Quotient Renormalization and Maximum Entropy

Hutter RNN Project

2026-01-31

1 Working in $\mathbb{N}$ (Log-Space)

Let $E = \{0, \ldots, 255\}$ be the event space (bytes). Define the bijection to $\mathbb{N}$:

$$t_e = \log_2(\text{count}_e)$$

This is the "thought" or support for event e in log-space.

2 The Quotient Map

The quotient $\pi : E \to E/\sim$ sends each byte to its ES class. The pushforward:

$$t_{[e]} = \log_2 \left(\sum_{e' \sim e} 2^{t_{e'}} \right) = \log_2(\text{count}_{ES})$$

From 100M bytes of enwik9:

ES	$t_{[es]}$	count
Digit	21.05	2,177,350
Punct	21.26	2,515,063
Vowel	24.69	27,071,333
Whitespace	23.80	14,652,205
Other	25.68	53,584,049

3 Fiber Structure

The within-ES structure lives in the fiber over each quotient point:

$$t_{e|es} = t_e - t_{[es]} = \log_2 P(e|es)$$

This decomposes the full thought:

$$t_e = t_{[es(e)]} + t_{e|es}$$

Bijecting back to E:

$$P(e) = P(es) \times P(e|es)$$

4 Maximum Entropy Renormalization

If we only know the quotient (ES-level) and assume max entropy within:

$$\tilde{P}(e) = P(es) \times \frac{1}{|es|}$$

This means $t_{e|es} = -\log_2(|es|)$ uniformly.

4.1 The Gap

ES	Size	H_{maxent}	H_{actual}	Gap
Digit	10	3.32	3.18	0.15
Punct	6	2.58	2.00	0.58
Vowel	10	3.32	2.43	0.89
Whitespace	4	2.00	0.40	**1.60**
Other	226	7.82	4.84	**2.98**

Weighted total:

$$H_{\text{maxent}}(\text{byte}|ES) = 5.52 \text{ bits} \tag{1}$$

$$H_{\text{actual}}(\text{byte}|ES) = 3.43 \text{ bits} \tag{2}$$

$$\text{Gap} = 2.09 \text{ bits} \tag{3}$$

5 Interpretation

5.1 What the Gap Means

The 2.09 bits/char is the **cost of renormalization**—the within-ES structure that max entropy throws away.

- **Whitespace**: 94% is space. Assuming uniform over 4 chars loses 1.6 bits.
- **Other**: 226 bytes, but ‘t’, ‘n’, ‘r’, ‘s’ dominate. Uniform loses 3 bits.
- **Digit**: Nearly uniform (‘0’-‘9’ all common). Gap only 0.15 bits.

5.2 The Factorization in $\mathbb{N}$

In log-space, the quotient factorization is *additive*:

$$t_e = t_{[es]} + t_{e|es}$$

This means learning decomposes:

1. Learn ES-level structure: $t_{[es]}$ (1.65 bits of entropy)
2. Learn within-ES structure: $t_{e|es}$ (3.43 bits of entropy)

Total: $1.65 + 3.43 = 5.08$ bits $= H(\text{byte})$. ✓

5.3 Why Max Entropy Fails

Max entropy assumes:

$$t_{e|es} = -\log_2(|es|) \quad \forall e \in es$$

But actual within-ES distributions are highly non-uniform:

- 'e' is 38% of vowels (should be 10% under max ent)
- Space is 94% of whitespace (should be 25%)
- 't' is 11% of Other (should be 0.4%)

6 Connection to ES Features

When we add ES features to the RNN input, we're giving it $t_{[es]}$ for free. But the 2.09 bits of within-ES structure must still be learned from the byte input.

The theoretical benefit of ES features:

$$\Delta H = H(ES) - H(ES|\text{prev}) = 0.19 \text{ bits}$$

Not 2.09 bits. The within-ES structure is *independent* of ES features.

7 Path Forward

To improve compression beyond ES:

1. Learn within-ES patterns (the 2.09 bits)
2. Or: define finer ESs that reduce the gap
3. Or: use hierarchical coding: ES first, then byte within ES

The current ES definition is poor for "Other" (226 bytes, 3 bits wasted). Should split into consonants, uppercase, symbols, etc.