# Online Probability Model Estimation For Video Compression

Yue Sun[†‡], Jingning Han[†], Yaowu Xu[†]
[†]Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA, 94043, US
[‡] University of Washington
185 W Stevens Way NE
Seattle, WA, 98195, US
yuesun@uw.edu jingning,yaowu@google.com

## Abstract

Modern video codec uses arithmetic coding for entropy coding. The arithmetic coding asymptotically achieves the entropy bound provided the true probability distribution. Hence the compression efficiency heavily relies on the ability to capture the time-variant probability model in video signals. Variants of first-order linear probability model update schemes have been used in recent generation video codecs. Built on top of those, a multimodal estimation scheme that forms a higher order probability model update has been proposed in this work. We experimentally demonstrate its coding efficiency.

## 1 Introduction

Lossless entropy coding that aims for compressing a sequence of symbols in an informationally efficient way is a well-studied area in the last few decades. It is known that the lower bound of the length of the compressed sequence is the entropy of the original sequence [1], hence a good algorithm should generate a code whose length approaches the entropy. Denote a sequence $s$ of length $N$, the entropy associated with binary codewords is defined as

$$\sum_{t=1}^{N} -\log_2(p(s_t|s_{t-1},...,1)) := \sum_{t=1}^{N} -\log_2(p_t(s_t)), \tag{1}$$

where $p_t$ is the probability distribution of symbols at time $t$ conditioned on the previous observations. It is known that arithmetic code [2] optimally uses the probability to construct the codewords, which has been widely used in modern video codecs since [3][4].

In the context of video coding, the codec only receives a streaming sequence of symbols without knowing their probability distribution. The codec needs to estimate $p_t$ online. Denote the estimation by $\hat{p}_t$, the optimal code length approaches to $\sum_{t=1}^{N} -\log_2(\hat{p}_t(s_t))$.

If the codec estimates the probability well, i.e., making $\hat{p}_t$ close to $p_t$, the true code length approaches to the lower bound (1). Clearly probability estimation is crucial for video coding. The main difficulty however lies in that the probability is by nature time variant in video signals, which means we cannot estimate $p_t$ as a single number $p$ whose accuracy would improve with more samples observed. Instead, the probability estimator needs to properly track the underlying model on-the-fly with provided observations.

Typical solutions include a first-order linear update approach and its variations. Denote $\hat{p}_t \in \mathbb{R}^2$ as the probability estimation at time $t$, and $\hat{p}_t(0)$ and $\hat{p}_t(1)$ are probability of symbol 0 and 1 such that $\hat{p}_t(0) + \hat{p}_t(1) = 1$. In the CABAC framework [5]-[8], the probability model is updated as

$$\hat{p}(0)_+ = \alpha\hat{p}(0) + (1-\alpha)\mathbf{1}(s=0), \tag{2}$$

where $\alpha$ is a constant close to 0.95 and $\mathbf{1}(s=0)$ is 1 when $s=0$ and 0 otherwise. A barrier is set to prevent the estimated probability being too close to 0 or 1. The AV1 [9, 10] uses an adaptive $\alpha$ that depends on the number of symbols observed so far (denoted by SymNum):

$$\alpha = 2^{-(3+\mathbf{1}(SymNum>15)+\mathbf{1}(SymNum>31))}. \tag{3}$$

This update rule corresponds to a linear dynamical system, which is used for prediction of sequential data such as [11].

In this paper, we propose a multimodal probability estimation scheme. It is formed as a combination of the above first-order linear algorithms and their variations. The combination weights adapt to the prior observations in the format of maximum likelihood optimization. It is shown that the multimodal scheme effectively establishes a higher order system, which provides more flexibility over the first-order form to capture the time variant probability distributions. We conduct experiments on several datasets and demonstrate the compression efficiency gains.

## 2 Multimodal Probability Estimation

In this work, we propose and study several multimodal probability estimation schemes at various processing complexity.

### *2.1 Maximum Likelikhood Estimation*

In addition to the first-order linear estimator, we first consider a maximum likelihood estimate (MLE) of i.i.d symbols based on counting as derived next. We consider the binary random variable case for notion simplicity.

