Sequential Random Binning for Streaming Distributed Source Coding

Stark C. Draper, Cheng Chang, and Anant Sahai
Dept. of EECS, University of California, Berkeley, CA, 94720
{sdraper,cchang,sahai}@eecs.berkeley.edu

***Abstract*— Random binning arguments underlie many results in information theory. In this paper we introduce and analyze a novel type of causal random binning – "sequential" binning. This binning is used to get streaming Slepian-Wolf codes with an "anytime" character. At the decoder, the probability of estimation error on any particular symbol goes to zero exponentially fast with delay. In the non-distributed context, we show equivalent results for fixed-rate streaming entropy coding. Because of space constraints, we present full derivations only for the latter, stating the results for the distributed problem. We give bounds on error exponents for both universal and maximum-likelihood decoders.**

I. Introduction

Consider the "lossless" entropy coding of a discrete memoryless source. One approach is to use a fixed-length block code, and accept some probability of encoding error. Errors occur when the realized source sequence is sufficiently atypical that it is not indexed by the code. The probability of such an event can be made as small as desired by using a sufficiently long block length. This block-length induces an end-to-end system delay. An alternate approach is to use a variable-length code. These codes achieve a zero-error probability by using longer codewords to encode more atypical sequences. These codes are characterized by variable delay – for a fixed communication rate, the more atypical the source sequence, the more bits to encode, and therefore the longer the delay before decoding.

Both fixed and variable-length codes can be made universal over all stationary memoryless sources with an entropy lower than the target coding rate. For fixed-length codes, the encoder can simply "bin" the observed sequence. The decoder can then use a minimum empirical entropy rule to decode universally, without knowledge of source statistics. In the universal variable-length case, it is the encoder that traditionally does an explicit or implicit estimation of statistics so that it can assign longer codewords to less likely sequences.

Now consider lossless entropy coding in the context of Slepian-Wolf codes [6]. In Slepian-Wolf coding, we cannot use variable-rate codes to get a zero probability of error, even with known statistics. This is easiest to see by example. Suppose $\mathbf{x}$ is a sequences of independent identically distributed (i.i.d.) uniform binary random variables, related to $\mathbf{y}$ through a memoryless binary symmetric channel with crossover probability $\rho < 0.5$. The Slepian-Wolf sum-rate bound is $H(x, y) = 1 + H(\rho) < 2$. But, since the individual encoders only see uniformly distributed binary sources, they do not know when the sources are behaving jointly atypically. Therefore, they have no basis on which to adjust their encoding rates. For this reason, variable-rate approaches do not yield zero-error Slepian-Wolf coding.

Motivated by work in "anytime" channel coding [5], we ask whether we can design a streaming Slepian-Wolf system. We relax the demand for zero probability of error with a random delay (as in variable-length coding) and instead ask for an exponentially decreasing probability of error for all decoding delays. To build toward this goal, we introduce a sequential binning scheme in Section II. We use it to build a streaming fixed-rate universal entropy code. Using a sequential version of a minimum entropy decoding rule, the probability of decoding error decreases exponentially in the delay for all sources with entropies below the rate of the code. In Section III, we state our results for streaming Slepian-Wolf systems under both universal and maximum-likelihood (ML) decoding. Derivations will appear in [2]. Finally, in Section IV we discuss and illustrate some of the differences between streaming and block coding systems.

II. Streaming entropy coding via sequential random binning

Source Model: A sequence of i.i.d. random symbols, x_i, $i = 1, 2, \ldots$ is observed at the encoder. The distribution of each x_i is denoted by p_x, where $p_{x_i}(x) = p_x(x)$ for all i. At time l the encoder transmits a message m_l which is a function of $x^l = [x_1, x_2, \ldots, x_l]$ to the decoder where $m_l \in \{1, \ldots, \exp[R_x]\}$. For convenience we measure rate in nats and for simplicity we assume

that $R_x/\ln 2$ is an integer.[1]

Goal: For any time $n \geq n_0$ the decoder wants to make an estimate $\hat{x}^{n_0} = \hat{x}^{n-\Delta} = D_\Delta(m_1, m_2, \ldots, m_n)$, where Δ is the decoding delay. We want $\Pr[\hat{x}^{n-\Delta} \neq x^{n-\Delta}]$ to decay exponentially in Δ for all n and Δ.

