Marginal Dominance in the UM Forward Pass

Claude and MJC

February 2026

1 Setup

We have a UM $u = (E, P)$ with four event spaces:

- I (byte_input, 256 events): current input byte
- O (byte_output, 256 events): predicted next byte
- M (marginal, 1 event): always active
- B (bigram, dynamic): joint input events born from $I \times O$

Three LPPs, all projecting onto O:

- $L_0 : M \to O$ with weights $w_0(b) =$ log-stochastic count of output byte b
- $L_1 : I \to O$ with weights $w_1(a, b) =$ log-stochastic count of (a, b) pair
- $L_2 : B \to O$ with weights $w_2(\beta, b) =$ log-stochastic count of (β, b) pair

The forward pass is the standard UM update $f_p(t)_j = \max_i \min(t_i, p_{ij})$. We set the input support to 255 for the active events in I, M, and B. Since $255 \geq w$ for all pattern weights, the min gives $\min(255, w) = w$, so the output support for byte b is:

$$s(O_b) = \max\big(w_0(b),\ w_1(\text{curr}, b),\ w_2(\beta, b)\big) \tag{1}$$

where the w_2 term is only present when a bigram event β is active.

Learning function ω_0 is log-stochastic counting: on observing a joint event (e_a, e_b), the support s for the corresponding LPP entry is incremented with probability 2^{-s}, giving $\mathbb{E}[s] \approx \log_2(\text{count})$ after many observations.

2 The Marginal Dominance Theorem

Theorem 1 (Marginal dominance)**.** *For any output byte b, input byte a, and bigram event β:*

$$w_0(b) \geq w_1(a, b) \geq w_2(\beta, b)$$

with equality only in degenerate cases.

Proof. Let $c_0(b)$ be the marginal count of byte b, $c_1(a, b)$ the joint count of (a, b), and $c_2(\beta, b)$ the count of (β, b).

Since every observation of (a, b) is also an observation of b:

$$c_0(b) = \sum_{a'} c_1(a', b) \geq c_1(a, b)$$

Since bigram β represents a specific (a', a) pair, and β is only active when that pair occurs:

$$c_1(a, b) = \sum_{\beta'} c_2(\beta', b) \geq c_2(\beta, b)$$

Log-stochastic counting preserves this ordering (in expectation): $\mathbb{E}[\text{support}] = \log_2(\text{count})$, and $\log_2$ is monotone.

Therefore $w_0(b) \geq w_1(a, b) \geq w_2(\beta, b)$ in expectation, and concentration is tight for large counts. □

Proposition 1 (Higher-order patterns never contribute)**.** *Under the max operation in equation (1), the L_0 term dominates: $s(O_b) = w_0(b)$ for all b, regardless of context. The unigram and bigram LPPs contribute nothing to the output support.*

Proof. By the marginal dominance theorem, $w_0(b) \geq w_1(a, b) \geq w_2(\beta, b)$. Since $\max(w_0, w_1, w_2) = w_0$, the higher-order terms are absorbed. □

3 Consequences

The output distribution is obtained by softmax on log-support:

$$p(b) = \frac{2^{s(O_b)}}{\sum_{b'} 2^{s(O_{b'})}}$$

By marginal dominance, $s(O_b) = w_0(b)$ for all b, so:

$$p(b) = \frac{2^{w_0(b)}}{\sum_{b'} 2^{w_0(b')}} \approx \frac{c_0(b)}{\sum_{b'} c_0(b')} = \hat{p}_{\text{marginal}}(b)$$

The model predicts the marginal byte distribution regardless of context. This is exactly what we observe: 5.3 bpc is consistent with the entropy of the marginal byte distribution on enwik9 (which has many rare bytes from XML markup, Unicode, etc., inflating entropy).

The model has strictly more information available (unigram and bigram contexts) but cannot use it because the max-min forward pass with absolute log-counts always lets the lower-order model dominate.

4 The Open Question

The UM forward pass $f_p(t)_j = \max_i \min(t_i, p_{ij})$ is monotonically increasing in each pattern's contribution: adding patterns can only increase output support, never decrease it. This means the UM cannot express "byte b is common overall but rare after this input"—precisely the conditional information that makes higher-order models useful.

KN smoothing solves this via discount: $p_k = (c - D)/\text{tc}$ subtracts evidence before normalizing. The KN quotient papers (kn-quotient.pdf, kn-quotient-v2.pdf) formalize this as division by GCD—removing common evidence shared between the prior and the conditional.

The question for the UM is: **where does the discount come from?**

Three possibilities:

4.1 Source support encodes specificity

Instead of setting all input supports to 255, the source support could encode how informative the context is. A marginal event (always true) is minimally informative—perhaps its support should be 1. An input byte event narrows to 1/256 of positions—perhaps support ≈ 8. A bigram event narrows further.

Under this scheme, $\min(\text{source}, w)$ would give:

- L_0: $\min(1, w_0(b)) = 1$ for all b with nonzero marginal
- L_1: $\min(8, w_1(a, b))$—capped at 8
- L_2: $\min(16, w_2(\beta, b))$—capped at 16

Now higher-order models CAN dominate when their pattern weights exceed the lower-order source support. The max would select the most specific applicable context. This is structurally similar to KN's interpolation chain where higher orders override lower orders when they have enough data.

The source support would be related to the self-information of the context: $\text{support}(\text{event}) = -\log_2 p(\text{event})$, which is the information gained by knowing the event is true.

4.2 Pattern weights encode conditional log-probability

Instead of log joint count $\log_2 c(a, b)$, the weight could be a scaled conditional: $w = \alpha \cdot \log_2(c(a, b)/c(a))$ for some scale α. This makes higher-order patterns competitive because $c(a, b)/c(a)$ can exceed $c(b)/N$ when the conditional probability exceeds the marginal.

However, this requires maintaining the marginal count $c(a)$ alongside the joint, and introduces the scale α as a free parameter. It also moves away from pure counting.

4.3 Evidence subtraction as a pattern operation

The KN discount $c - D$ can be seen as removing common evidence. In UM terms, this could be an explicit pattern operation: before applying a higher-order LPP, subtract the lower-order support from the output ES. This would require negative support or a two-pass forward pass (first apply and record, then subtract and re-apply).

This seems most consistent with the KN quotient interpretation (discount as GCD removal) but requires extending the UM with subtraction, which is not part of the standard max-min algebra.

5 Experimental Results

Data size	bpc	Bigram events	LPP entries (mg/uni/bi)
1,024	5.406	24	52 / 231 / 29
4,096	5.372	64	68 / 535 / 158
8,192	5.594	157	82 / 973 / 789
65,536	5.235	571	155 / 2068 / 4724
1,000,000	5.329	2058	195 / 5800 / 23325

The score is flat at $\sim$5.3 bpc across three orders of magnitude in data size. This confirms marginal dominance: the model learns richer structure (2058 bigram events, 23K bigram LPP entries at 1M) but cannot use it because the marginal LPP always dominates.

For comparison, KN-6 achieves 2.40 bpc on 1M bytes. The 2.9 bpc gap is entirely due to the inability of max-min with absolute log-counts to use contextual information.

6 Next Steps

The first approach (source support encodes specificity) is the most natural in UM terms and requires no new operations. The experiment: set marginal source support to $-\log_2(1) = 0$ (trivially true), input byte source support to $\log_2(256) = 8$ (one of 256 possible), bigram source support to $\log_2(256^2) = 16$ (one of 65536 possible).

If this works, it reveals that the UM's min operation already implements discount—the source support IS the discount, limiting how much evidence any single context can contribute. A highly specific context (high source support) can contribute more evidence than a vague one.