Arithmetic coding

Irina Bocharova,
University of Aerospace Instrumentation,
St.-Petersburg, Russia

Outline

- Shannon-Fano-Elias coding
- Gilbert-Moore coding
- Arithmetic coding as a generalization of SFE and GM coding
- Implementation of arithmetic coding

Let $x \in X = \{1, \ldots, M\}$, $p(x) > 0$,
$p(1) \geq p(2) \geq \cdots \geq p(M)$.
The cumulative sum is associated with the symbol x

$$Q(x) = \sum_{a<x} p(a),$$

that is,
$Q(1) = 0$,$Q(2) = p(1), \ldots, Q(M) = \sum_{i=1}^{M-1} p(i)$.

Then $\lfloor Q(m) \rfloor_{l_m}$ is a codeword for m,
where $l_m = -\lceil \log_2 p(m) \rceil$

x	$p(x)$	Q	Q in binary	$l(x)$	codeword
1	0.6	0	0.0	1	0
2	0.3	0.6	0.1001...	2	10
3	0.1	0.9	0.1110...	4	1110

$L = 1.6$ bits $H(X) = 1.3$ bits

If l_m binary symbols have been already transmitted then the length of the interval of uncertainty is 2^{-l_m}. Thus we can decode uniquely if

$$2^{-l_m} \leq p(m)$$

or

$$l_m \geq -\log_2 p(m)$$

Choosing length l_m we used only right segment with respect to the point $Q(m)$. This segment is always shorter than the corresponding left segment since symbol probabilities are ordered in descending order.

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

Let $x \in X = \{1, \ldots, M\}$, $p(x) > 0$.
The cumulative sum is associated with the symbol x

$$Q(x) = \sum_{a<x} p(a),$$

that is,
$Q(1) = 0$, $Q(2) = p(1), \ldots, Q(M) = \sum_{i=1}^{M-1} p(i)$.

Introduce $\sigma(x) = Q(x) + \frac{p(x)}{2}$

Then $\widehat{\sigma}(m) = \lfloor \sigma(m) \rfloor_{l_m}$ is a codeword for m,
where $l_m = -\lceil \log_2(p(m)/2) \rceil$.

We put point $\sigma(m)$ to the center of the segment $Q(m)+p(m)/2$ and choose length of codeword in such a manner that if l_m binary symbols have been transmitted the length of the interval of uncertainty is less than or equal to $p(m)/2$.

x	$p(x)$	Q	σ	l	GM	ShFE
1	0.1	0.0	0.00001...	5	00001	0000
2	0.6	0.0001..	0.01100...	2	01	0
3	0.3	0.10110...	0.11011...	3	110	10

$L = 2.6$ bits $H(X) = 1.3$ bits

Let $i < j$ then $\sigma(j) > \sigma(i)$

$$\sigma(j) - \sigma(i) = \sum_{l=1}^{j-1} p(l) - \sum_{l=1}^{i-1} p(l) + \frac{p(j)}{2} - \frac{p(i)}{2}$$

$$= \sum_{l=i}^{j-1} p(l) + \frac{p(j) - p(i)}{2} \geq p(i) + \frac{p(j) - p(i)}{2}$$

$$\geq \frac{p(i) + p(j)}{2} \geq \frac{\max\{p(i), p(j)\}}{2}$$

Since $l_m = \lceil -\log_2 \frac{p(m)}{2} \rceil \geq -\log_2 \frac{p(m)}{2}$

we obtain

$$\sigma(j) - \sigma(i) \geq \frac{\max\{p(i), p(j)\}}{2} \geq 2^{-\min\{l_i, l_j\}}.$$

$$H(X) + 1 \leq L < H(X) + 2$$

When symbol-by-symbol coding is not efficient?

1. **Memoryless source**

For symbol-by-symbol coding

$$R = H(X) + \alpha,$$

where α is coding redundancy.

For block coding

$$R = \frac{H(X^n) + \alpha}{n} = \frac{nH(X) + \alpha}{n} = H(X) + \frac{\alpha}{n},$$

where $H(X^n)$ denotes entropy of n random variables.

If $H(X) << 1$ $R \geq 1$ for symbol-by-symbol coding. For binary memoryless source with $p(0) = 0.99$, $p(1) = 0.01$ $H(X) = 0.081$ bits and we can easily construct the Huffman code with $R = 1$ bit but it is impossible to obtain $R < 1$ bit.

2.**Source with memory**

$H(X^n) \leq nH(X)$ and

$$R = \frac{H(X^n) + \alpha}{n} \leq H(X) + \frac{\alpha}{n}.$$

$$R \to H_\infty(X)$$

when $n \to \infty$, $H_\infty(X)$ denotes entropy rate.

How to implement block coding?