Sequential Binning Encoder: The encoder works by randomly assigning "parity bits" in a causal manner to each possible source sequence. At each time step, each possible source sequence so far is assigned $R_x/\ln 2$ new parity bits where the parities are all Bernoulli-(0.5). Since parity bits are assigned causally, if two source sequences share the same length-l prefix, then their first $lR_x/\ln 2$ parity bits must match. Subsequent parities are drawn independently.[2] The set (or "bin") of sequences that share the same parities as the length-n source $\mathbf{x}$ is denoted $\mathcal{B}_x(\mathbf{x})$. Consider some other possible source sequence $\tilde{\mathbf{x}}$. If, for example, $\tilde{x}^l = x^l$, but $\tilde{x}_{l+1} \neq x_{l+1}$ then there are $n-l$ opportunities for the parities of $\tilde{\mathbf{x}}$ and $\mathbf{x}$ to differ. Therefore, by the parity generation process, $\Pr[\tilde{\mathbf{x}} \in \mathcal{B}_x(\mathbf{x})] = \exp\{-(n-l)R_x\}$.

Decoder: We use a "competitive minimum-entropy-suffix" decoding rule. The decoding rule $\hat{x}^{n-\Delta} = D_\Delta(m_1, \ldots, m_n)$ starts by first identifying candidate sequences whose parities match the received bit stream up to time n. If we observe $\mathbf{x} = \mathbf{x}$, this is $\{\bar{\mathbf{x}}$ s.t. $\bar{\mathbf{x}} \in \mathcal{B}_x(\mathbf{x})\}$. We score each of these sequences in a two-stage manner. We get $|\mathcal{B}_x(\mathbf{x})| - 1$ preliminary scores for each $\bar{\mathbf{x}} \in \mathcal{B}_x(\mathbf{x})$ by comparing it to each sequence $\bar{\bar{\mathbf{x}}} \in \mathcal{B}_x(\mathbf{x})$ according to the function

$$S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}}) = \begin{cases} l & \text{if } H(\bar{x}^n_{l+1}) \geq H(\bar{\bar{x}}^n_{l+1}) \\ & \text{and where } \bar{x}^l = \bar{\bar{x}}^l, \bar{x}_{l+1} \neq \bar{\bar{x}}_{l+1} \\ n & \text{else,} \end{cases} \quad (1)$$

where, e.g., $H(\bar{x}^n_{l+1})$ is the empirical suffix entropy of $\bar{\mathbf{x}}$. If this entropy is greater than or equal to that of $\bar{\bar{\mathbf{x}}}$, its score is the length l of the longest shared prefix. Else, if its entropy is strictly less, its score is n.[3] The final score of each sequence is the minimum of all its preliminary scores

$$S(\bar{\mathbf{x}}) = \min_{\bar{\bar{\mathbf{x}}} \in \mathcal{B}_x(\mathbf{x}), \bar{\bar{\mathbf{x}}} \neq \bar{\mathbf{x}}} S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}}). \quad (2)$$

A sequence gets a high score if it is only beaten by sequences that share long prefixes with it. Each sequence $\bar{x} \in \mathcal{B}_x(\mathbf{x})$ is assigned a final prefix defined as

$$\pi(\bar{\mathbf{x}}) = \bar{x}^{S(\bar{\mathbf{x}})}. \quad (3)$$

A sequence's prefix $\pi(\bar{\mathbf{x}})$ is the the shortest longest prefix that $\bar{\mathbf{x}}$ shares with any other sequence $\bar{\bar{\mathbf{x}}}$ that is also in its bin, and which also has a lower suffix entropy. If $S(\bar{\mathbf{x}}) \leq S(\bar{\bar{\mathbf{x}}})$, and $\bar{x}^{S(\bar{\mathbf{x}})} = \bar{\bar{x}}^{S(\bar{\mathbf{x}})}$, i.e., $\pi(\bar{\mathbf{x}})$ is a prefix of $\pi(\bar{\bar{\mathbf{x}}})$ then we use the notation $\pi(\bar{\mathbf{x}}) \sqsubseteq \pi(\bar{\bar{\mathbf{x}}})$ to denote this subsequence relationship. In fact, we show in Lemma 1 that if $S(\bar{\mathbf{x}}) \leq S(\bar{\bar{\mathbf{x}}})$ then $\pi(\bar{\mathbf{x}}) \sqsubseteq \pi(\bar{\bar{\mathbf{x}}})$.

