Introduction (Part 2)

CO3096/7096

Modelling and Coding

- We will see that most files cannot be compressed.
- To compress files, one must make assumptions about the file.
  - These assumptions, in mathematical form, are the compressors model.
- When the file fits the model: compression.
  - When the file does not fit the model: no or negative compression.

Modelling and Coding

- Know the kind of redundancy in your raw data (data model)
  - Detailed model:
    - Better compression on a smaller set of files.
  - Less detailed model:
    - Poorer compression on a larger set of files.
  - ‘All-rounders’ usually beaten by a ‘specialist’.
- Design a method for getting rid of the redundancy (data coding)

Data models for text files

1. File containing text.
   - Some characters appear more frequently than others.
2. File containing English text.
   - ’e’ is most frequent, ’t’, ’a’, ’i’, ’o’, ’n’, ’s’ etc are roughly next most frequent. ’th’ often followed by ’e’ · · ·
3. File containing plays by Shakespeare.
   - ‘e’ is most frequent, ’t’, ’a’, ’i’, ’o’, ’n’, ’s’ etc are roughly next most frequent. ’th’ often followed by ’e’, unless it is at the end of a word (“droppeth”). Some odd words may occur frequently e.g. “Hamlet”, “Macbeth”, “exeunt” · · ·

(3) probably too detailed.

"Memoryless" model

- Used primarily for symbolic data; model contains only occurrence frequencies of individual symbols.

A 0.057305	H 0.042915	O 0.058215	V 0.009882
B 0.014876	I 0.053475	P 0.021034	W 0.007576
C 0.025775	J 0.002931	Q 0.000973	X 0.002264
D 0.026811	K 0.001016	R 0.048819	Y 0.011702
E 0.112578	L 0.031403	S 0.060289	Z 0.001502
F 0.022875	M 0.015892	T 0.078085	
G 0.009523	N 0.056035	U 0.018474	

[Occurrence statistics in the US constitution.]

- Generates random sequence with right frequencies.

- Coding: variable-length codes or entropy coding

Variable-length codes

- Symbols vary in frequency: don’t give all symbols the same length of code.
  - Frequent symbols: short code
  - Infrequent symbols: long code.
- Aim: minimise average code length, where average takes frequencies into account.

