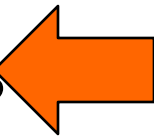
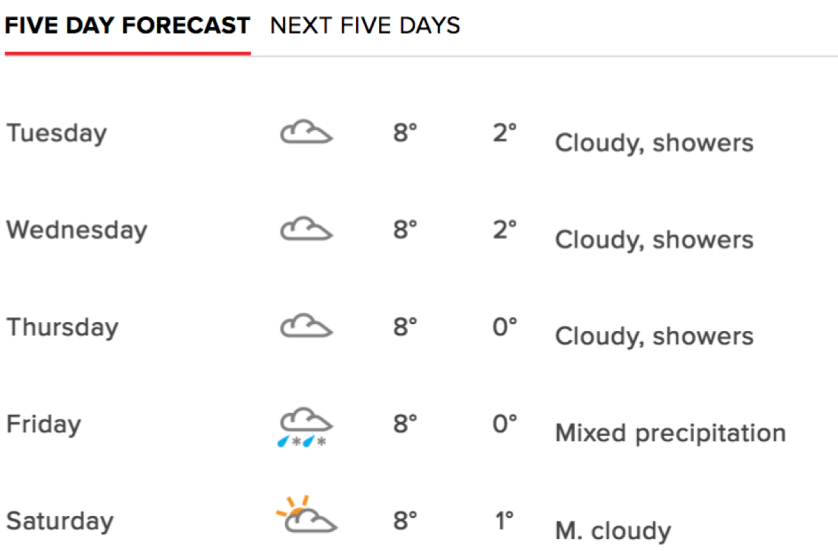
Plan

- Entropy is a measure of
  - Information
  - uncertainty
- Entropy is used for measuring password strength
- Information source: constructing probabilities
- Encoding source output using binary digits
  - Efficient codes
- Expected length of the best code



Information source in practice

- In real life we see data points
  - Read daily temperature, humidity, wind..
- Model data to make predictions, analysis
- A probabilistic model describes observed data.
  - Can generate data with the same distribution
    - Can be implemented as a program to generate sample data.



→ information source

Discrete source

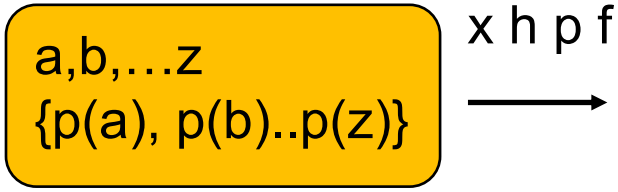
- The simplest model of information source:

- A finite set $\mathcal{X} = \{a_1, a_2, \ldots, a_n\}$, called the source alphabet.
- Probabilities: $\{p(a_1), p(a_2)..p(a_n)\}$

- The source output is a sequence, $X_1, X_2, X_3, \ldots$, of symbols drawn from $\mathcal{X}$ according to the probabilities

Discrete source: Example

- Data: English text
- Source output data: all English texts
- Source Alphabet & distribution: {a, b, c….z}
  {p(a), p(b)..p(z)}



- Output one character at a time with associated probabilities.

- There are different probabilistic models.
  - More complex models can give a more accurate model.

"alphabet," the 26 letters and a space.

1. Zero-order approximation (symbols independent and equiprobable).

Alphabet – equiprobable

XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZL-HJQD.

2. First-order approximation (symbols independent but with frequencies of English text).

OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

Alphabet , single letter prob

3. Second-order approximation (digram structure as in English).

Alphabet , digram prob

ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TU-COOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

4. Third-order approximation (trigram structure as in English).

Alphabet , trigram prob

IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONS-TURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

5. First-order word approximation. Rather than continue with tetragram, . . . , n-gram structure it is easier and better to jump at this point to word units. Here words are chosen independently but with their appropriate frequencies.