We now define the maximum-scoring sequence as

$$x^n_{\max} \equiv \arg\max_{\bar{\mathbf{x}} \in \mathcal{B}_x(\mathbf{x})} S(\bar{\mathbf{x}}). \quad (4)$$

If there is a tie in (4), one maximum-scoring candidates is selected randomly. This does not effect our results. The source estimate at time n is[4]

$$\hat{x}^{n-\Delta} = \begin{cases} x^{n-\Delta}_{\max} & \text{if } S(x^n_{\max}) \geq n - \Delta, \\ \text{error} & \text{else.} \end{cases} \quad (5)$$

An example of the scoring mechanism is given in Fig. 1. In this example, $S(x^8|\tilde{x}^8) < S(\tilde{x}^8|x^8)$, $S(\tilde{x}^8|\bar{x}^8) = S(\bar{x}^8|\tilde{x}^8)$, and $S(\bar{x}^8|x^8) < S(x^8|\bar{x}^8)$. This shows that the preliminary scores by themselves do not yield an ordering. An alternate approach is to decode to the minimum entropy sequence. This gives an ordering and does not require a two-stage decoding approach. However, a term that is polynomial in n (not Δ) ends up multiplying the exponential decay (in Δ) of the error probability. This is why we introduce the suffix-decoding rule.

The central characteristics of the decoding rule that we exploit is encapsulated in the following lemma:

x^8	$\tilde{x}^8$	$\bar{x}^8$
0 0 0 0 0 1 0 1	0 0 0 0 1 1 1 1	0 1 0 0 0 1 0 1
$S(x^8\|\tilde{x}^8) = 4$	$S(\tilde{x}^8\|x^8) = 8$	$S(\bar{x}^8\|x^8) = 1$
$S(x^8\|\bar{x}^8) = 8$	$S(\tilde{x}^8\|\bar{x}^8) = 1$	$S(\bar{x}^8\|\tilde{x}^8) = 1$
$S(x^8) = 4$	$S(\tilde{x}^8) = 1$	$S(\bar{x}^8) = 1$
$\pi(x^8) = 0\,0\,0\,0$	$\pi(\tilde{x}^8) = 0$	$\pi(\bar{x}^8) = 0$

Fig. 1. Example of scoring mechanism and prefix assignment. In this example $x^8_{\max} = x^8$, so $S(x^8_{\max}) = 4$ and $\pi(x^8_{\max}) = 0\,0\,0\,0$.

[1] Note that we can always transform a source with a non-nearly-integer entropy into another with near-integer entropy by grouping the right numbers of source symbols together into super-symbols to form a new source with nearly-integer entropy.

[2] We call these "parity bits" as they can be generated using an infinite constraint-length time-varying convolutional code.

[3] Note that maximum likelihood decoders perform a similar suffix comparison implicitly since, for a memoryless source, the shared prefix has the same probability. Thus the sequence with the higher probability suffix is always more likely. On the other hand, for empirical entropy decoders, the sequence that has the lower suffix entropy does not necessarily have the lower overall empirical entropy. Say that the shared prefix is the all-zeros sequence. Consider two suffixes, the all-ones suffix, and a suffix that is half ones and half zeros, both of which are equal in length to the prefix. Although the all-ones suffix has zero empirical entropy, the sequence that results when the all-ones suffix is concatenated together with the all-zeros prefix has a higher empirical entropy than the half-half suffix and the all-zeros prefix.

[4] We could instead have used $\hat{x}^{S(x^n_{\max})} = \pi(x^n_{\max})$ or $\hat{x}^n = x^n_{\max}$, both of which would generally yield a longer estimate. We choose the current definition (5) to simplify the decoding error – either we get all symbols correct up to time $n - \Delta$, or we get an error.

Lemma 1: If $S(\bar{\mathbf{x}}) \leq S(\bar{\bar{\mathbf{x}}})$ then $\pi(\bar{\mathbf{x}}) \sqsubseteq \pi(\bar{\bar{\mathbf{x}}})$.

Proof: The proof makes use of two aspects of the scoring function. First, that $S(\bar{\mathbf{x}}) \leq S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}})$ for all $\bar{\bar{\mathbf{x}}}$ by definition (2). And second, that $\min\{S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}}), S(\bar{\bar{\mathbf{x}}}|\bar{\mathbf{x}})\}$ is the length of the longest shared prefix of $\bar{\mathbf{x}}$ and $\bar{\bar{\mathbf{x}}}$.

