Data Compression

Based partly on D. Cyganski, J. Orr, "Information Technology:Inside and Outside", Prentice Hall 2000.

General problem, but particularly important with images since so much data is involved.

Basic idea is to remove redundancy in the data and thus save space.

Recall the example with a 250 x 400 image of all one colour.

In a simple bitmap format we are writing the same colour number 100,000 times! Really just want to say 250 x 400 and the colour number.

Informed about information

Need to think about what information really is.

Consider three related terms:

- Message: what we are trying to convey. For example, "Digital technology is important to our economy."
- Data: specific symbols used to convey the message. Could be letters in the English language or 1s and 0s representing those letters or whatever.
- Information: More difficult idea.
  - Really the amount of surprise.
  - No information in "Digital technology is important to our economy" because you already know that.
  - The statement "Technology will stop advancing in 2019" would have information. That's something you didn't know.

Redundancy

Basic idea is to avoid telling you things twice and avoid telling you things you know.

Want to store and/or send pure information.

The study of what constitutes information and quantification of how much info we have is called Information Theory.

A Bit of History

Whole field of information theory began with two papers published by Claude Shanon in 1948.

Maths

Need some ideas from probability and statistics.

P=0 means you don't believe the event will ever happen.

P=1 means you believe the event will happen.

0 < P < 1 means you're not sure the outcome of any particular trial, but expect event to occur Px100% of the time on average.

Example: probability or rolling heads = 0.5.
Means we expect heads about half the time.

Independent Events

Two events are independent if they don't affect each other.

Toss coin. Get heads or tails.

Toss it again. Result has nothing to do with the first toss.
The tosses are independent events.

Probabilities for independent events multiply.

Probability of getting heads twice is $P = \frac{1}{2} * \frac{1}{2} = \frac{1}{4}$.

Example Data Compression

Consider the customers at a mobile hamburger stand that sells meat burgers and veggieburgers. .

Home office is doing marketing research and wants to know about the purchase of meat burgers and veggieburgers.

Store assigns 0 to meat, 1 to veggie, and sends a data stream to the home office.

011100100110101001100010101000110101001100011 …..

Turns out the customers are 50% meat eaters and 50% veggieburger eaters.

Does no good to try to guess the choice of the next customer.

Choice of each new patron is a surprise. Each one is a bit of info. There is one bit of info per bit of data.

A Vegetarian Convention

Now suppose the same marketing research is done at a stand in near a vegetarian convention. 80% of patrons are vegetarians.

You would do well to guess the next sale is V. There's less surprise and thus less information if the sale is V.

Seems like there should be a way to take advantage of that.

Here's an approach. Group the patrons in twos. Assign the following codes to the possible combinations. Note: Reversible!

V-V: 0

V-M: 10

M-V: 110

M-M: 111

Seems like I don't gain. I have to send three bits instead of two if the data is M-V or M-M.

Vegetarian Convention

Not so! Look at the probabilities:

V-V: 0 0.64

V-M: 10 0.16

M-V: 110 0.16

M-M: 111 0.04

Most of the time I send only one bit!

On average I send how many bits?

Number bits = 0.64*1 + 0.16*2 + 0.16*3 + 0.04*3 = 1.56 bits.

But that's for two patrons.

So I send only 0.78 bits per patron.

I have achieved data compression.

And it is lossless – I can recover the exact sequence of patron genders from the compressed data.

Vegetarian Convention

Compressing the Data

Data 11 11 10 11 01 11 11 00 11 11 11 10 11 11

V-V: 0	0.64
V-M: 10	0.16
M-V: 110	0.16
M-M: 111	0.04

Compressed string 20 bits

0 0 10 0 110 0 0 111 0 0 0 10 0 0

Vegetarian Convention

Recovering the data

V-V: 0	0.64
V-M: 10	0.16
M-V: 110	0.16
M-M: 111	0.04

00100110001110001000   compressed string 20 bits

0 0 10 0 110 0 0 111 0 0 0 10 0 0

vv vv vm vv mv vv vv mm vv vv vv vm vv vv

11 11 10 11 01 11 11 00 11 11 11 10 11 11

Recovered original 28 bits

Theory

A detailed theoretical analysis of this kind of problem was carried out by Shanon.

Showed that, with 80% 1s and 20% 0s, the information content is 0.71 bits of info per bit of data and I should be able to make a compression routine to send only 0.71 times as many bits as I collect without losing anything.

Just make larger groups of customers in this example.

An Important Idea

Lossless data compression now involves a lot of very fancy algorithms, but it's all based on these same ideas.

Difference between data and information

Getting rid of redundancy in the data.

Lossless Image Compression

We can expect this to work very well on images and movies.

Not much variation from one pixel to the next within an image or the image is just a jumble.

In a movie, not much variation from one frame to the next or the motion is a jumble.

Can predict the value of the next pixel or frame mostly and record only the surprise (change).

Lossy Image Compression

Lossless data compression applies to data of all sorts.

Makes use of the difference between information and data to reduce redundancy in the data and shorten the stream of 1s and 0s.

Original data can be recovered completely.

Images (and sounds) can be further compressed by reducing their quality slightly or maybe more than slightly.

Not applicable to most data. Change one word in a contract and …. Well you see the point.

Drop some of the colour or spatial resolution in an image and it looks the same of at least very similar to the eye.

Summary

Claude Shanon largely started an entirely new discipline, INFORMATION THEORY, with papers around 1948.

We now use Shanon’s ideas about the difference between message, data, and information to remove redundancies in data.

A shorter string of 1s and 0s can carry the same information and the original string of 1s and 0s can be recovered.

Applies to all kinds of data but is particularly important for images.

Unlike most data, images can be further compressed if one is willing to live with a reduction in the quality.