**Theorem 1.** *Suppose $s_1...s_t$ is i.i.d Bernoulli where 0 happens with probability $p$. Suppose we do not have any preference of $p$, i.e., the prior of $p$ is $U[0,1]$. From our observation of the sequence, if 0 happens $k$ times and 1 happens $l$ times, $\hat{p} = \frac{k+1}{k+l+2}$ is the estimator that satisfies*

$$\underset{\hat{p}}{\text{argmin}} \ -\mathbf{E}_p\left(p\log(\hat{p}) + (1-p)\log(1-\hat{p})\right) \tag{4}$$

**Proof.** The aprior probability $f$ satisfies

$$f(p|s_t...s_1) = \frac{f(s_t...s_1|p)f(p)}{f(s_t...s_1)} \tag{5}$$

Note $f(p) = 1$, $f(s_t...s_1)$ is a constant and $f(s_t...s_1|p) = p^k(1-p)^l$, so

$$f(p|s_t...s_1) = p^k(1-p)^l / \int_0^1 p^k(1-p)^l dp. \tag{6}$$

We consider minimizing

$$-\mathbf{E}_p\left(p\log(\hat{p})+(1-p)\log(1-\hat{p})\right)=\int_0^1\left(p\log(\hat{p})+(1-p)\log(1-\hat{p})\right)f(p|s_t...s_1)dp.$$

Take derivative over $\hat{p}$,

$$0=\nabla_{\hat{p}}\mathbf{E}_p\left(p\log(\hat{p})+(1-p)\log(1-\hat{p})\right)=\int_0^1\left(\frac{p}{\hat{p}}-\frac{1-p}{1-\hat{p}}\right)f(p|s_t...s_1)dp. \tag{7}$$

we have

$$\hat{p}=\int_0^1 pf(p|s_t...s_1)dp=\frac{\int_0^1 p^{k+1}(1-p)^l dp}{\int_0^1 p^k(1-p)^l dp}=\frac{k+1}{k+l+2}. \tag{8}$$

We then extend the MLE approach to track probabilities conditioned on the previous $2^\tau$ observations. A list of size $2^\tau$ stores the estimated probabilities conditioned on context sequences $s_{t-1}:s_{t-\tau}$. When a symbol arrives, we look at its previous $\tau$ symbols and use it as the context to fetch the corresponding probability in the list and update the count. The probability estimation follows (8). To code each symbol, we combine the outcomes from MLE and CABAC by taking their average. When the context has too few observations, the estimation is not stable. In such case, we fall back to the CABAC update approach. We summarize the algorithm in Algorithm 1 and refer to it as Multimodal Fixed, since the weight coefficients are pre-decided.

### *2.2 Multimodal Estimation*

Now that we have a collection of estimation methods including variants of the first-order linear update (2) and the MLE. All produce fairly good estimate. A natural question raised here is whether they could be combined for further improved prediction accuracy.

Related work that combines two first-order linear estimators has been proposed in [12], where it keeps two first-order linear estimators (2) with different update rate $\alpha$. The effective probability fed into the arithmetic coder is the average of the two outcomes, which serves as a balance between fast and slow update rates.

In this work, we look at the above problem from an alternative perspective. Let's consider a general weighted average of two probability updates

$$q_{t+1}=aq_t+(1-a)u_t,\quad r_{t+1}=br_t+(1-b)u_t,\quad p_t=wq_t+(1-w)r_t. \tag{9}$$

Using $z$ transform, we have

$$zQ=aQ+(1-a)U,\quad zR=bR+(1-b)U,\quad P=wQ+(1-w)R. \tag{10}$$

By sovling $P$ we get

$$P=(\frac{w(1-a)}{z-a}+\frac{(1-w)(1-b)}{z-b})U. \tag{11}$$

The inverse $z$ transform gives

$p_{t+1}=(a+b)p_t-abp_{t-1}+(w(1-a)+(1-w)(1-b))u_t+(ab-(1-w)a-wb)u_{t-1}$.

Clearly this is a second order system that subsumes (2) if $a = b = 0.95$. Otherwise it cannot be trivially reduced to a first order system that only involves $p_{t+1}, p_t, u_t$. This observation premises the fusion of a bigger class of the simple update models by optimizing the parameters e.g., $a, b, w$.