Alphabet , word prob

REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NAT-URAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE H

C.E. Shannon, A Mathematical Theory of Communication
http://www.mast.queensu.ca/~math474/shannon1948.pdf

Probabilistic Models for English text

- Different models
  - Alphabet with uniform probabilities
  - Alphabet &Single letter probabilities
  - Digram & their probabilities – dependency between letters
  - Trigram & their probabilities – higher dependency between letters
  - ….
- Higher degree statistics give more “accurate” models
  - Model output is “closer” to the real data

The entire science of information theory grew out of one electrifying paper that Shannon published in 1948, when he was a 32-year-old researcher at Bell Laboratories. Shannon showed how the once-vague notion of information could be defined and quantified with absolute precision. He demonstrated the essential unity of all information media, pointing out that text, telephone signals, radio waves, pictures, film and every other mode of communication could be encoded in the universal language of binary digits, or bits-a term that his article was the first to use in print. Shannon laid forth the idea that once information became digital, it could be transmitted without error. This was a breathtaking conceptual leap that led directly to such familiar and robust objects as CDs. Shannon had written “a blueprint for the digital age,” says MIT information theorist Robert Gallager, who is still awed by the 1948 paper.

Frequency of letters, bigrams and trigrams

Letter	Frequency
E	.127
T	.091
A	.082
O	.075
I	.070
N	.067
S	.063
H	.061
R	.060
D	.043
L	.040
C	.028
U	.028
M	.024
W	.023
F	.022
G	.020
Y	.020
P	.019
B	.015
V	.010
K	.008
J	.002
X	.001
Q	.001
Z	.001

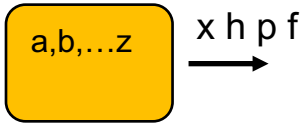
Digrams	in decreasing order	Trigrams
TH		THE
HE		ING
IN		AND
ER		HER
AN		ERE
RE		ENT
ED		THA
ON		NTH
ES		WAS
ST		ETH
EN		FOR
AT		DTH
TO		
NT		
HA		
ND		
OU		
EA		
NG		
AS		
OR		
TI		
IS		
ET		
IT		
AR		
TE		
SE		
HI		
OF		

English text: Information content

- How much information we receive from each letter?

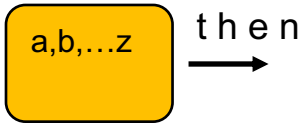
- Model English text with an “information source”:

1. source alphabet= {English letters}, { p(a)=p(b)…=p(z) }



- $F_0 = \log_2 26 = 4.7$ bits/letter

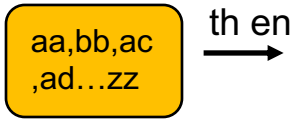
2. source alphabet= {English letters},



- Using letter probabilities:

  $F_1 = -\Sigma_{i \text{ in } \{a..z} p(i) \log_2 p(i) = 4.14$ bits/letter

3. source alphabet= {English digrams},



- Use digram probabilities
- $F_2 = -\Sigma_{x1x2 \text{ in } \{aa..zz\}} p(x_1x_2) \log_2 p(x_1x_2) = 7.12$ bits/digram

  → 3.56 bits/letter

C.E. Shannon, Prediction and Entropy of Printed English

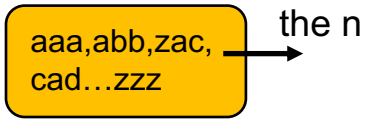
http://languagelog.ldc.upenn.edu/myl/

Entropy estimate of a letter in English text

- How much information we receive from each letter?

- …

3. source alphabet= {English trigrams},



- Use trigram probabilities
- $F_3 = -\Sigma_{x1x2x3 \text{ in } \{aaa..zzz\}} p(x_1x_2x_3) \log_2 p(x_1x_2 x_3) =$ 9.9 bits/trigram

→ 3.3 bits/letter

- Word chosen with probabilities: $F_{word} = 2.62$ bits/letter

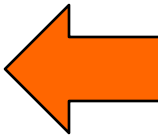
C.E. Shannon, Prediction and Entropy of Printed English
http://languagelog.ldc.upenn.edu/myl/

Why knowing this important?

