Basics of Information Theory

Samuel Brody

February 2009

Outline

Kullback Leibler (KL) divergence

The *Kullback Leibler (KL) divergence* between two probability distributions $p(x)$ and $q(x)$ is defined as:

$$D_{KL}[p||q] = \sum_x p(x) log \frac{p(x)}{q(x)}$$

where $0 log \frac{0}{q} = 0$ and $p log \frac{p}{0} = \infty$

- $D_{KL}[p; q] \geq 0$ and $D_{KL}[p; q] = 0 \Leftrightarrow p(x) = q(x) \quad \forall x \in X$
  - However, it does not satisfy the conditions of a *distance metric*
- it is not symmetric in p and q, so $D_{KL}[p||q] \neq D_{KL}[q||p]$
- it does not satisfy triangle inequality, i.e., it does not hold that

$$D_{KL}[p||r] \leq D_{KL}[p||q] + D_{KL}[q||r]$$

Also, if $q(x) = 0$ for some $x \in X$, we can get infinite divergence.

Jensen-Shannon (JS) Divergence

Jensen-Shannon (JS) divergence solves some of the problems of KL divergence.

$$JS_{\Pi}[p||q] = \pi_1 \cdot D_{KL}[p||r] + \pi_2 \cdot D_{KL}[q||r]$$

where $\Pi = \{\pi_1, \pi_2\}$ is a distribution, and we define:

$$r(x) = \pi_1 \cdot p(x) + \pi_2 \cdot q(x) \quad \forall x \in X$$

- if $\Pi = \{\frac{1}{2}, \frac{1}{2}\}$ is symmetric, then:
  - $JS[p||q]$ is symmetric in p and q
  - $JS[p||q] \neq \infty$
- JS is still not a metric, since is does not satisfy triangle inequality.

Cross Entropy

The cross entropy between a random variable X with true probability distribution p(x) and another probability distribution q (normally a model of p) is given by:

$$H(X,q) = H(X) + D_{KL}(p||q) = -\sum_{x \in X} p(x) log_2 q(x)$$

- i.e., the uncertainty inherent in X plus the uncertainty added by incorrectly modeling X.
- $p(x)$, and therefore $H(X)$ is usually unknown.
- We can get a approximation of it, through a test set:

$$H(T,q) \approx -\frac{1}{|T|} \sum_{t_i \in T} log_2 q(t_i)$$

Cross Entropy & Perplexity

Cross Entropy:

$$H(X,q) = H(X) + D(p||q) = -\sum_{x \in X} p(x) log_2 q(x)$$

- The better the model, the lower the cross entropy of the test set.
- For any interesting distribution, $H(X) > 0$ and unknown.
- Therefore, the cross entropy will be > 0 and meaningful only for comparing different models.

Perplexity: $perplexity(T,q) = 2^{H(T,q)} = \prod_{t_i \in T} q(t_i)^{-\frac{1}{|T|}}$

Outline

Channel Model

- Information theory was devised for modeling communication through a noisy channel.

  SOURCE $\rightarrow$ [NOISY CHANNEL] $\rightarrow$ TARGET

- Examples of noisy channels:
  - speaker $\rightarrow$ phone line $\rightarrow$ listener
  - modem $\rightarrow$ phone line $\rightarrow$ modem
  - computer memory $\rightarrow$ disk $\rightarrow$ computer memory

Encoding

- The important questions are:
  - How can we reduce errors?
  - How efficient can we be? (fewest bits)

- The Model:
  SOURCE $\rightarrow$ [*ENCODER*] $\rightarrow$ [NOISY CHANNEL] $\rightarrow$ [*DECODER*] $\rightarrow$ TARGET

Channel Theorem

Claude Shannon proved that, given certain assumptions,

- as the length of the message approaches infinity $n \rightarrow \infty$
- it is theoretically possible to reduce the error probability arbitrarily close to zero
- while maintaining a rate approaching one $R = \frac{|enc(X)|}{|X|} \rightarrow 1$

In Practice, message lengths can't be too big.