Denote $n_p$ as the number of kernels, let $\hat{p} \in \mathbb{R}^{n_p \times 2}$, where each row is a probability distribution (for binary random variable), and $w \in \mathbb{R}^{n_p}$ is the weight of the linear combination. We have $w^T\hat{p} = \sum w_i \hat{p}(i,:)$, a weighted average of $n_p$ simple probability estimators as the final estimation. We update each row of $\hat{p}$ using its own update algorithm, and fix $p(1,:)$ to be the baseline AV1 update approach (3). Hence the AV1 algorithm corresponds to the case $w_0 = 1, w_i = 0, \forall i \geq 2$. We use it as the initialization of linear weights in our algorithm.

Next we consider updating $w$. Since all update algorithms we choose as kernels should be "good", they should have positive correlation with our output $w^T\hat{p}$, so we constrain $w \geq 0$. Each probability estimation is non-negative. We further constrain $\mathbf{1}^T w = 1$. We use stochastic gradient descent (SGD) to update $w$. For each $s_t$, we incur the entropy

$$f(w, \hat{p}; s_t) = -\log_2((w^T\hat{p})(s_t)), \tag{12}$$

and we take gradient with respect to $w$,

$$\nabla_w f(w, \hat{p}; s_t) = -\frac{c}{(w^T\hat{p})(s_t)}\hat{p}(:, s_t), \ c = 1/\log(2). \tag{13}$$

At time $t$, we use step size $\eta_t = \eta_0/t$ which is standard for SGD ($\eta_t = \eta_0/t^r$, $r \in (0, 1]$ are allowed, and stochastic approximation defines $r \in (1/2, 1]$ [13]) and update $w$ by gradient step

$$w \leftarrow \underset{w_+ \geq 0, \mathbf{1}^T w_+ = 1}{\operatorname{argmin}} \ \|w_+ - (w - \eta_t \nabla_w f(w, \hat{p}; s_t))\|^2. \tag{14}$$

The detailed operation can be found in Algorithm 2. We denote this algorithm as Multimodal SGD. In practice we can also use a fixed step size $\eta = \eta_0$ to get a inner loop argument iterate $\tilde{w}_t$, and plug in for final probability estimation $w_t$ which satisfies $w_t = (\sum_{i=1}^t \tilde{w}_i)/t$ [14], which helps cancel the noise term in SGD. We also propose a linear dynamic update for variable iterate $w_{t+1} = \beta w_t + (1-\beta)\tilde{w}_t$ as a faster approach.

We can reduce the number of calls for to updating weight $w$, by using a batch version as shown in Algorithm 2 (referred as Multimodal Batch). Each epoch, we take a batch with increasing size $1, 4, 9, 16...$, and average the gradient in this batch. We update $w$ only at the end of each batch, with a fixed step size $\eta_0$. Empirically, the convergence rate of Algorithm 2 SGD and batch versions are similar. We also propose a fast algorithm that approximately solves the optimization problem.

**Fast projected optimization.** We consider the problem

$$\underset{w_+ \geq 0, \mathbf{1}^T w_+ = 1}{\operatorname{argmin}} \ \|w_+ - (w - \eta_t \nabla_w f(w, \hat{p}; s_t))\|^2. \tag{15}$$

To simplify notation, we consider $\underset{x \geq 0, \mathbf{1}^T x = 1}{\operatorname{argmin}} \frac{1}{2}\|x - y\|^2$. We write the Lagrangian

$$L(x, \lambda, \mu) = \frac{1}{2}\|x - y\|^2 - \lambda^T x + \mu(\mathbf{1}^T x - 1). \tag{16}$$

The KKT condition is

$$\nabla_x L(x,\mu) = x - y - \lambda + \mu\mathbf{1} = 0;\ \lambda \geq 0;\ x \geq 0;\ \lambda_i x_i = 0, \forall i. \tag{17}$$

The optimal $x$ is $x_i^* = \max(y_i - \mu, 0)$.