- It means we can store English text efficiently
- No need to use 4.7 storage bit for each letter:
  - 2.62 bits is enough!

Plan

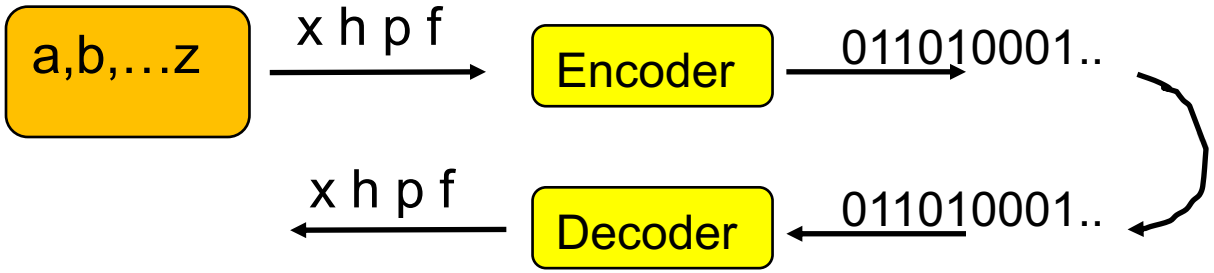
- Entropy is a measure of
  - Information
  - uncertainty
- Entropy is used for measuring password strength
- Information source: constructing probabilities
- Encoding source output using binary digits
  - Efficient codes
- Expected length of the best code



Source Coding

Problem:

- Given an information source:
  - Source alphabet $\mathcal{X}$,
  - Probability of each symbol p(x), for all x in $\mathcal{X}$
- Represent source output with a binary string.



- What is a "good" code?
  - Length efficiency
  - Computational efficiency
  - …

Discrete memory-less source (DMS)

- Source alphabet $\mathcal{X} = \{a_1, a_2, \ldots, a_m\}$
- Probabilities: $\{p(a_1), \ldots, p(a_m)\}$

- Output sequence: $X_1, X_2, X_3, \ldots,$

1. Each X_i is from $\mathcal{X}$ using the same probability distribution
2. Probability of $X_i = a_j$ does not depend on previous sequence of values
   - Probability of X_i is independent of other X_j

(Source) Codes for DMS

- Source alphabet $\mathcal{X} = \{a_1, a_2, \ldots, a_m\}$
- Probabilities: $\{p(a_1), \ldots, p(a_m)\}$

- Output sequence: $X_1, X_2, X_3, \ldots,$

- A source code C attaches a codeword (binary sequence) to each element of the alphabet

Codebook

- $C(a_1) = 00$
- $C(a_2) = 101$
- $C(a_3) = 1101$

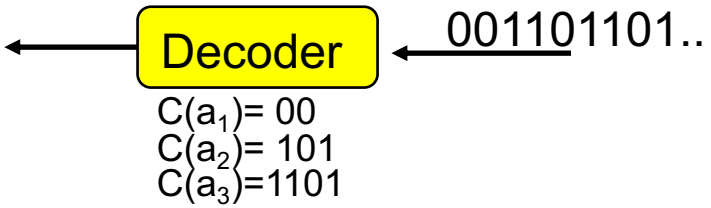
...

- Encoding $a_1a_3a_2a_2 \rightarrow$ 00 1101 101 101

Decoding

Given:

1. a binary string (coded output of the source)
   – 001101101101
2. The codebook



Find the sequence of source alphabet

Need to parse the binary string.

