COM3110/4115/6115:

Text Processing

Text Compression: Arithmetic coding

Rob Gaizauskas

Department of Computer Science

University of She eld

Rob Gaizauskas (U. She eld) COM3110/4115/6115: Text Processing 1 / 12

Overview

Models

Static

Semi-static Adaptive

Coding

Hu man Coding Arithmetic Coding

Further topics:

Symbolwise Models Dictionary Methods Synchronisation

Performance Issues

Rob Gaizauskas (U. She eld) [COM3110/4115/6115: Text Processing](#page1) 2 / 12

Arithmetic Coding

Arithmetic coding allows excellent compression

can code arbitrarily close to the entropy hence is optimal terms of compression

Wins over Hu man coding if distribution is very skewed

e.g. for two letter alphabet fa; bg where Pr [a] = :99 and Pr [b] = 0:01

a can be coded in log2Pr [s] = 0:015 bits

Hu man coding, however, requires at least one-bit/symbol same not true for arithmetic coding

Arithmetic coding suitable for sophisticated adaptive models Disadvantages:

slower than Hu man coding

not easy to start decoding in middle of compressed stream hence less suitable for full-text retrieval

(where random access to compressed text may be needed)

Thus: Hu man most useful for text; arithmetic coding for images

Rob Gaizauskas (U. She eld) [COM3110/4115/6115: Text Processing](#page1) 3 / 12



Arithmetic Coding (ctd)

Output of arithmetic coder is a stream of bits, but think of it as a fractional binary number between 0 and 1:

0.1011001

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| e.g. |  |  |  |  |  |  |  | 1 | + | 1 | + | 1 | + | 1 | =0:695 |  |
|  |  |  |  |  |  | 2 | 8 | 16 | 128 |  |
| 1 | 1 | 1 | 1 | 1 |  | 1 | 1 |  |  |  |  |  |  |  |  |  |
| / | / | / | / | / | | / | / |  |  |  |  |  |  |  |  |  |
| 2 | 4 | 8 | 16 | 32 | | 64 | 128 |  |  |  |  |  |  |  |  |  |

can drop the "0." pre x, as same on all outputs

Suppose we have a ternary alphabet fa; b; cg, and want to compress text bab

assume (static) model: Pr(a)=0.5 Pr(b)=0.4 Pr(c)=0.1

Arithmetic coder stores two numbers { high and low { representing a subinterval of [0; 1] used to code next symbol

initially high = 1 and low = 0

Rob Gaizauskas (U. She eld) [COM3110/4115/6115: Text Processing](#page1) 4 / 12



Arithmetic Coding (ctd)

Range between high and low sub-divided according to probability distribution of model: sub-intervals allocated for coding each symbol

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PR(C)=0.1 |  | 1.000 |  |  |
|  |  |  |  |
| PR(B)=0.4 |  | 0.900 | INTERVAL TO |  |
|  |  |  |
|  |  | ENCODE B |  |
|  |  |  |  |
| PR(A)=0.5 |  | 0.500 |  |  |
|  |  |  |  |

0.000

The coding step involves resetting the high/low values to narrow the recorded interval

here, to code b, set high = 0:9 and low = 0:5

Rob Gaizauskas (U. She eld) [COM3110/4115/6115: Text Processing](#page1) 5 / 12



Arithmetic Coding (ctd)

Coding continues by sub-dividing new current interval (between high/low ) and narrowing to sub-interval for coded symbol

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1.000 |  |  |  |  |  |  |
| PR(C)=0.1 | | |  |  |  |  |  |  |
|  |  | 0.900 |  |  |  | 0.900 |  |  |
|  |  |  | PR(C)=0.1 | |  | 0.860 |  |  |
|  |  |  |  |  |  |  |
|  |  |  | PR(B)=0.4 | |  |  |  |  |
| PR(B)=0.4 | | |  |  |  | 0.700 |  |  |
|  |  |  | PR(A)=0.5 | |  |  | INTERVAL TO |  |
|  |  |  |  |  | ENCODE A |  |
|  |  |  |  |  |  |  |  |
|  |  | 0.500 |  |  |  |  |  |  |
|  |  |  |  | 0.500 | |  |  |
|  |  |  |  |  |  |  |

PR(A)=0.5

0.000

here, set high = 0:7 and low = 0:5, to code symbol a

Rob Gaizauskas (U. She eld) [COM3110/4115/6115: Text Processing](#page1) 6 / 12



Arithmetic Coding (ctd)

