Introduction to Probability Theory 3

Clayton Greenberg
CoLi, CS, MMCI, LSV, CRC 1102 (IDeaL) B4

October 27, 2014

Schedule

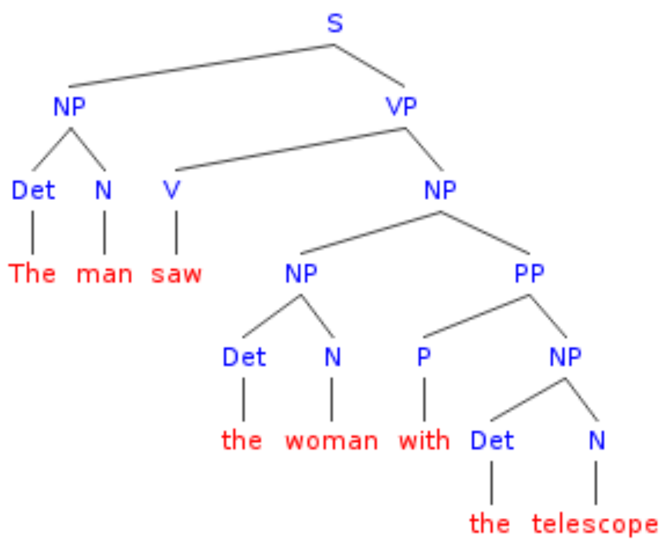
22.10.2014	Calculate the probability of a given parse
23.10.2014	Solve the medical test Bayes' Rule problem
27.10.2014	Create a code for simplified Polynesian
29.10.2014	Identify types of machine learning problems
31.10.2014	Find a regression line for 2D data

Wrap-up PCFG exercise

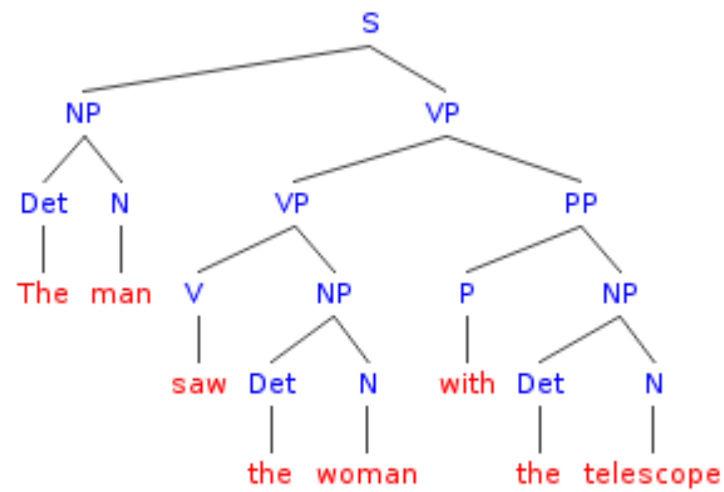
S → NP VP (1.0)
NP → Det N (0.8)
NP → NP PP (0.2)
PP → P NP (1.0)
VP → V NP (0.7)
VP → VP PP (0.3)

Det → “the” (1.0)
N → “man” (0.35)
N → “woman” (0.35)
N → “telescope” (0.2)
N → “hill” (0.1)
V → “saw” (1.0)
P → “with” (0.75)
P → “on” (0.25)

Our favorite sentence



TREE_1



TREE_2

Probability of the trees

S → NP VP (1.0)
NP → Det N (0.8)
VP → V NP (0.7)
NP → NP PP (0.2)
NP → Det N (0.8)
PP → P NP (1.0)
NP → Det N (0.8)
Det → “the” (1.0)
N → “man” (0.35)
V → “saw” (1.0)
Det → “the” (1.0)
N → “woman” (0.35)
P → “with” (0.75)
Det → “the” (1.0)
N → “telescope” (0.2)
Product: 0.00132

S → NP VP (1.0)
NP → Det N (0.8)
VP → VP PP (0.3)
VP → V NP (0.7)
NP → Det N (0.8)
PP → P NP (1.0)
NP → Det N (0.8)
Det → “the” (1.0)
N → “man” (0.35)
V → “saw” (1.0)
Det → “the” (1.0)
N → “woman” (0.35)
P → “with” (0.75)
Det → “the” (1.0)
N → “telescope” (0.2)
Product: 0.00198

Probability of the string

p("the man saw the woman with the telescope" | S)