First consider the case when $S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}}) \leq S(\bar{\bar{\mathbf{x}}})$. Under this assumption we have

$$\pi(\bar{\mathbf{x}}) \overset{(a)}{\equiv} \bar{x}^{S(\bar{\mathbf{x}})} \overset{(b)}{\sqsubseteq} \bar{x}^{S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}})} \overset{(c)}{=} \bar{\bar{x}}^{S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}})} \overset{(d)}{\sqsubseteq} \bar{\bar{x}}^{S(\bar{\bar{\mathbf{x}}})} \equiv \pi(\bar{\bar{\mathbf{x}}}).$$

The first equality (a) is the definition of $\pi(\bar{\mathbf{x}})$. The first subsequence relationship (b) comes from $S(\bar{\mathbf{x}}) \leq S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}})$ for all $\bar{\bar{\mathbf{x}}}$ by definition (2). The equality (c) comes from the assumption that $S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}}) \leq S(\bar{\bar{\mathbf{x}}})$. This implies that $S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}})$ is the length of the longest shared prefix of $\bar{\mathbf{x}}$ and $\bar{\bar{\mathbf{x}}}$ and therefore that their first $S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}})$ symbols match. The second subsequence inequality (d) also follows from the assumption that $S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}}) \leq S(\bar{\bar{\mathbf{x}}})$.

Second, consider the case when $S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}}) > S(\bar{\bar{\mathbf{x}}})$. Note that this implies that the length of the longest shared prefix $s_{\min} = \min\{S(\bar{\mathbf{x}}|\bar{\bar{\mathbf{x}}}), S(\bar{\bar{\mathbf{x}}}|\bar{\mathbf{x}})\} \geq S(\bar{\bar{\mathbf{x}}})$. We therefore have

$$\bar{x}^{s_{\min}} = \bar{\bar{x}}^{s_{\min}} \overset{(a)}{\sqsupseteq} \bar{\bar{x}}^{S(\bar{\bar{\mathbf{x}}})} \equiv \pi(\bar{\bar{\mathbf{x}}}) \overset{(b)}{\sqsupseteq} \bar{\bar{x}}^{S(\bar{\mathbf{x}})} \overset{(c)}{=} \bar{x}^{S(\bar{\mathbf{x}})} \equiv \pi(\bar{\mathbf{x}})$$

The relation (a) follows from $s_{\min} \geq S(\bar{\bar{\mathbf{x}}})$, and (b) by the given that $S(\bar{\bar{\mathbf{x}}}) \geq S(\bar{\mathbf{x}})$. Since $s_{\min} \geq S(\bar{\bar{\mathbf{x}}})$ and we are given that $S(\bar{\bar{\mathbf{x}}}) \geq S(\bar{\mathbf{x}})$ in the statement of the lemma, therefore $s_{\min} \geq S(\bar{\mathbf{x}})$. Therefore since $\bar{x}^{s_{\min}} = \bar{\bar{x}}^{s_{\min}}$ we also have $\bar{\bar{x}}^{S(\bar{\mathbf{x}})} = \bar{x}^{S(\bar{\mathbf{x}})}$, which we apply in (c). ■

Lemma 1 proves that the prefixes $\pi(\bar{\mathbf{x}})$ for $\bar{\mathbf{x}} \in \mathcal{B}_x(\mathbf{x})$ have a well-defined order. This means that that first $S(\mathbf{x})$ symbols of the estimate are correct, where $S(\mathbf{x})$ is the score of the random source $\mathbf{x}$. Thus, decoding errors can only occur when $S(\mathbf{x}) < n - \Delta$. Of course, $S(\mathbf{x})$ is random. In the remainder of this section we bound $\Pr[S(\mathbf{x}) < n - \Delta]$ by a decaying exponent in Δ.

Theorem 1: Given a rate $R_x > H(p_x)$, then for all $E < E_r(R_x)$ there is a constant $K > 0$ such that $\Pr[\hat{x}^{n-\Delta} \neq x^{n-\Delta}] \leq K \exp\{-\Delta E\}$ for all $n, \Delta \geq 0$ where

$$E_r(R_x) = \inf_q D(q\|p_x) + |R_x - H(q)|^+, \qquad (6)$$

and where $|z|^+ = z$ if $z \geq 0$ and $|z|^+ = 0$ if $z < 0$.