But information theory can help bring us close to optimal.

Reducing Errors

Strategies for reducing errors:

- simple repetition
- simple repetition
- error detection/correction codes

Repetition Codes

R_k : encode by repeating each bit k times, and decode by majority.

Example: R_3

Source	0	0	1	0	1	1	0
Encoding	000	000	111	000	111	111	000
Noise	000	001	000	000	101	000	000
Output	000	001	111	000	010	111	000
Decoding	0	0	1	0	0	1	0

Repetition Codes - cont.

What are the assumptions?

- noise is independently and randomly distributed
- probability of a bit flip < 0.5

Error Detection Codes

- Repetition decreases the probability of error, exponentially in k.
- But is multiplies the length by k.

- Suppose it was enough to know that an error occurred ...
- we could then ask for a re-send.

Parity Bit

- We can use a parity bit:

Source	000	001	111	000	010	101	110
Encoding	000 0	001 1	111 1	000 0	010 1	101 0	110 0
Noise	000 0	000 1	000 0	000 0	100 1	100 0	000 0
Output	000 0	001 0	111 1	000 0	110 0	001 0	110 0
Decoding	000	error	111	000	110	error	110

Error Detection and Correction

- By using a series of arity bits, it's possible to detect *and correct* single bit errors.
- Hamming codes are an example.
- Shorter than repetition: $n + log(n)$ rather than $k \cdot n$.

Error Detection and Correction

- Almost all forms of electronic data transfer use some kind of error detection and either correct or request resend.
- Media: disk drives, CDs, memory sticks
- Communication: phone, internet packages, wireless, TV broadcasts

Compression

- We've seen how errors can be detected and corrected at the cost of increased length.
- How do we make the most efficient use of the channel?
- Can we represent the data in a more efficient way, and save bits?

Compression

When is compression impossible?

- when the message is completely random.
- each letter/state is assigned a number, and if there are $n = 2^k$ states, we'll need k bits.

But if messages are random, we aren't transmitting information!

Compression - cont.

- Random messages (maximal entropy) are uncompressable.
- That's why finding out the entropy of English is helpful.
- We use patterns in the messages to compress them.

Simple Compression

- k repeats of a letter are replaced by a number representing the length
- For example: alphabet of two letter a, b
- We encode:
  - single ‘a’ = ‘00’
  - single ‘b’ = ‘01’
  - a repeated n times (up to 2^k) = ‘10 [binary representation of n]’
  - b repeated n times (up to 2^k) = ‘11 [binary representation of n]’

Simple Compression - cont.

- Example($k = 3$):

	a	b	aaaaa	bbbbbbb	aaaaaaa	bbbbbb	
comp.	00	01	10 101	11 111	10 111	11 110	24 bits
uncomp.	0	1	00000	1111111	0000000	111111	27 bits

- When does it work?
  - When you can expect long repeats of one letter/state.
- Used in fax machines, where there's lots of white stretches.

