Know identity of the next word when we have seen the “Sherlock”, so
- $P(W_n = \text{holmes}, W_{n-1} = \text{sherlock}) = P(W_{n-1} = \text{sherlock})$

# Complement of an event

- If event  $x$  does not happen, then the event  $\text{complement}(x)$  does happen.  
 $\text{complement}(\text{Dice}=2)$  is  $\text{Dice} = x$  s.t.  $x$  in  $1,3,4,5,6$

# Decomposing joint events

- In general, for any pair of words  $w_k, w_{k'}$ , we will have:
- $P(W_n = w_k, W_{n+1} = w_{k'}) = P(W_n = w_k) P(W_{n+1} = w_{k'} | W_n = w_k)$
- which is usually written more compactly:
- $P(w_n^k, w_{n+1}^{k'}) = P(w_n^k) P(w_{n+1}^{k'} | w_n^k)$

# Sidelight

- $P(w_n^k, w_{n+1}^{k'}) = P(w_n^k) P(w_{n+1}^{k'} | w_n^k)$
- This expression has 4 subscripts and 4 superscripts
- Also two random variables and a probability function.
- In order to keep this stuff straight, concentration and practice are essential.

# What is probability?

- Probability is like logic, but with real numbers instead of True and False

# What is plausibility?

- Plausibility is like logic, but with real numbers instead of True and False
- Assume, with Cox (1946), that we want the following things
  - The real number associated with an event depends on the information we have about it
  - Common sense: works like Aristotle's logic when Aristotle's logic has something to say
  - Consistency: if there are several different ways of getting a plausibility, must yield same result.

# Plausibility axioms

- The plausibility of a proposition is related to the plausibility of the complement. Double complementation leaves plausibility unchanged. There is a probability complementation function  $f$  s.t  $f(f(x)) = x$
- Plausibilities combine by an associative, binary operation over reals. All such operations are isomorphic (as it turns out) to multiplication of numbers in the range 0..1

# Plausibility axioms

- An axiom reflecting consistency, amounting to  $P(A,B|C) = P(A|C)P(B|A,C) = P(B|C)P(A|B,C)$
- If you have these and add  $P(\text{False})=0$ ,  $P(\text{True}) = 1$  you actually have probability
- Accept no substitutes.

# Pitfalls

- Just because Holmes is the only word that follows Sherlock, it need not be that Holmes is always preceded by Sherlock.

# Bayes' theorem

- $P(w_n^k, w_{n+1}^{k'}) = P(w_n^k) P(w_{n+1}^{k'} | w_n^k) = P(w_{n+1}^{k'}) P(w_n^k | w_{n+1}^{k'})$
- Divide through by  $P(w_n^k)$
- $P(w_{n+1}^{k'} | w_n^k) = \frac{P(w_{n+1}^{k'}) P(w_n^k | w_{n+1}^{k'})}{P(w_n^k)}$
- Which is an instance of Bayes' theorem
- $P(A|B) = \frac{P(A) P(B|A)}{P(B)}$

# Medical diagnosis

The doctors problem:  $P(S,C)$  vs  $P(S,P)$

Causal Information:  $P(S|C) = P(S|P) = 1$

Base Rates:  $P(P) = 10^{-6}$   $P(C) = 0.25$   $P(S) = 0.33$

Wants:  $P(C|S)$  and  $P(P|S)$

# Bayes' rule applied

$$P(P|S) = (P(P) \times P(S|P))/P(S)$$

In this case  $(10^{-6} \times 1)/0.33 = 3 \times 10^{-6}$

the doctor probably wouldn't panic.

In an epidemic the prior might change dramatically, affecting the outcome.

The prior dominates the posterior.

# Being Bayesian

Posterior  $\propto$  Prior  $\times$  Likelihood

$P(L|X) \propto P(L) \times P(X|L)$

Grammar inference = Prior beliefs about  
possible grammars  $\times$  Language model

# Linguistic diagnosis: Language Identification

e preebas bioquimica  
man immunodeficiency  
faits se sont produi

# Bad techniques for language identification

- Common words: needs longish test data.
- Proper names: especially bad, since very numerous, and highly likely to appear in documents in another language.
- Unique letter pairs or sequences: OK sometimes, but doesn't weight evidence by frequency

# Tutorial Paper by Ted Dunning

Dunning asks the following questions:

Q: How simple can the program be?

A: Small program based on statistical principles

Q: What does it need to learn?

A: No hand-coded linguistic knowledge is needed.

Only training data plus the assumption that texts are somehow made of bytes.

Q: How much training data needed?

A: A few thousand words of sample text from each language suffices. Ideally about 50 Kbytes

# More questions

Q: How much test data?