Example

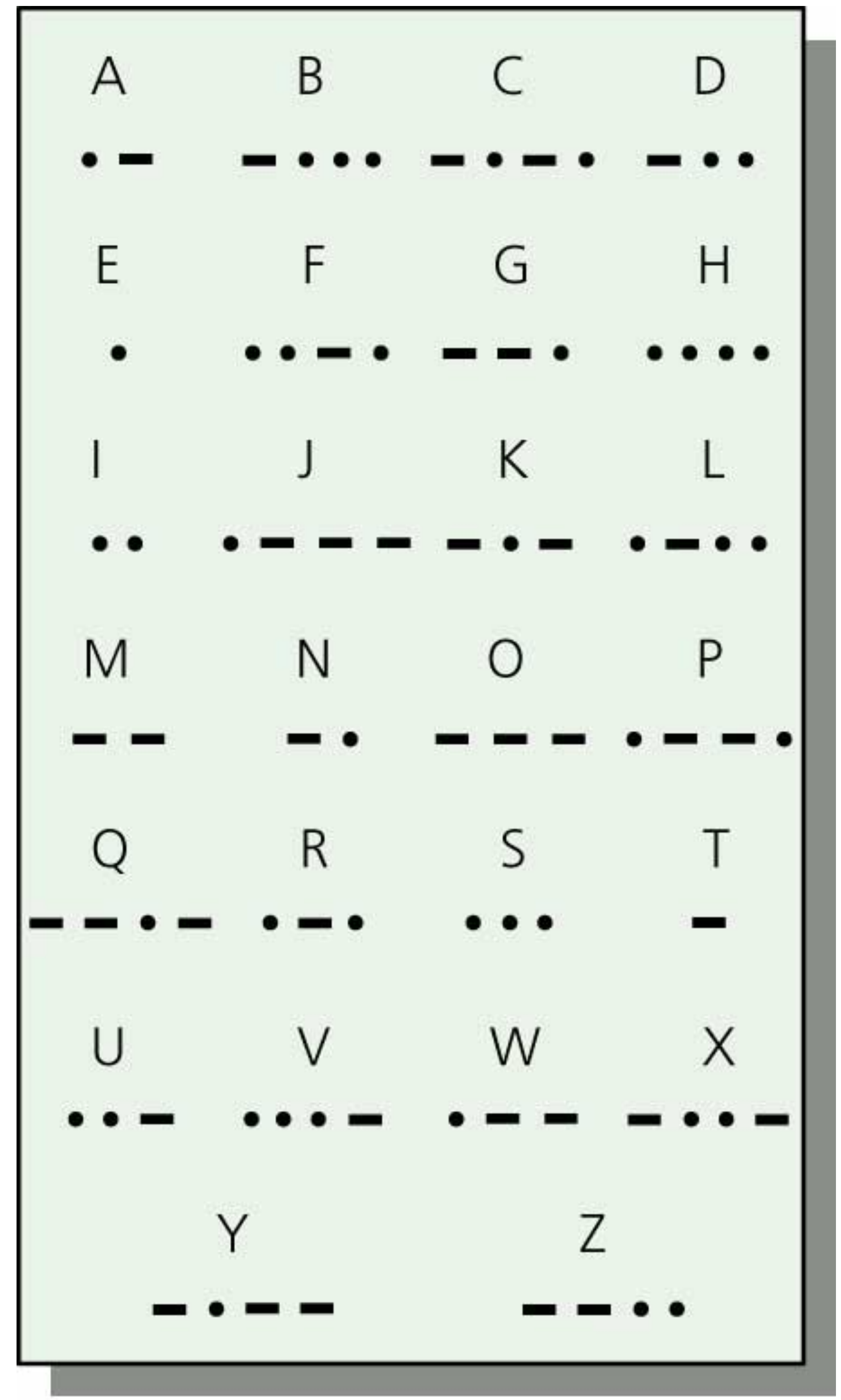
- Suppose there is a file containing the symbols A, C, G, T. Supposing their frequency of occurrence is as follows:
    - A (50%), C (25%), G (12.5%), T (12.5%)

A	00	A	0
C	01	C	10
G	10	G	110
T	11	T	111
Average = 2 bits		Average = 1.75 bits	
		(1+2+3+3)/4 = 2.25 (??)	

Morse code

A 0.057	H 0.043	O 0.058	V 0.010
B 0.015	I 0.053	P 0.021	W 0.008
C 0.026	J 0.003	Q 0.001	X 0.002
D 0.027	K 0.001	R 0.049	Y 0.012
E 0.113	L 0.031	S 0.060	Z 0.002
F 0.023	M 0.016	T 0.078	
G 0.010	N 0.056	U 0.018	

- A “dot” is one-third the length of a “dash”. Excluding spaces, the letter ‘Q’ takes ten times as long to transmit as the letter `E’



Academy Artworks

⠠⠛⠕⠕⠛⠇⠑

Sample output

- Here is some output from our memoryless model:

WHLTAESIHIPNFSETEELOTNRTEMTNEOPRERDDISIILNEE
MEACOFHOGSOUORSTNDSETUCTHNBVAARAYA

- Does not look like text at all. What's missing?

Markov models

- Likelihood of a symbol is considered fixed in memoryless model.
- However, likelihood of a symbol depends on context (preceding symbols):

```
_ s t a t i ?
```

- Context is modelled by Markov model
  - Much better compression for most symbolic data.
  - Formal definition later. Informally, model has frequencies of sequences of symbols (pairs, triples, words..)
  - Coding?

Adaptive vs. non-adaptive

- Model: Frequency counts for English text.

A 0.057305	H 0.042915	O 0.058215	V 0.009882
B 0.014876	I 0.053475	P 0.021034	W 0.007576
C 0.025775	J 0.002931	Q 0.000973	X 0.002264
D 0.026811	K 0.001016	R 0.048819	Y 0.011702
E 0.112578	L 0.031403	S 0.060289	Z 0.001502