So we solve

$$\max_{\mu} \frac{1}{2}\|\max(y - \mu\mathbf{1}, \mathbf{0}) - y\|^2 + \mu(\mathbf{1}^T \max(y - \mu\mathbf{1}, \mathbf{0}) - 1) \tag{18}$$

to get $\mu^*$, and $x^* = \max(y - \mu^*\mathbf{1}, \mathbf{0})$. Since (18) is 1-dimensional convex optimization problem, we can solve it by binary search. If there are $n_p$ baseline algorithms and we compute the optimizer $w^*$ up to error $\epsilon$, the complexity each step is $O(n_p \log(1/\epsilon))$ ($n_p$ always exists since it's the complexity of evaluating the function).

## 3 Experimental Results

We evaluate the estimation efficiency of the proposed Multimodal Fixed, Multimodal SGD, and Multimodal Batch methods in terms of the codeword length, as compared to the probability estimation schemes (2) used in CABAC and (3) in AV1 baseline.

We first use synthetic data to evaluate its efficiency in tracking the underlying probability. A test vector is formed as multiple fixed-length chunks of random binary variable outcomes. Each chunk has a constant underlying probability distribution to produce the random binary symbols. This probability model varies across chunks. Now between test vectors, we change the chunk size from 50 to 1000 symbols, which effectively reflects how frequent the underlying model would change.

The code length reduction relative to the update approach (2) used in CABAC is shown in Fig. 1. The horizontal axis marked as iteration refers to the observed symbols. When a new symbol arrives, one would run an iteration of probability model update to estimate the current model for the entropy coding of next symbol. The vertical axis shows the relative code length reduction in bits. A positive number means smaller code length.

Clearly the proposed Multimodal SGD tracks the varying probability model well and hence outperforms the first-order linear update when the probability model shifts frequently. When the underlying model remains stable, i.e. identical model over a large chunk size (e.g. 1000 symbols), both the multimodal scheme and first-order linear update would provide a fairly accurate estimate. In this setting, the multimodal scheme is expected to perform slightly worse due to its higher model complexity (degree of freedom) as compared to (2).

| algorithm/chunk size | 50 | 100 | 200 | 1000 |
|---|---|---|---|---|
| Entropy | 2875 | 2894 | 2811 | 2830 |
| Multimodal SGD | 3440 | 3298 | 3096 | 2929 |
| CABAC | 3772 | 3487 | 3128 | 2929 |

Table 1: The codeword length under various synthetic models. The optimal entropy is coded by the underlying probability used to generate the synthetic data.