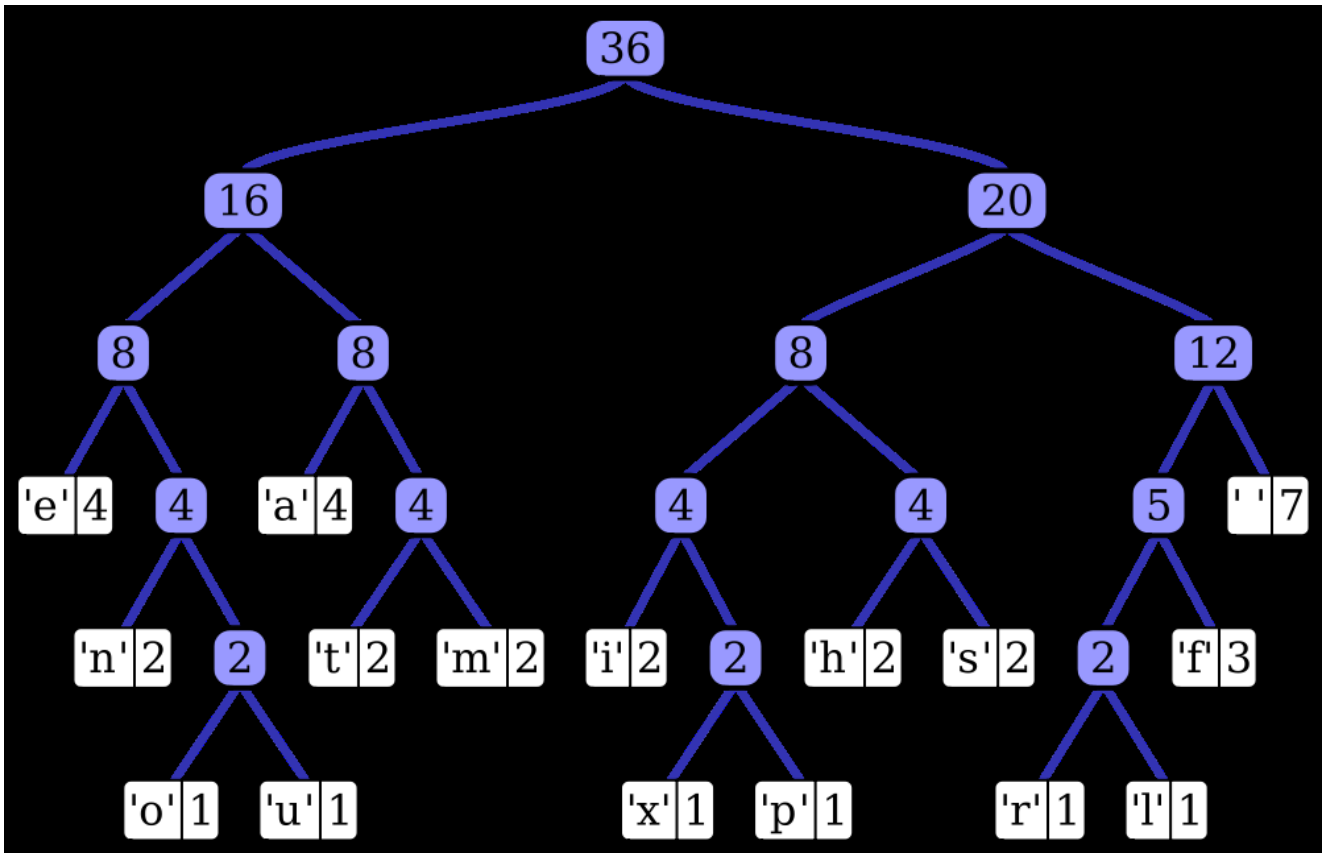
Using Probabilities

- Remember the two coin example?
- Your friend tosses two coins, you need to guess the number of heads with fewest guesses.
- What's the optimal strategy?

Morse Code

0.1304 E .	0.0249 M - -
0.1045 T -	0.0249 U . . -
0.0856 A . -	0.0199 G - - .
0.0797 O - - -	0.0199 Y - . - -
0.0707 N - .	0.0199 P . - - .
0.0677 R . - .	0.0149 W . - -
0.0627 I . .	0.0139 B - . . .
0.0607 S . . .	0.0092 V . . . -
0.0528 H	0.0042 K - . -
0.0378 D - . .	0.0017 X - . . -
0.0339 L . - . .	0.0013 J . - - -
0.0289 F . . - .	0.0012 Q - - . -
0.0279 C - . - .	0.0008 Z - - . .

Huffman Code



Char	Freq	Code
space	7	111
a	4	010
e	4	000
f	3	1101
h	2	1010
i	2	1000
m	2	0111
n	2	0010
s	2	1011
t	2	0110
l	1	11001
o	1	00110
p	1	10011
r	1	11000
u	1	00111
x	1	10010

Figure: Huffman tree generated from the exact frequencies of the text: "this is an example of a huffman tree"