…

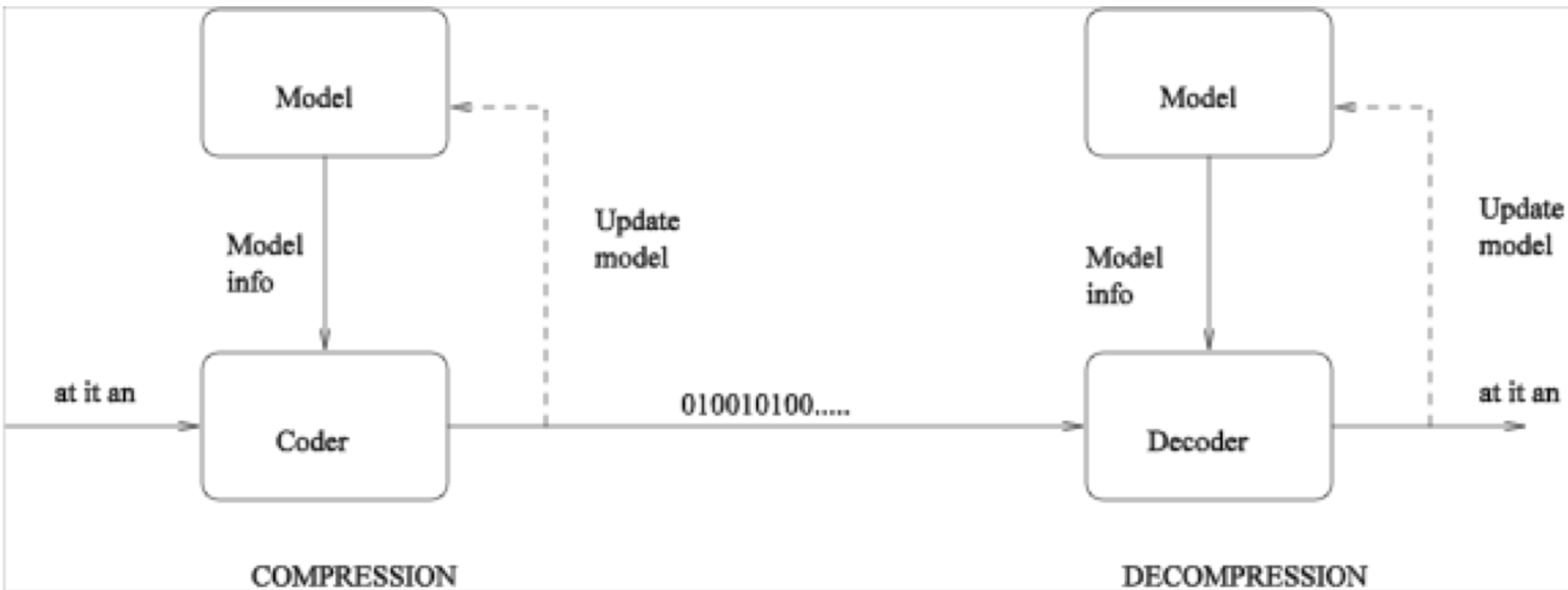
- What if text does not have these characteristics (e.g. text in Polish)?

- Need to send the model with the compressed file?

Adaptiveness

- adaptive: learn model parameters from input as it comes in bit by bit.
  - Often start with “empty” model;
  - Have start-up cost (which may be paid repeatedly) but are flexible;
  - Maintaining synchronization with de-coder is tricky (de-coder only sees compressed data).
- non-adaptive: the model is fixed.
- semi-adaptive: read entire input and get model parameters.
  - Most accurate model;
  - No start-up cost and flexible;
  - Normally must send model with the file;
  - Not always possible to read entire input.

Adaptive Compression



Summary

- Introduced (at a high level) a number of ways by which which symbolic data can be modelled and coded.
- Looked at adaptive vs non-adaptive algorithms.

Compressing Diffuse Data

- Models:
  - Models for diffuse data use mathematics that are beyond the scope of this module.
  - Cover some intuitions behind such models.
- Coding:
  - Predictive
  - Quantisation
  - Transform-based

Models for Images

Models for Images

Linear System Models

- Simple example:
  - Input is a sequence of values x_0, x_1, x_2, . . ., where

    $x_0 = g$, and
    $x_i = x_{i-1} + \varepsilon_i$, for $i = 1, 2, 3, \ldots$

    where g is any number and ε_i is a small random variable with mean value zero.

