Information Theory and Networks
Lecture 11: Coding Language

Matthew Roughan
<matthew.roughan@adelaide.edu.au>
http://www.maths.adelaide.edu.au/matthew.roughan/Lecture_notes/InformationTheory/

School of Mathematical Sciences,
University of Adelaide

September 18, 2013

Part I

Coding Language

Thanks to the redundancy of language, yxx cxn xndxrstxnd whxt x xm wrxtxng xvxn xf x rxplxcx xll thx vxwxls wxth xn "x".
t gts lttl hrdr f y dn't vn kn whr th vwls r.
Steven Pinker

Aoccdrnig to rscheearch at an Elingsh uinervtisy, it deosnt mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer is at the rghit pclae.
http://knowyourmeme.com/memes/aoccdrnig-to-rscheearch

TOBEORNOTTOBEORTOBEORNOT

Section 1

Language and Redundancy

Redundancy

Definition (redundant)

Adjective:

1. No longer needed or useful; superfluous.
2. (of words or data) Able to be omitted without loss of meaning or function.

English (and other languages) have a lot of redundancy.

- but the definition is misleading
  - it is needed, to ensure that communication is accurate even when there is noise
  - English has subtlety layered on complexity
  - we are ignoring poetic and aesthetic considerations
- but no doubt there is a lot that can be dropped in some cases

Redundancy
Definition (redundant)
Adjective:
1. No longer needed or useful; superfluous.
2. (of words or data) Able to be omitted without loss of meaning or function.
English (and other languages) have a lot of redundancy.
- but the definition is misleading
  - it is needed, to ensure that communication is accurate even when there is noise
  - English has subtlety layered on complexity
  - we are ignoring poetic and aesthetic considerations
- but no doubt there is a lot that can be dropped in some cases

Redundancy in English

- We use extra words
  "He was ready and able"
- We use more letters than we need: see earlier quotes
- We use indicators of all types to ensure clear meanings, when mostly it would be obvious (e.g., apostrophe's)
- Letters and words don't have equal frequencies

Redundancy in English
- We use extra words
  "He was ready and able"
- We use more letters than we need: see earlier quotes
- We use indicators of all types to ensure clear meanings, when mostly it would be obvious (e.g., apostrophe's)
- Letters and words don't have equal frequencies

Entropy, Redundancy and Compression

- So we know that there is something that could be compressed
  - at least for error free communication
- How can we exploit it?
  - lets go back to Entropy and coding, to start with

Entropy, Redundancy and Compression

- So we know that there is something that could be compressed
  - at least for error free communication
- How can we exploit it?
  - lets go back to Entropy and coding, to start with

Letters and numbers

All letters are just numbers!

- all data on the Internet is just stored as numbers
- standard methods
  - ASCII (American Standard Code for Information Interchange)
    - pronounced "Ass-kee"
    - developed from telegraphic codes (around 1960s)
    - includes 128 characters, including punctuation and 33 non-printing characters (line feeds, tabs, etc.)
    - so we need 7 bits
    - often done as 8 bits (which fits into one byte, and allows extra machine dependent codes)
  - Unicode is taking over
    - has support for non-English character sets
    - 110,000 characters covering 100 scripts

Letters and numbers

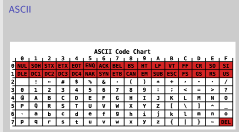
All letters are just numbers!

- all data on the Internet is just stored as numbers
- standard methods
  - ASCII (American Standard Code for Information Interchange)
    - pronounced "Ass-kee"
    - developed from telegraphic codes (around 1960s)
    - includes 128 characters, including punctuation and 33 non-printing characters (line feeds, tabs, etc.)
    - so we need 7 bits
    - often done as 8 bits (which fits into one byte, and allows extra machine dependent codes)
  - Unicode is taking over
    - has support for non-English character sets
    - 110,000 characters covering 100 scripts

ASCII

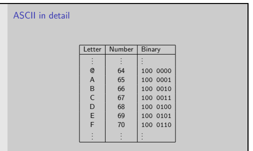
ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