Let $x \in X = \{1, \ldots, M\}$, and we are going to encode sequences $\mathbf{x} = (x_1, \ldots, x_n)$ which appear at the output of X during n consecutive time moments.

We can consider a new source X^n with symbols corresponding to the sequences $\mathbf{x} = (x_1, \ldots, x_n)$ of length n and apply any method of symbol-by-symbol coding to these symbols. We will obtain

$$R = \frac{H(X^n)}{n} + \frac{\alpha}{n},$$

where α depends on the chosen coding procedure.

The problem is **coding complexity**. The alphabet of the new source is of size M^n. For example, if $M = 2^8 = 256$ then for $n = 2$ $M^2 = 65536$, and for $n = 3$ $M^3 = 16777216$.

The arithmetic coding provides redundancy $2/n$ with complexity n^2.

Arithmetic coding is a direct extension of the Gilbert-Moore coding scheme.

Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be an M-ary sequence of length n. We construct the modified cumulative distribution function

$$\sigma(\mathbf{x}) = \sum_{\mathbf{a} \prec \mathbf{x}} p(\mathbf{a}) + \frac{p(\mathbf{x})}{2} = Q(\mathbf{x}) + \frac{p(\mathbf{x})}{2},$$

where $\mathbf{a} \prec \mathbf{x}$ means that $\mathbf{a}$ is lexicographically less than $\mathbf{x}$, $l(\mathbf{x}) = -\lceil \log_2(p(\mathbf{x})/2) \rceil$.

The code rate R is equal to

$$\frac{1}{n} \sum_{\mathbf{x}} p(\mathbf{x}) l(\mathbf{x}) = \frac{1}{n} \sum_{\mathbf{x}} p(\mathbf{x}) (\lceil \log_2 \frac{1}{p(\mathbf{x})} \rceil + 1)$$

$$< \frac{H(X^n) + 2}{n}$$

If the source generates symbols independently we obtain

$$R < H(X) + \frac{2}{n}.$$

For source with memory

$$R \to H_\infty(X)$$

when $n \to \infty$.

Consider

$$Q(\mathbf{x}_{[1,n]}) = \sum_{\mathbf{a} \prec \mathbf{x}} p(\mathbf{a}) =$$

$$\sum_{\mathbf{a}:\mathbf{a}_{[1,n-1]} \prec \mathbf{x}_{[1,n-1]}, a_n} p(\mathbf{a}) +$$

$$\sum_{\mathbf{a}:\mathbf{a}_{[1,n-1]} = \mathbf{x}_{[1,n-1]}, a_n \prec x_n} p(\mathbf{a}),$$

where $\mathbf{x}_{[1,i]} = x_1, x_2, \ldots, x_i$. It is easy to see that

$$Q(\mathbf{x}_{[1,n]}) = Q(\mathbf{x}_{[1,n-1]}) + \sum_{\mathbf{a}:\mathbf{a}_{[1,n-1]} = \mathbf{x}_{[1,n-1]}, a_n \prec x_n} p(\mathbf{a})$$

$$= Q(\mathbf{x}_{[1,n-1]}) + p(\mathbf{a}_{[1,n-1]}) \sum_{a_n \prec x_n} p(a_n/\mathbf{a}_{[1,n-1]}).$$

If the source generates symbols independently

$$p(\mathbf{a}_{[1,n-1]}) = \prod_{i=1}^{n-1} p(a_i),$$

$$\sum_{a_n \prec x_n} p(a_n/\mathbf{a}_{[1,n-1]}) = \sum_{a_n \prec x_n} p(a_n) = Q(x_n),$$

where $Q(x_i)$ denotes the cumulative probability for x_i.

0000

0001

0010

0011

0100

0101

0110

0111

1000

1001

...

We obtain the following recurrent equations

$$Q(\mathbf{x}_{[1,n]}) = Q(\mathbf{x}_{[1,n-1]}) + p(\mathbf{x}_{[1,n-1]})Q(x_n),$$

$$p(\mathbf{x}_{[1,n-1]}) = p(\mathbf{x}_{[1,n-2]})p(x_{n-1}).$$

Coding procedure

$\mathbf{x} = (x_1, \ldots, x_n)$

Initialization

$$F = 0; G = 1;\ Q(1) = 0;$$

for $j = 2 : M$

$$Q(j) = Q(j-1) + p(j-1);$$

end;

for $i = 1 : n$

$$F \leftarrow F + Q(x_i) \times G;$$

$$G \leftarrow G \times p(x_i);$$

end;

$$F = F + G/2; l = -\lceil \log_2 G/2 \rceil; \hat{F} \leftarrow \lfloor F * 2^l \rfloor;$$