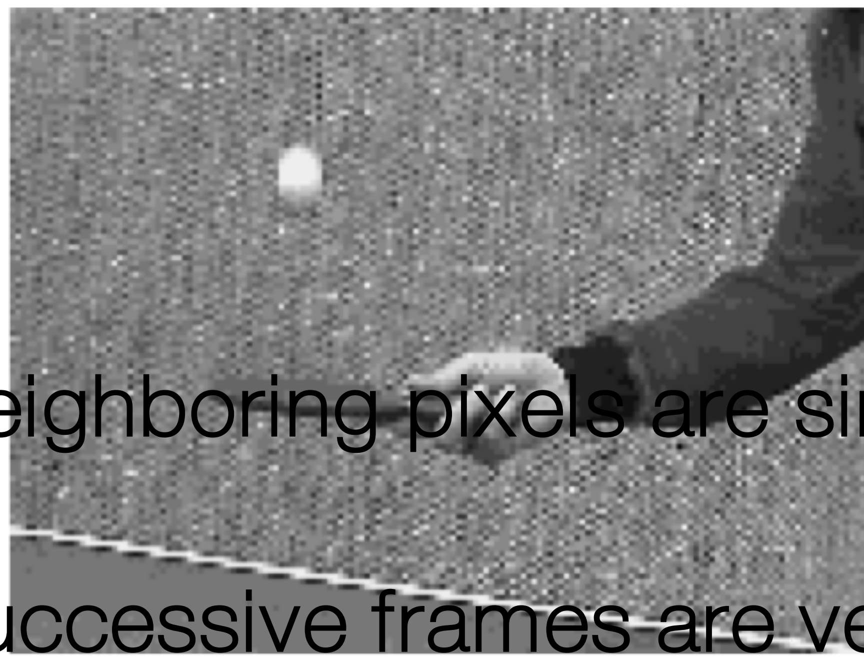
  - For example, if $g = 10$ this model could output:

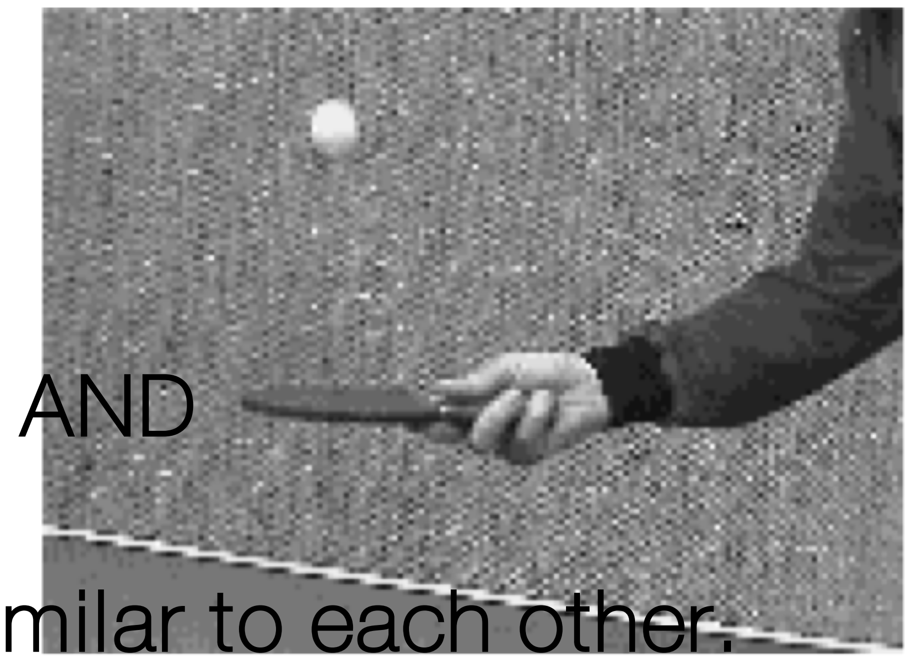
    10, 9.87, 9.95, 9.96, 10.04, 10.09, 10.07…

Predictive Coding

- Suitable for data modelled by a linear system: a sequence of roughly similar values.
- Encode differences between successive pixels/values: these are small/random.
    - (simplified view)

Video Model





- Neighboring pixels are similar AND
- Successive frames are very similar to each other.
  - “Motion-compensated” predictive coding

Summary

- Briefly considered a number of ways of modelling and coding diffuse data e.g.
    - linear system models and predictive coding
    - video models