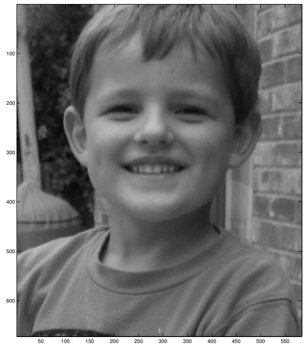


ASCII in detail

Letter	Number	Binary
⋮	⋮	⋮
@	64	100 0000
A	65	100 0001
B	66	100 0010
C	67	100 0011
D	68	100 0100
E	69	100 0101
F	70	100 0110
⋮	⋮	⋮



ASCII art



⇒

English

ASCII coding of letters uses 8 bits per letter (typically)

H for English

1. Simple random letters with typical frequencies

$$H(X) \simeq 4 \text{ bits per letter}$$

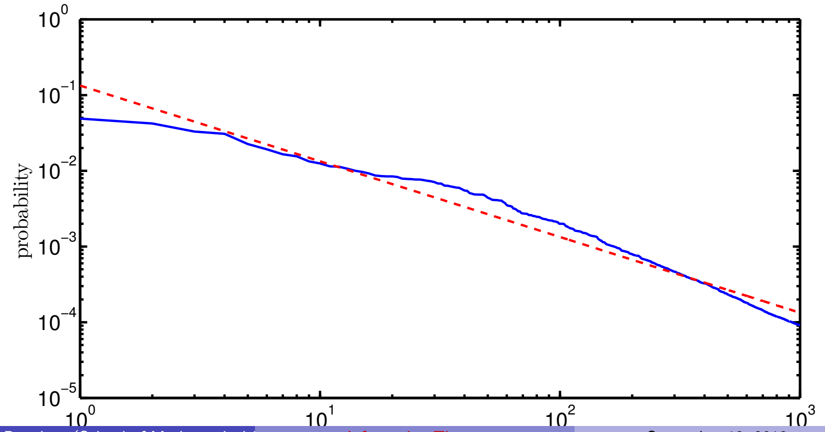
Compression ratio of about 2 to 1.

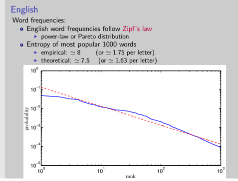
English
ASCII coding of letters uses 8 bits per letter (typically)
H for English
1. Simple random letters with typical frequencies
H(X) ≃ 4 bits per letter
Compression ratio of about 2 to 1.

English

Word frequencies:

- English word frequencies follow Zipf's law
  - power-law or Pareto distribution
- Entropy of most popular 1000 words
  - empirical: $\simeq 8$ (or $\simeq 1.75$ per letter)
  - theoretical: $\simeq 7.5$ (or $\simeq 1.63$ per letter)





Zipf's law = the frequency of any word is inversely proportional to its rank in the frequency table.

Data from TV scripts via
http://en.wiktionary.org/wiki/Wiktionary:Frequency_lists

English

ASCII coding of letters uses 8 bits per letter (typically)

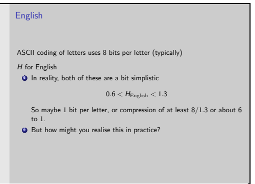
H for English

1. In reality, both of these are a bit simplistic

$$0.6 < H_{\text{English}} < 1.3$$

So maybe 1 bit per letter, or compression of at least 8/1.3 or about 6 to 1.

2. But how might you realise this in practice?



Upper Bound on Optimal Codes

Remember:

Theorem

The expected length L of the optimal code for a random variable X is bounded below by the entropy of X, i.e.,

$$H_D(X) \leq L < H_D(X) + 1.$$

That $+1$ could be really critical, if the entropy is less than 1 per character.

Upper Bound on Optimal Codes
Remember:
Theorem
The expected length L of the optimal code for a random variable X is bounded below by the entropy of X, i.e.,
$H_D(X) \leq L < H_D(X) + 1.$
That +1 could be really critical, if the entropy is less than 1 per character.

Remember the ceiling $\lceil x \rceil$ is the smallest integer $\geq x$.