= TREE_1 + TREE_2

= 0.00132 + 0.00198

= 0.00330

Surprisal (8 words) = 8.243 bits

Probability of a tree (PCFG) = product of its rules

Probability of a string (PCFG) = sum of its trees

Probability of the trees

S → NP VP (1.0)	S → NP VP (1.0)	S → NP VP (1.0)	S → NP VP (1.0)	S → NP VP (1.0)
NP → Det N (0.8)	NP → Det N (0.8)	NP → Det N (0.8)	NP → Det N (0.8)	NP → Det N (0.8)
VP → VP PP (0.3)	VP → VP PP (0.3)	VP → VP PP (0.3)	NP → NP PP (0.2)	NP → NP PP (0.2)
VP → VP PP (0.3)	NP → NP PP (0.2)	NP → NP PP (0.2)	NP → NP PP (0.2)	NP → NP PP (0.2)
VP → V NP (0.7)	VP → V NP (0.7)	VP → V NP (0.7)	VP → V NP (0.7)	VP → V NP (0.7)
PP → P NP (1.0)	PP → P NP (1.0)	PP → P NP (1.0)	PP → P NP (1.0)	PP → P NP (1.0)
NP → Det N (0.8)	NP → Det N (0.8)	NP → Det N (0.8)	NP → Det N (0.8)	NP → Det N (0.8)
NP → Det N (0.8)	NP → Det N (0.8)	NP → Det N (0.8)	NP → Det N (0.8)	NP → Det N (0.8)
PP → P NP (1.0)	PP → P NP (1.0)	PP → P NP (1.0)	PP → P NP (1.0)	PP → P NP (1.0)
NP → Det N (0.8)	NP → Det N (0.8)	NP → Det N (0.8)	NP → Det N (0.8)	NP → Det N (0.8)
Det → “the” (1.0)	Det → “the” (1.0)	Det → “the” (1.0)	Det → “the” (1.0)	Det → “the” (1.0)
N → “man” (0.35)	N → “man” (0.35)	N → “man” (0.35)	N → “man” (0.35)	N → “man” (0.35)
V → “saw” (1.0)	V → “saw” (1.0)	V → “saw” (1.0)	V → “saw” (1.0)	V → “saw” (1.0)
Det → “the” (1.0)	Det → “the” (1.0)	Det → “the” (1.0)	Det → “the” (1.0)	Det → “the” (1.0)
N → “woman” (0.35)	N → “woman” (0.35)	N → “woman” (0.35)	N → “woman” (0.35)	N → “woman” (0.35)
P → “with” (0.75)	P → “with” (0.75)	P → “with” (0.75)	P → “with” (0.75)	P → “with” (0.75)
Det → “the” (1.0)	Det → “the” (1.0)	Det → “the” (1.0)	Det → “the” (1.0)	Det → “the” (1.0)
N → “telescope” (0.2)	N → “telescope” (0.2)	N → “telescope” (0.2)	N → “telescope” (0.2)	N → “telescope” (0.2)
P → “on” (0.25)	P → “on” (0.25)	P → “on” (0.25)	P → “on” (0.25)	P → “on” (0.25)
Det → “the” (1.0)	Det → “the” (1.0)	Det → “the” (1.0)	Det → “the” (1.0)	Det → “the” (1.0)
N → “hill” (0.1)	N → “hill” (0.1)	N → “hill” (0.1)	N → “hill” (0.1)	N → “hill” (0.1)
Product: 0.0000119	Product: 0.00000790	Product: 0.00000790	Product: 0.00000527	Product: 0.00000527

Probability of the string

p("the man saw the woman with the telescope on the hill" | S)

= TREE_1 + TREE_2 + TREE_3 + TREE_4 + TREE_5

= 0.0000119 + 0.00000790 + 0.00000790 + 0.00000527 + 0.00000527

= 0.0000382

Surprisal (11 words) = 14.676 bits

Probability of a tree (PCFG) = product of its rules

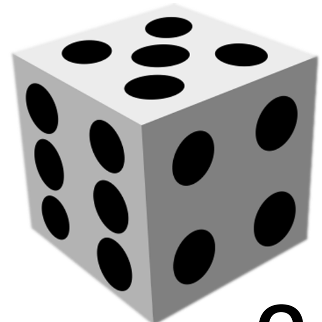
Probability of a string (PCFG) = sum of its trees

Green statement review

- probability = what you want / what is possible
- “and” = * (times) [if independent]
- “or” = + (plus) [if mutually exclusive]
- surprisal = the negative logarithm of probability
- conditional = joint / normalizer
- chain rule: joint = conditional of last * joint of rest
- probability of a tree (PCFG) = product of its rules
- probability of a string (PCFG) = sum of its trees
- Bayes’ rule: posterior = likelihood * prior / normalizer