Proof Strategy: To lead to a decoding error, some other sequences $\tilde{\mathbf{x}}$ must (i) satisfy the parity bits, i.e., $\tilde{\mathbf{x}} \in \mathcal{B}_x(\mathbf{x})$, and (ii) must give a score $S(\mathbf{x}|\tilde{\mathbf{x}}) = l < n - \Delta$. If $S(\mathbf{x}) \geq n - \Delta$, no such sequence exists. We bound $\Pr[S(\mathbf{x}) < n - \Delta]$ by partitioning all possibly misleading sequences $\{\tilde{\mathbf{x}} \text{ s.t } \tilde{\mathbf{x}} \in \mathcal{B}_x(\mathbf{x}), \tilde{\mathbf{x}} \neq \mathbf{x}\}$, into disjoint subsets determined by the first symbol in which they differ from $\mathbf{x}$. Thus, $\Pr[S(\mathbf{x}) < n - \Delta] = \ldots$

$$= \sum_{P_x} \sum_{\mathbf{x} \in \mathcal{T}_{P_x}} \Pr[S(\mathbf{x}) < n - \Delta | \mathbf{x}] \; p_{\mathbf{x}}(\mathbf{x}) \qquad (7)$$

$$= \sum_{P_x} \sum_{\mathbf{x} \in \mathcal{T}_{P_x}} \sum_{l=1}^{n-\Delta-1} \Pr[\exists \; \tilde{\mathbf{x}} \text{ s.t. } \tilde{x}^l = x^l, \tilde{x}_{l+1} \neq x_{l+1}, \ldots S(\mathbf{x}|\tilde{\mathbf{x}}) = l, \tilde{\mathbf{x}} \in \mathcal{B}_x(\mathbf{x}) | \mathbf{x}] \; p_{\mathbf{x}}(\mathbf{x}) \qquad (8)$$

$$\leq \sum_{l=1}^{n-\Delta-1} \sum_{P_x} \sum_{\mathbf{x} \in \mathcal{T}_{P_x}} \min[1, \sum_{\substack{\tilde{\mathbf{x}} \text{ s.t. } \tilde{x}^l = x^l, \\ \tilde{x}_{l+1} \neq x_{l+1}, \\ S(\mathbf{x}|\tilde{\mathbf{x}}) = l}} \Pr[\tilde{\mathbf{x}} \in \mathcal{B}_x(\mathbf{x})]] \; p_{\mathbf{x}}(\mathbf{x})$$

$$= \sum_{l=1}^{n-\Delta-1} \sum_{P^l, P^{n-l}} \sum_{\substack{x^l \in \mathcal{T}_{P^l}, \\ x_{l+1}^n \in \mathcal{T}_{P^{n-l}}}} \min[1, \sum_{\substack{\tilde{\mathbf{x}} \text{ s.t. } \tilde{x}^l = x^l, \\ \tilde{x}_{l+1} \neq x_{l+1}, \\ H(\tilde{x}_{l+1}^n) \leq H(x_{l+1}^n)}} \ldots \exp\{-(n-l)R_x\}] \; p_{\mathbf{x}}(\mathbf{x}) \qquad (9)$$

$$= \sum_{l=1}^{n-\Delta-1} \sum_{P^l, P^{n-l}} \sum_{\substack{x^l \in \mathcal{T}_{P^l}, \\ x_{l+1}^n \in \mathcal{T}_{P^{n-l}}}} \min[1, \sum_{\substack{\tilde{P}^{n-l} \text{ s.t.} \\ H(\tilde{P}^{n-l}) \leq H(P^{n-l})}} \ldots \sum_{\tilde{x}_{l+1}^n \in \mathcal{T}_{\tilde{P}^{n-l}}} \exp\{-(n-l)R_x\}] \; p_{\mathbf{x}}(\mathbf{x}) \qquad (10)$$

$$= \sum_{l=1}^{n-\Delta-1} \sum_{P^l, P^{n-l}} \sum_{\substack{x^l \in \mathcal{T}_{P^l}, \\ x_{l+1}^n \in \mathcal{T}_{P^{n-l}}}} \min[1, (n-l+1)^{|\mathcal{X}|} \ldots \exp\{-(n-l)[R_x - H(P^{n-l})]\}] \; p_{\mathbf{x}}(\mathbf{x}) \qquad (11)$$