Block encoding

- We can see that there is at least a small loss of efficiency for codes, when we don't have natural integer length codes.
- This can actually be quite a big cost, in terms of optimality
  - in binary codes its up to one bit per symbol
- We can spread the overhead out by coding blocks of symbols at a time
  - to understand how to do this properly, we need a better model for language, and the entropy thereof

Block encoding
We can see that there is at least a small loss of efficiency for codes, when we don't have natural integer length codes.
This can actually be quite a big cost, in terms of optimality
in binary codes its up to one bit per symbol
We can spread the overhead out by coding blocks of symbols at a time
to understand how to do this properly, we need a better model for language, and the entropy thereof

Block encoding

Encode blocks $X_1, X_2, \ldots, X_n$, then the expected code length for the entire block will be

$$H(X_1, X_2, \ldots, X_n) \leq E\left[\ell(X_1, X_2, \ldots, X_n)\right] < H(X_1, X_2, \ldots, X_n) + 1$$

If the X_i are IID, then

$$H(X_1, X_2, \ldots, X_n) = nH(X)$$

so the length of code per input symbol satisfies

$$H(X) \leq L_n < H(X) + 1/n$$

If we use large blocks, we can achieve very close to the best possible efficiency, but the assumption that the symbols are IID is a little too strong.

- We need to deal with more general stochastic processes
- We need to incorporate correlations

Block encoding
Encode blocks $X_1, X_2, \ldots, X_n$, then the expected code length for the entire block will be
$$H(X_1, X_2, \ldots, X_n) \leq E\left[\ell(X_1, X_2, \ldots, X_n)\right] < H(X_1, X_2, \ldots, X_n) + 1$$
If the X_i are IID, then
$$H(X_1, X_2, \ldots, X_n) = nH(X)$$
so the length of code per input symbol satisfies
$$H(X) \leq L_n < H(X) + 1/n$$
If we use large blocks, we can achieve very close to the best possible efficiency, but the assumption that the symbols are IID is a little too strong.
- We need to deal with more general stochastic processes
- We need to incorporate correlations

Section 2

Markov Chains

Section 2
Markov Chains

Stochastic Process

Definition (Stochastic Process)

A stochastic process is an indexed series of random variables

$$(X_1, X_2, \dots)$$

characterised by the joint PMFs

$$P\big((X_1, X_2, \dots, X_n) = (x_1, x_2, \dots, x_n)\big) = p(x_1, x_2, \dots, x_n)$$

for all n.

In general:

- the X_i don't have to come from the same sample set Ω
- the X_i can have any dependency structure you like

this is a little hard to handle, so we will restrict our attention.

Stochastic Process
Definition (Stochastic Process)
A stochastic process is an indexed series of random variables
$(X_1, X_2, \dots)$
characterised by the joint PMFs
$P((X_1, X_2, \dots, X_n) = (x_1, x_2, \dots, x_n)) = p(x_1, x_2, \dots, x_n)$
for all n.
In general:
the X_i don't have to come from the same sample set Ω
the X_i can have any dependency structure you like
this is a little hard to handle, so we will restrict our attention.

Stationarity

Definition (Stationary)

A stochastic process is said to be stationary if the joint distribution of any subset of the sequence of RVs is invariant with respect to shifts in the time index, i.e.,

$$\begin{aligned} & P\big((X_1, X_2, \dots, X_n) = (x_1, x_2, \dots, x_n)\big) \\ = \; & P\big((X_{1+t}, X_{2+t}, \dots, X_{n+t}) = (x_1, x_2, \dots, x_n)\big) \end{aligned}$$

for any shift t, and for all $x_i \in \Omega$.

Stationarity significantly restricts the processes we consider, but not always enough, so we shall do one further restriction.

Stationarity
Definition (Stationary)
A stochastic process is said to be stationary if the joint distribution of any subset of the sequence of RVs is invariant with respect to shifts in the time index, i.e.,
$P((X_1, X_2, \dots, X_n) = (x_1, x_2, \dots, x_n)) = P((X_{1+t}, X_{2+t}, \dots, X_{n+t}) = (x_1, x_2, \dots, x_n))$
for any shift t, and for all $x_i \in \Omega$.
Stationarity significantly restricts the processes we consider, but not always enough, so we shall do one further restriction.

Implicitly, for a stationary sequence, all the RVs X_i are in the same sample set Ω.