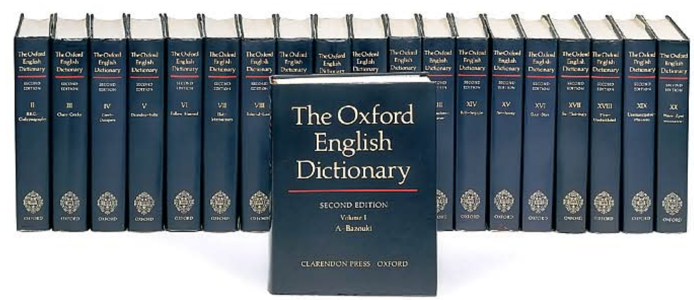
Probabilistic outcomes

$\Omega = \{H, T\}$



$\Omega = \{1, 2, 3, 4, 5, 6\}$

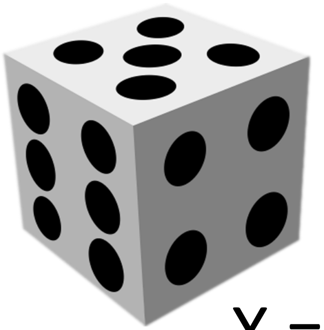
$\Omega = Z^*$



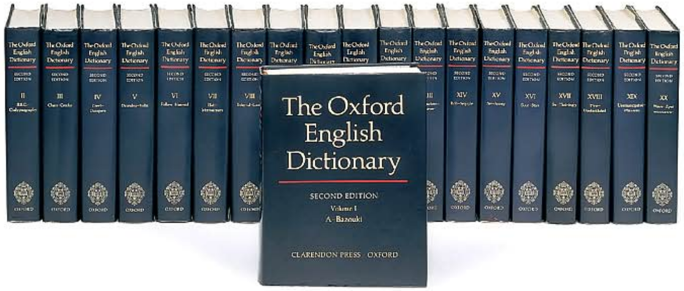
Ω = Vocabulary

Random variables

$X = \{0, 1\}$



$X = \{1, 2, 3, 4, 5, 6\}$



X = Surprisal(word | Context)

Random variables

- Map from outcomes to real numbers.
- Define random variable X for each outcome.
- Event **A**: all outcomes ω such that X(ω) = x.
- Probability mass function: **p**(x) = p(X = x) = p(**A**)

Expectation

- Expectation = weighted average of random variable

$$E(X) = \sum_{x} \mathrm{p}(x) \cdot x$$

- Fair die roll: **p**(x) = x/6
- E(X) = 1/6 + 2/6 + 3/6 + 4/6 + 5/6 + 6/6 = 3.5
- A function of X is a new random variable, say g(x).

$$E(g(X)) = \sum_{x} \mathrm{p}(x) g(x)$$

Properties of expectation

Always: $E(X + Y) = E(X) + E(Y)$

Independent: $E(XY) = E(X)E(Y)$

Variance

Variance(X) = $E[(X - E(X))^2]$

1. Subtract average from each data point.
2. Square these numbers.
3. Take a weighted average (expectation).

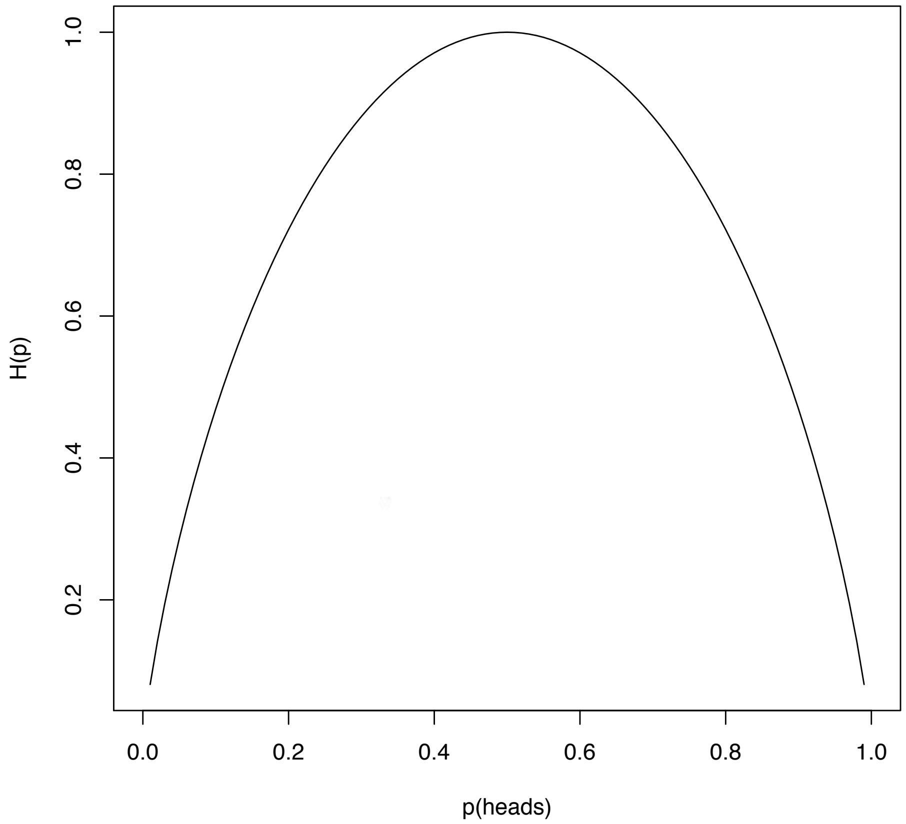
Standard deviation: $\sigma(X) = \text{sqrt}(\text{Var}(X))$