This is easy if all codewords have the same lengths

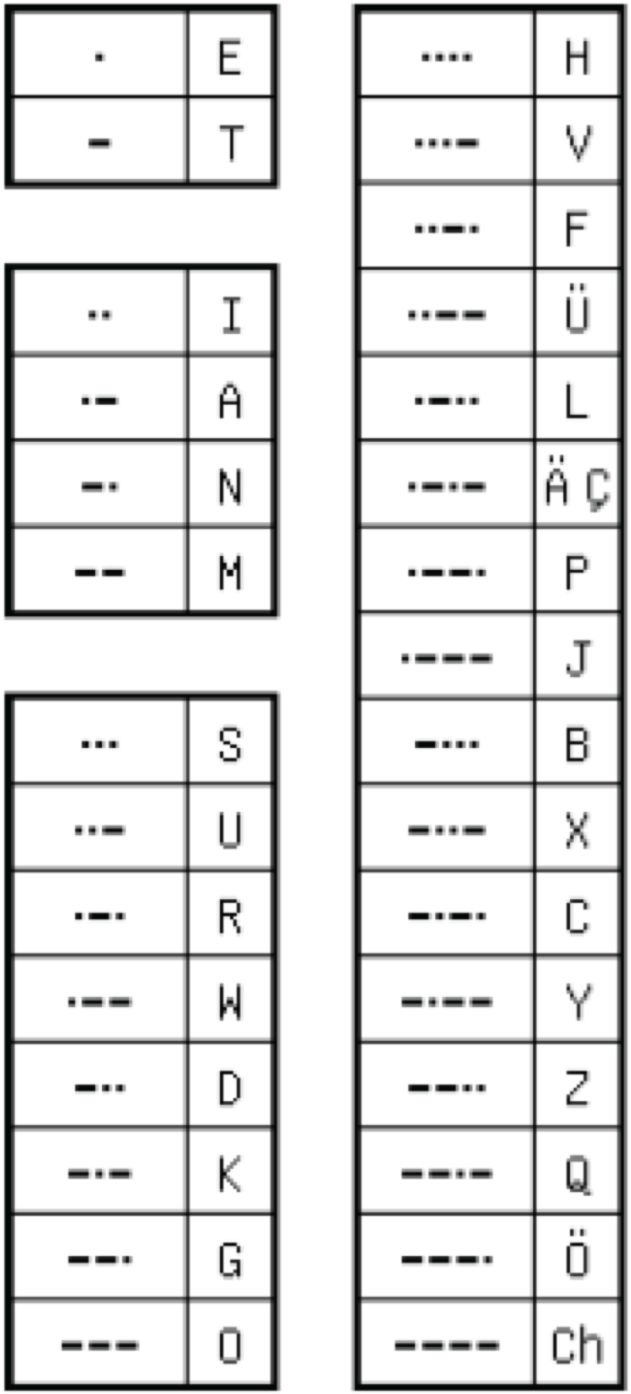
Hex	Dec	Char		Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
0x00	0	NULL	null	0x20	32	Space	0x40	64	@	0x60	96	`
0x01	1	SOH	Start of heading	0x21	33	!	0x41	65	A	0x61	97	a
0x02	2	STX	Start of text	0x22	34	"	0x42	66	B	0x62	98	b
0x03	3	ETX	End of text	0x23	35	#	0x43	67	C	0x63	99	c
0x04	4	EOT	End of transmission	0x24	36	$	0x44	68	D	0x64	100	d
0x05	5	ENQ	Enquiry	0x25	37	%	0x45	69	E	0x65	101	e
0x06	6	ACK	Acknowledg			&	0x46	70	F	0x66	102	f
0x07	7	BELL	Bell			'	0x47	71	G	0x67	103	g
0x08	8	BS	Backspace			(	0x48	72	H	0x68	104	h
0x09	9	TAB	Horizontal			)	0x49	73	I	0x69	105	i
0x0A	10	LF	New line			*	0x4A	74	J	0x6A	106	j
0x0B	11	VT	Vertical tab	0x2B	43	+	0x4B	75	K	0x6B	107	k
0x0C	12	FF	Form Feed	0x2C	44	,	0x4C	76	L	0x6C	108	l
0x0D	13	CR	Carriage return	0x2D	45	-	0x4D	77	M	0x6D	109	m
0x0E	14	SO	Shift out	0x2E	46	.	0x4E	78	N	0x6E	110	n
0x0F	15	SI	Shift in	0x2F	47	/	0x4F	79	O	0x6F	111	o
0x10	16	DLE	Data link escape	0x30	48	0	0x50	80	P	0x70	112	p
0x11	17	DC1	Device control 1	0x31	49	1	0x51	81	Q	0x71	113	q
0x12	18	DC2	Device control 2	0x32	50	2	0x52	82	R	0x72	114	r
0x13	19	DC3	Device control 3	0x33	51	3	0x53	83	S	0x73	115	s
0x14	20	DC4	Device control 4	0x34	52	4	0x54	84	T	0x74	116	t
0x15	21	NAK	Negative ack	0x35	53	5	0x55	85	U	0x75	117	u
0x16	22	SYN	Synchronous idle	0x36	54	6	0x56	86	V	0x76	118	v
0x17	23	ETB	End transmission block	0x37	55	7	0x57	87	W	0x77	119	w
0x18	24	CAN	Cancel	0x38	56	8	0x58	88	X	0x78	120	x
0x19	25	EM	End of medium	0x39	57	9	0x59	89	Y	0x79	121	y
0x1A	26	SUB	Substitute	0x3A	58	:	0x5A	90	Z	0x7A	122	z
0x1B	27	FSC	Escape	0x3B	59	;	0x5B	91	[	0x7B	123	{
0x1C	28	FS	File separator	0x3C	60	<	0x5C	92	\	0x7C	124	|
0x1D	29	GS	Group separator	0x3D	61	=	0x5D	93	]	0x7D	125	}
0x1E	30	RS	Record separator	0x3E	62	>	0x5E	94	^	0x7E	126	~
0x1F	31	US	Unit separator	0x3F	63	?	0x5F	95	_	0x7F	127	DEL