In practice, we often use a weaker form of stationarity (called weak stationarity, or second order stationarity) that requires that the mean and variance are invariant to time shifts.

Markov Chains

Definition (Markov Chain)

We call a discrete stochastic process a Markov Chain if the next state change depends only on the current state, not the entire history of the process, i.e., for $n = 1, 2, \ldots$

$$\begin{aligned} & P\big(X_{n+1} = x_{n+1} \big| X_1, X_2, \ldots, X_n\big) \\ = \quad & P\big(X_{n+1} = x_{n+1} \big| X_n\big) \end{aligned}$$

for all $x_i \in \Omega$.

This expresses a type of conditional independence of the process, namely that the past and future are independent, conditional on the current state.

We call a Markov Chain time invariant or homogeneous if it is also stationary, and we will assume that all of our Markov Chains are thus, unless otherwise specified.

For more details on Markov Chains and their extension to continuous time, Markov Processes, see other courses, e.g. Applied Probability III.

Transition Matrix

Definition (Probability Transition Matrix)

For a time-invariant Markov Chain, we define the probability transition matrix $P = [p_{ij}]$ by

$$p_{ij} = P\big(X_{n+1} = j \big| X_n = i\big)$$

for $i, j = 1, 2, \ldots, m$ where $\Omega = \{1, 2, \ldots, m\}$.

The probability transition matrix is a stochastic matrix, i.e., its elements are non-negative, and its rows sum to one.

Some more definitions

Definition (Irreducible)

We say a time-invariant Markov Chain is irreducible if it is possible to go from any state to any other state with positive probability.

Definition (Periodic)

A state i has period k if any return to the state must occur in multiples of k time steps. If the only valid $k = 1$, then we say the state is aperiodic. A Markov Chain is aperiodic if all states are aperiodic.

Some more definitions
Definition (Irreducible)
We say a time-invariant Markov Chain is irreducible if it is possible to go from any state to any other state with positive probability.
Definition (Periodic)
A state i has period k if any return to the state must occur in multiples of k time steps. If the only valid $k = 1$, then we say the state is aperiodic. A Markov Chain is aperiodic if all states are aperiodic.

An irreducible Markov Chain needs only one aperiodic state to imply the Markov chain is aperiodic.

Some more definitions

Definition (Recurrence)

A state is transient if, given we start in state i, there is a non-zero probability that we will never return to i. If the state is not transient, it is recurrent, and it is positive recurrent if the expected time to the next recurrence is finite.

Definition (Ergodic)

A state i is called ergodic if it is aperiodic and positive recurrent, and if all states in an irreducible Markov Chain are ergodic, we say the Markov Chain is ergodic.

Typically, we will assume our Markov Chains are homogeneous and ergodic.

Some more definitions
Definition (Recurrence)
A state is transient if, given we start in state i, there is a non-zero probability that we will never return to i. If the state is not transient, it is recurrent, and it is positive recurrent if the expected time to the next recurrence is finite.
Definition (Ergodic)
A state i is called ergodic if it is aperiodic and positive recurrent, and if all states in an irreducible Markov Chain are ergodic, we say the Markov Chain is ergodic.
Typically, we will assume our Markov Chains are homogeneous and ergodic.

There are many other properties and theorems we could present, but this set should suffice for our work.

Markov Chain Results

Given a current state probabilities at time n, $\boldsymbol{\mu}^{(n)}$, i.e.,

$$\mu_i^{(n)} = P(X_n = i),$$

We can calculate the state probabilities after a transition by

$$\begin{aligned} \mu_i^{(n+1)} &= P(X_{n+1} = i) \\ &= \sum_j P(X_{n+1} = i | X_n = j) P(X_n = j) \\ &= \sum_j \mu_j^{(n)} p_{ji} \end{aligned}$$

or, in vector notation

$$\boldsymbol{\mu}^{(n+1)} = \boldsymbol{\mu}^{(n)} P$$

and hence

$$\boldsymbol{\mu}^{(n+1)} = \boldsymbol{\mu}^{(1)} P^n$$

Markov Chain Results
Given a current state probabilities at time n, $\boldsymbol{\mu}^{(n)}$, i.e.,
$\mu_i^{(n)} = P(X_n = i)$,
We can calculate the state probabilities after a transition by
$\mu_i^{(n+1)} = P(X_{n+1} = i) = \sum_j P(X_{n+1} = i | X_n = j) P(X_n = j) = \sum_j \mu_j^{(n)} p_{ji}$
or, in vector notation $\boldsymbol{\mu}^{(n+1)} = \boldsymbol{\mu}^{(n)} P$
and hence $\boldsymbol{\mu}^{(n+1)} = \boldsymbol{\mu}^{(1)} P^n$