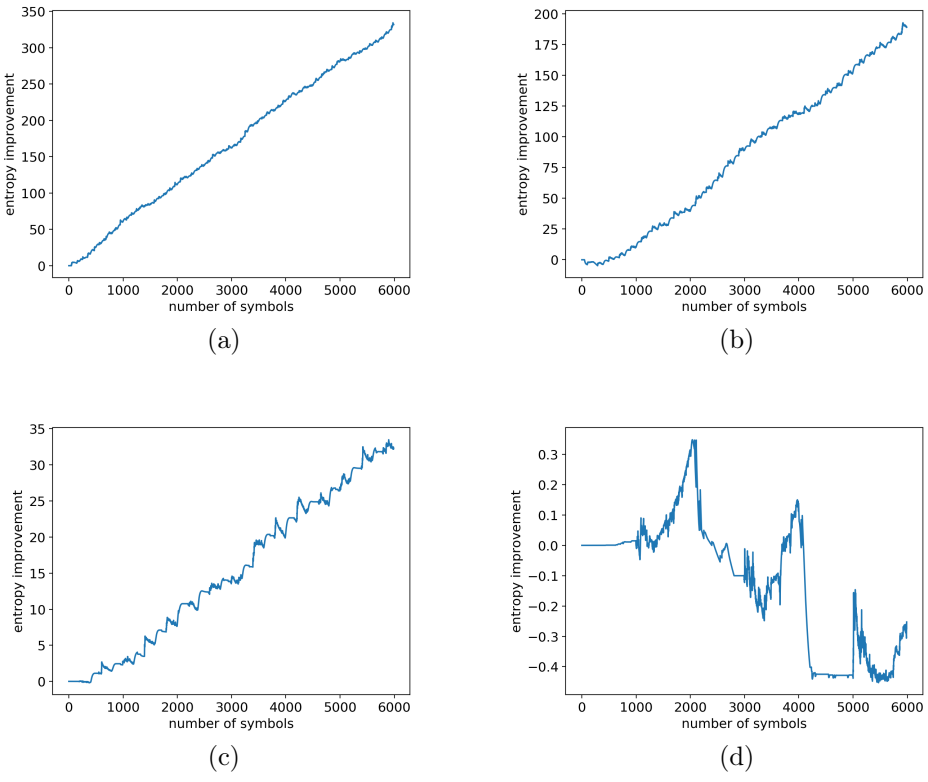


Figure 1: Codeword length reduction on synthetic data. Total length is 6000 symbols, we use chunk size (a) 50, (b) 100, (c) 200, (d) 1000 and generate symbol with Bernoulli 0.01,0.3 alternatively. We plot the improvement of entropy of the proposed Multimodal SGD (Algorithm 2) as compared to CABAC.

Next we apply the proposed multimodal schemes to the data extracted from the compressed video. We encode the video clips using the AV1 encoder at high complexity and high compression efficiency mode (speed 1, two pass encoding). The source code can be found at [15]. We extract the binary outcomes for whether a 8x8 transform block is coded as all zero coefficients under context index 1. In other words, this is the random binary sequence after context modeling, which is fed into the arithmetic coder and probability model estimator. We compare the compression performance in terms of codeword length as shown in Table 2-6. The proposed multimodal schemes generally achieve better coding performance results. The coding gains vary across the statistical characteristics of the video signals.

## 4 Conclusions

A multimodal probability estimation framework is derived in this work. It is formed as a linear combination of multiple simple first-order linear update models, whose weight coefficients are adaptive to the observed data on-the-fly. The framework effective establishes a higher order estimation system that provides more flexibility to track the variation of the underlying probability distributions. It is demonstrated that the proposed scheme outperform conventional first-order update system, especially when

| algorithm/dataset | 200 | 400 | 800 | 1200 | 2000 | 2800 | 3600 | 5200 |
|---|---|---|---|---|---|---|---|---|
| **Multimodal Fixed** | 1368 | 2587 | 2940 | 3860 | 3770 | 3709 | 3465 | 2388 |
| **Multimodal SGD** | **1364** | **2571** | **2930** | **3822** | **3733** | **3671** | **3433** | 2363 |
| **Multimodal Batch** | 1375 | 2577 | **2930** | 3827 | 3734 | 3673 | 3437 | **2358** |
| CABAC | 1375 | 2592 | 2951 | 3873 | 3789 | 3727 | 3476 | 2401 |
| AV1 | 1382 | 2580 | 2939 | 3843 | 3760 | 3698 | 3455 | 2380 |

Table 2: The test clip is cheer at SIF. The bit-rate used to generate the bit-stream is shown in the first row. The numbers in the following rows show the codeword length needed to compress the extracted binary sequences provided the probability model estimation scheme.

| algorithm/dataset | 200 | 400 | 800 | 1200 | 2000 | 2800 | 3600 | 5200 |
|---|---|---|---|---|---|---|---|---|
| **Multimodal Fixed** | 215 | 267 | 477 | 726 | 623 | 472 | 322 | **184** |
| **Multimodal SGD** | 214 | **265** | 474 | 719 | 613 | **468** | 320 | 185 |
| **Multimodal Batch** | **213** | **265** | **470** | **717** | **612** | 469 | **319** | 186 |
| CABAC | 217 | 269 | 481 | 732 | 627 | 473 | 323 | 186 |
| AV1 | 215 | 267 | 475 | 726 | 619 | 472 | 321 | **184** |

Table 3: The test clip is harbour at CIF. The target bit-rate is shown in the first row. The numbers in the following rows show the codeword length needed to compress the extracted binary sequences provided the probability model estimation scheme.

the underlying model is highly time variant.


[1] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[2] J. Rissanen and G. Langdon, "Universal modeling and coding," *IEEE Transactions on Information Theory*, vol. 27, no. 1, pp. 12–23, 1981.