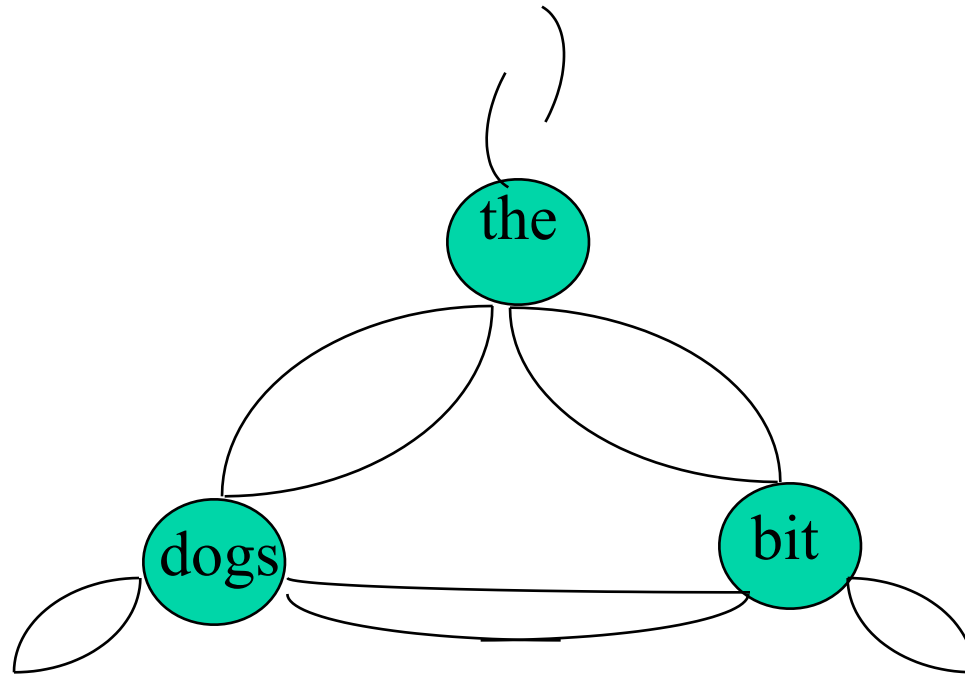
A: 10 characters work, 500 characters very well.

Q: Can it generalise?

A: If trained on French, English and Spanish,  
thinks German is English.

# Markov Models

- States and transitions (with probabilities)



# Tabular form of Markov models

- Transition Matrix(A)

The Dogs Bit

The 0.33 0.33 0.33

Dogs 0.33 0.33 0.33

Bit 0.33 0.33 0.33

- Start with initial probabilities  $\underline{p}(0)$

The 0.7

Dogs 0.2

Bit 0.1

# Using Markov models

- Choose initial state from  $\underline{p}(0)$
- Choose transition from relevant row of  $\underline{A}$
- Repeat ad lib
- Simple probabilistic generator for sequences of words. Yields sequence and its probability.
- Same can be done with characters, giving a probabilistic generator for character strings.

# Higher order Markov models

- If you want to take account of more context, you can re-draw the machine so that each state is labelled with a longer fragment of the context.
- Example

# Randomly generated strings from Markov models

0 hm 1 imuando~doc ni leotLs Aiqe1pdt6cf tlc.teontctrrdsxo~es loo oil3s  
1 ~ a meston s oflas n, ~ 2 nikexihiomanotrmo s,~125 0 3 1 35 fo there  
2 s ist ses anat p sup sures Alihows raaial on terliketicany of prelly  
3 approduction where. If the linal wate probability the or likelihood  
4 sumed normal of the normal distribution. Gale, Church, Willings. This  
5 ~k sub 1} sup {n-k} .EN where than roughly 5. This agreemented by th  
6 these mean is not words can be said to specify appear McDonald. 1989

# Bayesian Decision Rules

- Spanish and English as diseases causing the symptom “immunotechno”
- Compare  $P(\text{immunotechno}, \text{Spanish})$  with  $P(\text{immunotechno}, \text{English})$
- If we want we can compare  $P(\text{immunotechno} | \text{Spanish})p(\text{Spanish})$  with  $P(\text{immunotechno} | \text{English})p(\text{English})$

# Bayesian Decision Rules2

- Assume that  $P(\text{Spanish}) = P(\text{English})$ .
- So compare  $P(\text{immunotechnol}|\text{Spanish})$  with  $P(\text{immunotechnol}|\text{English})$
- To do this, we need a model which will generate character strings from data.
- A character-based Markov model is what is needed

# Character Markov Models

- $P(s^0 \dots s^n) = P(s^0)P(s^1 | s^0) P(s^2 | s^0, s^1)P(s^3 | s^0, s^1, s^2) \dots P(s^n | s^0 \dots s^{n-1})$
- The above is exact, but impractical, because we don't have sufficient data for reliable estimates.
- Approximate with  $P^*(s^0 \dots s^n) = P(s^0)P(s^1 | s^0) P(s^2 | s^0, s^1)P(s^3 | s^1, s^2) \dots P(s^n | s^{n-2}, s^{n-1})$

# Character Markov Models

- That was for character-level Markov models of order 2, which corresponds to a *trigram* model. Word-level trigram models are common.
- For the language identification application we actually use character 4-grams.

# Probability estimation

- We need transition probabilities  $p(s_4 | s_1, s_2, s_3)$
- Obvious way to proceed is to collect  $|s_1, s_2, s_3, s_4|$  and  $|s_1, s_2, s_3|$  then divide.
- This gives zero if  $|s_1, s_2, s_3, s_4| = 0$ , undefined if  $|s_1, s_2, s_3| = 0$

# Probability estimation 2

- It's a great model for the training set, but fails catastrophically in the real world.
- There may be  $k+1$ -grams in the test data which are absent from the training data.

# Probability estimation 3

- By bad luck may be a  $k+1$  gram in the training data for one language, even though it is in fact rare in all the languages.
- If this happens, all strings containing that  $k+1$ -gram will be judged to be from that language, because all the others will be 0.
- maximum likelihood estimator is too brittle.

# Smoothing

- $P((s_4 | s_1, s_2, s_3, \text{Lang}) = (|s_1, s_2, s_3, s_4| + 1) / (|s_1, s_2, s_3| + M)$
- this is more stable, can be seen as mixing in a small measure of uniform distribution into the empirical data
- *Discounts* the data, but also prevents zeros
- Laplace's M-estimate

# Results

- In a binary choice between English and Spanish strings drawn from a bilingual corpus, an accuracy of 92% from 20 bytes of test data and 50Kbytes of training data,
- improving to about 99.9% when 500 bytes of test data are allowed.