$$\leq \sum_{l=1}^{n-\Delta-1} (n-l+1)^{|\mathcal{X}|} \sum_{P^{n-l}} \sum_{x_{l+1}^n \in \mathcal{T}_{P^{n-l}}} \exp\{-(n-l)[\ldots |R_x - H(P^{n-l})|^+ + D(P^{n-l}\|p_x) + H(P^{n-l})]\} \qquad (12)$$

$$\leq \sum_{l=1}^{n-\Delta-1} (n-l+1)^{|\mathcal{X}|} \sum_{P^{n-l}} \exp\{-(n-l)\ldots \inf_q [D(q\|p_x) + |R_x - H(q)|^+]\} \qquad (13)$$

$$\leq \sum_{l=1}^{n-\Delta-1} (n-l+1)^{2|\mathcal{X}|} \exp\{-(n-l)E_r(R_x)\} \qquad (14)$$

$$\leq \sum_{l=1}^{n_0-1} K_1 \exp\{-(n_0 + \Delta - l)[E_r(R_x) - \gamma]\} \qquad (15)$$

$$\leq K_2 \exp\{-\Delta[E_r(R_x) - \gamma]\}. \qquad (16)$$

After conditioning on the realized source sequence in (7), the remaining randomness is only in the binning. In (8) we decompose the error event into a number of mutually exclusive events, and in the following line apply the union bound. In (9) we reenumerate the possible source

sequences in terms of the shared prefix $x^l = \tilde{x}^l$ and the differing suffixes $x_{l+1}^n \neq \tilde{x}_{l+1}^n$. We define P^l and P^{n-l} as the types, and $\mathcal{T}_{P^l}$ and $\mathcal{T}_{P^{n-l}}$ as their type classes, where $x^l \in \mathcal{T}_{P^l}$ and $x_{l+1}^n \in \mathcal{T}_{P^{n-l}}$. We also rewrite the constraint $S(\mathbf{x}|\tilde{\mathbf{x}}) = l$ explicitly as $H(\tilde{x}_{l+1}^n) \leq H(x_{l+1}^n)$. In (10) by the scoring rule (1), the only suffixes $\tilde{x}_{l+1}^n$ that can cause $S(\mathbf{x})$ to be below $n-\Delta$ are those with lower suffix entropy than $H(x_{l+1}^n) = H(P^{n-l})$. We sum over this set. In going from (10) to (11) we first note that the argument of the inner-most summation (over $\tilde{x}_{l+1}^n$) does not depend on $\mathbf{x}$. We then use the following relations: (i) $\sum_{\tilde{x}_{l+1}^n \in \mathcal{T}_{\tilde{P}^{n-l}}} = |\mathcal{T}_{\tilde{P}^{n-l}}| \leq \exp\{(n-l)H(\tilde{P}^{n-l})\}$, which is a standard bound on the size of the type class, (ii) $H(\tilde{P}^{n-l}) \leq H(P^{n-l})$ by the minimum-suffix-entropy decoding rule, and (iii) the polynomial bound on the number of types, $|\{\tilde{P}^{n-l}\}| \leq (n-l+1)^{|\mathcal{X}|}$. In (12) we use the memoryless property of the source to sum out over $p_{x^l}(x^l)$, and pull the polynomial term out of the minimization. We also use $p_{x_{l+1}^n}(x_{l+1}^n) = \exp\{-(n-l)[D(P^{n-l}\|p_x) + H(P^{n-l})]\}$ for all $\mathbf{x} \in \mathcal{T}_{P^{n-l}}$ and combine the exponents. As the expression no longer depends on x_{l+1}^n, in (13) we simplify by using $|\mathcal{T}_{P^{n-l}}| \leq \exp\{(n-l)H(P^{n-l})\}$. In (14) we define the error exponent $E_r(R_x) \triangleq \inf_q[D(q\|p_x) + |R_x - H(q)|^+]$, and use the polynomial bound on the number of types. In (15) we incorporate the polynomial into the exponent. Namely, for all $a > 0$, $b > 0$, there exists a C such that $z^a \leq C\exp\{b(z-1)\}$ for all $z \geq 1$. We then use $n = n_0 + \Delta$ to make explicit the delay-dependent term. Pulling out the exponent in Δ, the remaining summation is a sum over decaying exponentials, and can be bounded by a constant. Together with K_1, this gives the constant K_2 in (16). ■

This proves Theorem 1. Note that the γ in (16) does not enter the optimization because $\gamma > 0$ can be picked equal to any constant. The choice of γ effects the constant K in Theorem 1.