ASCII Table:
Fixed length code

l(ai) =8

Codewords with different length

- Morse code for English alphabet
- How should we parse a coded sequence?



Properties of code books

- Unique decodability: one-to-one mapping for any sequence of source symbols
  - Any source sequence is mapped into a unique bit sequence.

- Example: $\mathcal{X}$={a,b,c}
- C(a) =0, C(b) =1, C(c) =10
  is not uniquely decodable

- Prefix code: A code is prefix-free if no codeword is a prefix of any other codeword.
  - Prefix is the initial substring.

- Prefix codes are uniquely decodable.

Source coding: Efficiency

- Length of a codeword $C(a_j)$ is denoted by $\ell(a_j)$: The number of bits in $C(a_j)$

- A source:
  alphabet $\mathcal{X}$ = {a,b,c}, { p(a),p(b), p(c) }

- C is a code for the source
- C(a)= 0 ℓ(a)=1 binary digits
- C(b)= 10 ℓ(b)=2 binary digits
- C(c)= 11 ℓ(c)=2 binary digits

Measure of efficiency for codes

- Expected length of the code

$$L_{\text{exp}} = \sum_{i=1}^{n} p_i l_i$$

- Communication efficiency
  - not computational efficiency

Example

- Source $\mathcal{X}$= {a,b,c,d}
- p(a)= p(b)= 0.125, p(c) = 0.25, p(d) = 0.5

Code I:

- d→ 0, c→ 11 a→ 101 b→100
- L_{exp}= 1 x 0.5 + 2 x 0.25 + 2 x 3 x 0.12
- L_{exp}(Code I) = 1.75 binary digits

Code II:

- d→ 00 c→ 11 a→ 10 b→01
- L_{exp}(Code II) = 2 binary digits

Noiseless coding theorem

Fundamental Theorem of source coding

- Theorem: $L_{exp} \geq H(X)$

The expected length of any prefix code for a DMC source in lower bounded by the entropy of the source.

(proof omitted –in CT, or reference below)

- That is, the expected number of binary digits to encode a source will be at least equal to the average information of the source.

http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-450-principles-of-digital-communications-i-fall-2006/lecture-notes/book_2.pdf