$X = \{a, b, c\}$,
$p(a) = 0.1$, $p(b) = 0.6$, $p(c) = 0.3$

$\mathbf{x} = (bcbab)$, $n = 5$

i	x_i	$p(x_i)$	$Q(x_i)$	F	G
0	-	-	-	0.0000	1.0000
1	b	0.6	0.1	0.1000	0.6000
2	c	0.3	0.7	0.5200	0.1800
3	b	0.6	0.1	0.5380	0.1080
4	a	0.1	0.0	0.5380	0.0108
5	b	0.6	0.1	0.5391	0.0065

Codeword length $-\lceil \log_2 G \rceil + 1 = 9$

$F + G/2 = 0.5423...$ and
codeword $\hat{F} = \lfloor F + G/2 \rfloor_l = 100010101$

$H(X) = 1.3$ bits $R = 1.8$ bit/symbol

At each step of the coding algorithm we perform 1 addition and 2 multiplications.
Let $p(1), \ldots, p(M)$ be numbers with binary representation of length d. Then at the first step F and G will be numbers with binary representaton of length $2d$. Next steps will require length of binary representation $3d, \ldots, nd$.

The complexity of coding procedure can be estimated as

$$d + 2d + \cdots + nd = \frac{n(n+1)d}{2}$$

PROBLEMS

1. Algorithm requires high computational accuracy (theoretically infinite)

2. Computational delay=length of the sequence to be encoded.

Decoding of Gilbert-Moore code

$Q(m)$, $m = 1, \ldots, M$ are known.
Input:$\hat{\sigma}$.

$$\textbf{Set } m = 1$$
$$\textbf{While } Q(m+1) < \hat{\sigma}\ m \leftarrow m+1$$
$$\textbf{end;}$$
$$\text{Output: } x(m)$$

Example.

$\hat{\sigma} = 0.01 \rightarrow \hat{\sigma} = 0.25$

$Q(2) = 0.1 < 0.25$ $m = 2$

$Q(3) = 0.7 > 0.25$ stop with $m = 2$.

Decoding procedure:

$$\hat{F} \leftarrow \hat{F}/2^l; S = 0; G = 1;$$

for $i = 1 : n$

$$j = 1;$$

while $S + Q(j+1) \times G < \hat{F}$ **and** $j \leq M$

$$j \leftarrow j + 1$$

end;

$$S \leftarrow S + Q(j) \times G;$$

$$G \leftarrow G \times Q(j);$$

$$x_i = j;$$

end;

At the ith step $G = p(\mathbf{x}_{[1,i]})$ and $S = Q(\mathbf{x}_{[1,i]})$.

a, b, c $p(a) = 0.1$, $p(b) = 0.6$, $p(c) = 0.3$

Codeword 0100010101 $\hat{F} = 0.541$

$\hat{F} = 0.0100010101$

S	G	Hyp.	Q	$S + QG$	x_i	p
0.0000	1.000	a	0.0	$0.0000 < \hat{F}$		
		b	0.1	$\mathbf{0.1000} < \hat{\mathbf{F}}$	b	0.6
		c	0.7	$0.7000 > \hat{F}$		
0.1000	0.6000	a	0.0	$0.1000 < \hat{F}$		
		b	0.1	$0.1600 < \hat{F}$	c	0.3
		c	0.7	$\mathbf{0.5200} < \hat{\mathbf{F}}$		
0.5200	0.1800	a	0.0	$0.5200 < \hat{F}$		
		b	0.1	$\mathbf{0.5380} < \hat{\mathbf{F}}$	b	0.6
		c	0.7	$0.6460 > \hat{F}$		
0.5380	0.1080	a	0.0	$\mathbf{0.5380} < \hat{\mathbf{F}}$		
		b	0.1	$0.5488 > \hat{F}$	a	0.1
0.5380	0.0108	a	0.0	$0.5380 < \hat{F}$		
		b	0.1	$\mathbf{0.5391} < \hat{\mathbf{F}}$	b	0.6
		c	0.7	$0.5456 > \hat{F}$		

1. $High < 0.5$

$Bit = 0;$

Normalization:

$Low = Low \times 2$
$High = High \times 2$
$Low = 0; High = 0.00011000001$
$Bit = 0; High = 0.0011000001$
$Bit = 0; High = 0.011000001$
$Bit = 0; High = 0.11000001$

2.$Low > 0.5$

$Bit = 1;$

Normalization:

$Low = Low - 0.5; Low = Low \times 2$
$High = High - 0.5;\ High = High \times 2$
$Low = 0.11000011$
$Bit = 1; Low = 0.1000011$
$Bit = 1; Low = 0.000011$

3. $Low < 0.5$ $High > 0.5$

0.011111...1

It can be 0.01111...10 or 0.10000...01

$Low = 0.0110 < 0.5$ $High = 0.1010 > 0.5$

Count=1;

Read next symbol

$Low = 0.10001 = 0.0110 + 0.00101$

$High = 0.10101$

Bit=1; Output: 10