Entropy

- Entropy is expected surprisal.
- Entropy is a measure of uncertainty or disorder.
- Entropy shows the cost of transmitting information about the outcome.

$$H(X) = H(\mathrm{p}) = E(-\log_2(\mathrm{p}(x))) = -\sum_{X} \mathrm{p}(x)\log_2(\mathrm{p}(x))$$

Entropy of a coin



UNIVERSITÄT DES SAARLANDES

Kullback-Leibler (KL) divergence

- KL-divergence = how different two distributions are.
- Also known as: relative entropy
- Or, average number of bits wasted by using the second distribution to encode the first.

$$D(p||q) = \sum_{x} p(x) \log \left(\frac{p(x)}{q(x)} \right)$$

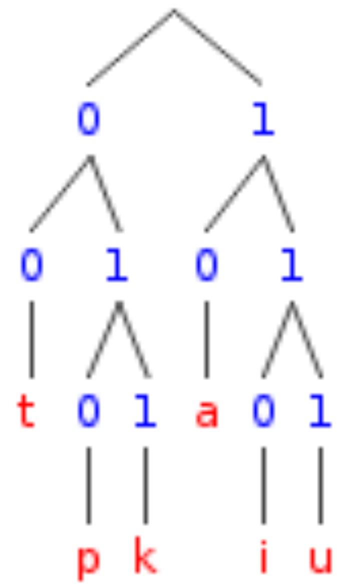
Huffman coding

1. Initialize: make each symbol a node
2. Build: While there is more than one node:
   Join the two least probable nodes into one.
3. Assign: each symbol gets a binary code

Simplified Polynesian 1

Suppose our language has 6 letters:

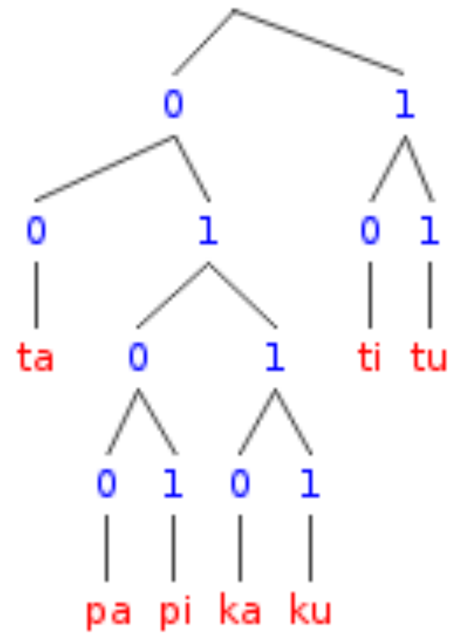
p	t	k
1/16	3/8	1/16
a	**i**	**u**
1/4	1/8	1/8



Simplified Polynesian 2

We notice that words consist of CV sequences.

	p	t	k	
a	1/16	3/8	1/16	1/2
i	1/16	3/16	0	1/4
u	0	3/16	1/16	1/4
	1/8	3/4	1/8	



Exercises

1. Memorize:
    1. expectation = weighted average of random variable
    2. entropy = expected surprisal
    3. KL-divergence = how different two distributions are

2. Encode simplified Polynesian by letter and by syllable (slides 20 and 21).

3. Find a symbol distribution such that the expected symbol code length for the Huffman code equals the entropy.