Stationary Distribution

Give a homogeneous Markov Chain, then the vector $\boldsymbol{\pi}$ is called a stationary distribution (or invariant measure) if

- It is a PMF (e.g., non-negative, and summing to one)
- And

$$\boldsymbol{\pi} = \boldsymbol{\pi} P$$

An irreducible Markov Chain has a stationary distribution iff all its states are positive recurrent, in which case $\boldsymbol{\pi}$ is unique, and

$$\lim_{n \to \infty} P^n = \mathbf{1}\boldsymbol{\pi}$$

This is often called the equilibrium distribution of the Markov chain.

Stationary Distribution
Give a homogeneous Markov Chain, then the vector $\boldsymbol{\pi}$ is called a stationary distribution (or invariant measure) if
- It is a PMF (e.g., non-negative, and summing to one)
- And $\boldsymbol{\pi} = \boldsymbol{\pi} P$
An irreducible Markov Chain has a stationary distribution iff all its states are positive recurrent, in which case $\boldsymbol{\pi}$ is unique, and
$\lim_{n \to \infty} P^n = \mathbf{1}\boldsymbol{\pi}$
This is often called the equilibrium distribution of the Markov chain.

Higher Order Markov Chains

What if process depends on some history?

- Create a new process, whose states include some history: e.g.,
  - assume $n+1$ state depends on n and $n-1$
  - create
$$Y_n = (X_n, X_{n-1})$$
  - Now Y_n is a Markov chain
$$\begin{aligned} P\big(Y_{n+1}\big|Y_1,\ldots,Y_n\big) &= P\big(Y_{n+1}\big|Y_n\big) \\ &= P\big((X_{n+1},X_n)\big|(X_n,X_{n-1})\big) \\ &= P\big(X_{n+1}\big|X_n,X_{n-1}\big) \end{aligned}$$
- We can do this for an arbitrary amount of the history
  - 2nd order = case above
  - 3rd order = include three states of the history
- Note that state space expands
  - nth order Markov chain on m states has m^d states

Markov Chains for Letters [Sha48]

- **0th order:** equiprobable letters
XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZL HJQD
- **1st order:** IID letter frequencies
OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL
- **2nd order:** simple Markov Chain, i.e., diagrams
ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE
- **3rd order:** 2nd order Markov chain, i.e., trigrams
N NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE

Sorry that the “order” used above (the order of the approximation) doesn't match the “order” of the Markov Chain, but the examples above come directly from Shannon's paper [Sha48], so I am trying to be consistent with his notation, as well as common Markov chain notation.

There are 27 symbols modelled, the 26 letters, and the space (so 27 states in the 1st order Markov Chain).

Markov Chains for Letters [Sha48]

- **1st order words:** just based on word frequencies
  REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.
- **2nd order words:** simple Markov Chain for words
  THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

Markov Chains for Letters [Sha48]
- 1st order words: just based on word frequencies REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.
- 2nd order words: simple Markov Chain for words THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

The examples above come directly from Shannon's paper [Sha48].

Entropy for Markov Chains

- So, to get coding closer to the real entropy of English
  1. have to account for these correlations, across letters and/or words
  2. have to have a way of calculating entropy
  3. need to have a way to code

Further reading I

Gjerrit Meinsma, *Data compression & information theory*, Mathematisch cafe, 2003, `wwwhome.math.utwente.nl/~meinsmag/onzin/shannon.pdf`.

C.E. Shannon, *A mathematical theory of communication*, The Bell System Technical Journal **27** (1948), 379–423,623–656, `http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html`.