[3] C. Gonzales, "Dct coding of motion sequences including arithmetic coder," *ISO/IEC JCT1/SC2/WP8, MPEG*, vol. 89, p. 187, 1989.

[4] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still image data compression standard*. Springer Science & Business Media, 1992.

[5] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the h. 264/avc video compression standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 620–636, 2003.


| algorithm/dataset | 100 | 200 | 300 | 400 | 600 | 800 | 1000 |
|---|---|---|---|---|---|---|---|
| **Multimodal Fixed** | 1591 | 2317 | **2819** | 3155 | **3692** | **4122** | **3994** |
| **Multimodal SGD** | **1571** | **2306** | 2826 | **3147** | 3701 | 4130 | 4023 |
| **Multimodal Batch** | **1571** | **2306** | 2868 | 3155 | 3725 | 4133 | 4056 |
| CABAC | 1597 | 2333 | 2840 | 3177 | 3725 | 4161 | 4035 |
| AV1 | 1583 | 2316 | 2831 | 3156 | 3708 | 4139 | 4027 |

Table 4: The test clip is ice at 240p. The target bit-rate is shown in the first row. The numbers in the following rows show the codeword length needed to compress the extracted binary sequences provided the probability model estimation scheme.

| algorithm/dataset | 100 | 200 | 400 | 600 | 800 | 1200 | 2000 | 2800 |
|---|---|---|---|---|---|---|---|---|
| **Multimodal Fixed** | 842 | 1515 | 1728 | 1612 | 1848 | 1767 | 1792 | 1465 |
| **Multimodal SGD** | 836 | **1506** | 1706 | **1600** | **1814** | 1748 | 1770 | **1453** |
| **Multimodal Batch** | **835** | 1509 | **1705** | 1602 | 1816 | **1747** | **1768** | 1458 |
| CABAC | 845 | 1522 | 1731 | 1621 | 1855 | 1777 | 1802 | 1474 |
| AV1 | 838 | 1512 | 1718 | 1610 | 1837 | 1761 | 1784 | 1463 |

Table 5: The test clip is keiba at 240p. The target bit-rate is shown in the first row. The numbers in the following rows show the codeword length needed to compress the extracted binary sequences provided the probability model estimation scheme.

| algorithm/dataset | 200 | 400 | 800 | 1200 | 2000 | 2800 | 3600 | 5200 |
|---|---|---|---|---|---|---|---|---|
| **Multimodal Fixed** | 1178 | 1625 | 1948 | 1951 | 1449 | 984 | 653 | **488** |
| **Multimodal SGD** | **1167** | 1617 | **1932** | 1935 | **1430** | 980 | **649** | 492 |
| **Multimodal Batch** | 1170 | **1616** | 1934 | **1934** | 1431 | **977** | **649** | 493 |
| CABAC | 1186 | 1638 | 1951 | 1962 | 1459 | 992 | 658 | 492 |
| AV1 | 1174 | 1624 | 1938 | 1946 | 1445 | 981 | 652 | 491 |

Table 6: The test clip is soccer at CIF. The target bit-rate is shown in the first row. The numbers in the following rows show the codeword length needed to compress the extracted binary sequences provided the probability model estimation scheme.


[6] "Recommendation h. 264 and draft iso/iec 14496-10 avc. joint video team of iso," IEC JTC1/SC29/WG11 & ITU-T SG16/Q. 6 Doc. JVT-G050, T. Wieg, Ed., Pattaya, Tech. Rep., 2003.

[7] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[8] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[9] P. de Rivaz and J. Haughton, "Av1 bitstream & decoding process specification," *The Alliance for Open Media*, p. 182, 2018.

[10] Y. Chen, D. Murherjee, J. Han, A. Grange, Y. Xu, Z. Liu, S. Parker, C. Chen, H. Su, U. Joshi *et al.*, "An overview of core coding tools in the av1 video codec," in *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 41–45.

[11] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages," *International journal of forecasting*, vol. 20, no. 1, pp. 5–10, 2004.

[12] A. Alshin, E. Alshina, and J. Park, "High precision probability estimation for cabac," in *2013 Visual Communications and Image Processing (VCIP)*. IEEE, 2013, pp. 1–6.