Repeat process to code third symbol b:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 1.000 |  |  |  |  |  |  |  |  |  |  |
| PR(C)=0.1 | | |  |  |  |  |  |  |  |  |  |  |
|  |  | 0.900 |  |  |  | 0.900 |  |  |  |  |  |  |
|  |  |  | PR(C)=0.1 | |  | 0.860 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | PR(B)=0.4 | |  |  |  |  |  |  |  |  |
| PR(B)=0.4 | | |  |  |  | 0.700 |  |  |  | 0.700 |  |  |
|  |  |  |  |  |  |  | PR(C)=0.1 | |  | 0.680 | INTERVAL TO |  |
|  |  |  | PR(A)=0.5 | |  |  | PR(B)=0.4 | |  | 0.600 | ENCODE B |  |
|  |  |  |  |  |  |  | PR(A)=0.5 | |  |  |  |  |
|  |  | 0.500 |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 0.500 | |  |  | 0.500 | |  |  |
|  |  |  |  |  |  |  |  |  |

PR(A)=0.5

0.000

set high = 0:68 and low = 0:6, to code nal symbol b

nal interval represents full input bab

Rob Gaizauskas (U. She eld) [COM3110/4115/6115: Text Processing](#page1) 7 / 12

Arithmetic Coding (ctd)

Encoder has processed full input bab

coding process has narrowed interval so: high = 0:68, low = 0:6

Encoder now transmits this content by outputting any number in the range between high and low

choose number with shortest code (of course!)

e.g. choose 101 (0.625), not 10101 ( 0.656), though both in range

Transmitted code, plus model, is su cient for decoding

decoder simulates steps in encoding process and as it does, recovers encoded content

The smaller the nal interval is, the more bits that will be needed to specify a number that falls within it

e.g. for our example, low probability input ccc has nal interval width

0.001, and requires more bits to transmit, c.f. input aaa with interval

width 0.125

Rob Gaizauskas (U. She eld) [COM3110/4115/6115: Text Processing](#page1) 8 / 12

Arithmetic Coding (ctd)

Intuitively, method works because high probability events narrow the interval much less than low probability events do

The number of bits required is proportional to the negative logarithm of the size of the interval

the interval size is the product of the probabilities of the coded symbols the logarithm of this quantity is the sum of the logarithms of the

individual probabilities

symbol s of probability P[s] contributes log2P[s] bits to output

this is equivalent to the symbol's information content

Hence method is near-optimal

code size identical to the theoretical bound given by the entropy high probability symbols can be coded in a fraction of a bit

Rob Gaizauskas (U. She eld) [COM3110/4115/6115: Text Processing](#page1) 9 / 12

Arithmetic Coding (ctd)

Method is limited in practice by:

need to transmit, eventually, a whole number of bits/bytes limited precision arithmetic

As described, method produces no output until encoding complete i.e. until all input processed

In practice, possible to output bits during coding

avoids having to work with higher and higher precision numbers

key observation: when range is small, high/low have common pre x

e.g. if range is high = 0:6667, low = 0:6334 might:

output bits for pre x (here 6)

reset range as high = 0:667, low = 0:334

allows output to be generated incrementally

Rob Gaizauskas (U. She eld) [COM3110/4115/6115: Text Processing](#page1) 10 / 12

Arithmetic Coding (ctd)

Arithmetic coding more commonly used with adaptive modelling

probabilities used based on counts observed in text

Consider earlier example with alphabet fa; b; cg, coding input bab:

initialiase counts to 1 (avoid zero-frequency problem)

initial probabilities then: Pr (a) = 13 Pr (b) = 13 Pr (c) = 13

this model used to code rst character b | then counts updated updated model then: Pr (a) = 14 Pr (b) = 24 Pr (c) = 14

after 2nd char a coded, model: Pr (a) = 25 Pr (b) = 25 Pr (c) = 15

after 3nd char b coded, model: Pr (a) = 26 Pr (b) = 36 Pr (c) = 16 and so on . . .

see Witten et al. for a worked example

As before, given output, decoder can simulate the encoding process, including all the counting/model update

Rob Gaizauskas (U. She eld) [COM3110/4115/6115: Text Processing](#page1) 11 / 12

Reading

I. H. Witten, A. Mo at, T. C. Bell. Managing Gigabytes: Compressing and Indexing Documents and Images, 2nd ed. Morgan Kaufmann. 1999.

Baeza-Yates and Ribeiro-Neto, Modern Information Retrieval, Addison Wesley

Nam Phamdo. Theory of Data Compression. www.data-compression.com/theory.html

Rob Gaizauskas (U. She eld) [COM3110/4115/6115: Text Processing](#page1) 12 / 12