[13] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[14] Y. Fang, J. Xu, and L. Yang, "Online bootstrap confidence intervals for the stochastic gradient descent estimator," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 3053–3073, 2018.

[15] "https://aomedia.googlesource.com."


## A Appendix – algorithms

**Algorithm 1** Entropy coding and probability estimation (Multimodal Fixed)

Fixed quantity: $\alpha = (\frac{0.01875}{0.5})^{1/63}$, $p_{62} = 0.5\alpha^{62}$, $w = 0.5$, $t_{\text{thres}} = 25$, $\tau = 5$ for binary and 8 for multi-symbol, $L = 2^\tau$, mode $= 0$.
Initialization in outer loop: List $= [[0, 0]]^L$.
**Def** ProbUpdate($\hat{p}_{\text{inf}}, \hat{p}_{\text{CABAC}}, \tau, s_{t-\tau} : s_t, t, \alpha, p_{62}, w, t_{\text{thres}}, \text{List}, \text{mode}$) **:**
  \\ $t$ is the current number of received symbols and $s_{t-\tau} : s_t$ is the last $\tau$ symbols.
  $\hat{p}_{\text{inf}} \leftarrow \text{ProbUpdateCount}(\hat{p}_{\text{inf}}, s_t, t)$.
  $\hat{p}_{\text{CABAC}} = \text{ProbUpdateCABAC}(\hat{p}_{\text{CABAC}}, s_t, \alpha)$.
  $t_{\text{tmp}} = 0$.
  **if** $t > \tau$ **then**
    $\text{List}(s_t; s_{t-\tau} : s_{t-1}) \leftarrow \text{List}(s_t; s_{t-\tau} : s_{t-1}) + 1$. $t_{\text{tmp}} \leftarrow \tau$
    **while** $t_{\text{tmp}} > 0$ and $\sum_i \text{List}(i; s_{t-t_{\text{tmp}}+1} : s_t) < t_{\text{thres}}$ **do**
      $t_{\text{tmp}} \leftarrow t_{\text{tmp}} - 1$.
    **end while**
    **if** $t_{\text{tmp}} > 0$ **then**
      $p_{\text{cond}} \leftarrow \frac{\text{List}(:; s_{t-t_{\text{tmp}}+1}:s_t)}{\sum_i \text{List}(i; s_{t-t_{\text{tmp}}+1}:s_t)}$.
    **end if**
  **end if**
  **if** $t_{\text{tmp}} > 0$ **then**
    **if** mode **then**
      $p \leftarrow w\hat{p}_{\text{inf}} + (1 - w)p_{\text{cond}}$.
    **else**
      $p \leftarrow w\hat{p}_{\text{CABAC}} + (1 - w)p_{\text{cond}}$.
    **end if**
  **else**
    **if** mode **then**
      $p \leftarrow \hat{p}_{\text{inf}}$.
    **else**
      $p \leftarrow \hat{p}_{\text{CABAC}}$.
    **end if**
  **end if**
  **return** $p, \hat{p}_{\text{inf}}, \hat{p}_{\text{CABAC}}, \text{List}$. \\ $p$ is used for encoding.
**Def** ProbUpdateCABAC($p, s, \alpha$) **:**
  Find LPS: $\sigma = \text{argmin}_{i \in \{0,1\}} p(i)$.
  **if** $s = \sigma$ **then**
    $p(\sigma) \leftarrow \max(\alpha p(\sigma), p_{62})$.
  **else**
    $p(\sigma) \leftarrow \alpha p(\sigma) + 1 - \alpha$.
  **end if**
  $p(1 - \sigma) \leftarrow 1 - p(\sigma)$.
  **return** $p$
**Def** ProbUpdateCount($p, s, t$) **:**
  $p = \frac{t-1}{t} p$.
  $p(s) = p(s) + \frac{1}{t}$
  **return** $p$

**Algorithm 2** Entropy coding and adaptive probability estimation (Multimodal SGD and Batch)

**Require:** Sequence of binary symbols $s \in \{0,1\}^N$ arriving online.

Initialization: $n_p = 18$, $\hat{p} = \mathbf{1}_{n_p} \cdot [0.5, 0.5]$. $w = \tilde{w} = w = [1, 0, ..., 0]^T \in \mathbb{R}^{n_p}$, $\alpha = 0.99 \cdot 2^{-[0:n_p-2]/4(n_p-2)} \in \mathbb{R}^{n_p-2}$, $\eta_0 = 5$, $r = 1$, $b_- = b = 0$, $g = \mathbf{0}^{n_p}$, $\beta = 0.95$. \\$\alpha_{\min} = 0.84$, $r \in (1/2, 1]$.

Choose mode from SGD decreasing step size, SGD average argument, SGD dynamic argument or SGD batch.

**while** $t \leq N$ **do**

Receive symbol $s_t$.

Encode $s_t$ by $w^T \hat{p}$.

$\hat{p}(1,:) \leftarrow \text{ProbUpdateAV1}(\hat{p}(1,:), s_t, t, 2)$.

$\hat{p}(2,:) \leftarrow \text{ProbUpdateCount}(\hat{p}(2,:), s_t, t)$.

$\hat{p}(i,:) \leftarrow \text{ProbUpdateCABAC}(\hat{p}(i,:), s_t, \alpha_{i-2})$ for all $3 \leq i \leq n_p$.

\\ Can wrap in any prabability estimate algorithms.

**if** SGD decreasing step size **then**

$w \leftarrow \underset{w_+ \geq 0, \mathbf{1}^T w_+ = 1}{\text{argmin}} \|w_+ - (w - \frac{\eta_0}{t^r} \cdot \frac{\hat{p}(:,s_t)}{(w^T \hat{p})(s_t)})\|^2$.

\\ Can be solved by fast projected optimization algorithm.

**else if** SGD average argument **then**

$\tilde{w} \leftarrow \underset{w_+ \geq 0, \mathbf{1}^T w_+ = 1}{\text{argmin}} \|w_+ - (\tilde{w} - \eta_0 \cdot \frac{\hat{p}(:,s_t)}{(\tilde{w}^T \hat{p})(s_t)})\|^2$.

$w \leftarrow (1 - \frac{1}{t+1})w + \frac{1}{t+1}\tilde{w}$. \\ $= \frac{1}{t+1}\sum_{i=1}^{t+1} \tilde{w}_i$

**else if** SGD dynamic argument **then**

$\tilde{w} \leftarrow \underset{w_+ \geq 0, \mathbf{1}^T w_+ = 1}{\text{argmin}} \|w_+ - (\tilde{w} - \eta_0 \cdot \frac{\hat{p}(:,s_t)}{(\tilde{w}^T \hat{p})(s_t)})\|^2$.

$w \leftarrow \beta w + (1 - \beta)\tilde{w}$.

**else if** SGD batch **then**

**if** $t \leq b$ **then**

$g \leftarrow g + \frac{1}{b - b_- + 1} \frac{1}{(w^T \hat{p})(s_t)} \hat{p}(:, s_t)$. \\ Batch from $b_-$ to $b$, size $1, 4, 9, 16, ...$

**end if**

**if** $t = b$ **then**

$w \leftarrow \underset{w_+ \geq 0, \mathbf{1}^T w_+ = 1}{\text{argmin}} \|w_+ - (w - \eta_0 g)\|^2$.

$g \leftarrow \mathbf{0}^{n_p}$.

$b_+ \leftarrow b + (\sqrt{b - b_- + 1} + 1)^2$, $b_- \leftarrow b + 1$, $b \leftarrow b_+$. \\ Update batch.

**end if**

**end if**

$t \leftarrow t + 1$.

**end while**

**Def** $\text{ProbUpdateAV1}(p, s, t, \text{NumOfSyms})$ **:**

$p_0 \leftarrow 0.0076$.

$r \leftarrow 3 + (t > 15) + (t > 31) + (\text{NumOfSyms} > 2) + (\text{NumOfSyms} > 4)$

$p \leftarrow \max((1 - 2^{-r})p, p_0)$.

$p(s) \leftarrow p(s) + 1 - \sum_{i=1}^{\text{NumOfSyms}} p(i)$

**return** $p$