III. Streaming Slepian-Wolf Coding

In this section we present random coding error exponents for streaming Slepian-Wolf systems. The proof techniques used are extensions of those used for streaming entropy coding in Section II. We give results both for universal and maximum-likelihood decoders.

In a Slepian-Wolf system the source $(\mathbf{x}, \mathbf{y})$ is jointly distributed in a memoryless manner where $p_{x^n,y^n}(x^n, y^n) = \prod_{i=1}^n p_{x,y}(x_i, y_i)$ for all n. One encoder observes the x^n stream, and causally encodes it using the sequential random binning strategy presented in Section II at rate R_x. The second encoder observes y^n, and uses the same strategy at rate R_y. We want to design a system such that $\Pr[(x^{n-\Delta}, y^{n-\Delta}) \neq (\hat{x}^{n-\Delta}, \hat{y}^{n-\Delta})]$ decays exponentially in Δ.

For universal decoding we use a weighted joint variant of the preliminary scoring rule (1). In particular, say that $\bar{x}^l = \bar{\bar{x}}^l, \bar{x}_{l+1} \neq \bar{\bar{x}}_{l+1}$, and $\bar{y}^k = \bar{\bar{y}}^k, \bar{y}_{k+1} \neq \bar{\bar{y}}_{k+1}$. Then, if $l \leq k$, the preliminary score is $S(\bar{\mathbf{x}}, \bar{\mathbf{y}}|\bar{\bar{\mathbf{x}}}, \bar{\bar{\mathbf{y}}}) = \ldots$

$$\begin{cases} l & \text{if} \quad (n-l)H(\bar{x}_{l+1}^n|\bar{y}_{l+1}^n) + (n-k)H(\bar{y}_{k+1}^n) \geq \\ & \qquad (n-l)H(\bar{\bar{x}}_{l+1}^n|\bar{\bar{y}}_{l+1}^n) + (n-k)H(\bar{\bar{y}}_{k+1}^n) \\ n & \text{else.} \end{cases}$$

If $k < l$, swap x with y and l with k. The rest of the decoding rule is defined as before. This gives,

Theorem 2: Given a rate pair (R_x, R_y) such that $R_x > H(p_{x|y})$, $R_y > H(p_{y|x})$, $R_x + R_y > H(p_{xy})$, then for all $E < E_r(R_x, R_y)$ there is a constant $K > 0$ such that $\Pr[(\hat{x}^{n-\Delta}, \hat{y}^{n-\Delta}) \neq (x^{n-\Delta}, y^{n-\Delta})] \leq K\exp\{-\Delta E\}$ for all $n, \Delta \geq 0$ where $E_r(R_x, R_y) =$

$$\min\Bigg[\inf_{q_y, \bar{q}_y, q_{x|y}, \lambda}\Big\{\lambda D(q_{x|y}q_y\|p_{xy}) + \bar{\lambda}D(q_{x|y}\bar{q}_y\|p_{xy})\ldots + \big|\lambda[R_x - H(q_{x|y}|q_y)] + \bar{\lambda}[R_x + R_y - H(q_{x|y}\bar{q}_y)]\big|^+\Big\},$$
$$\inf_{q_x, \bar{q}_x, q_{y|x}, \lambda}\Big\{\lambda D(q_{y|x}q_x\|p_{xy}) + \bar{\lambda}D(q_{y|x}\bar{q}_x\|p_{xy})\ldots + \big|\lambda[R_y - H(q_{y|x}|q_x)] + \bar{\lambda}[R_x + R_y - H(q_{y|x}\bar{q}_x)]\big|^+\Big\}\Bigg]$$

where $0 \leq \lambda \leq 1$, $\bar{\lambda} = (1-\lambda)$, and where the probability is taken over both random encoders and the random source. This distributions $q_{x|y}$ and $q_{y|x}$ are conditional, q_y, $\bar{q}_y$, q_x, and $\bar{q}_x$ are marginal distributions, and, e.g., $q_{x|y}q_y$ is a joint distribution with a conditional entropy expressed as $H(q_{x|y}|q_y)$.

In the maximum-likelihood context, the decoder selects the most likely pair of sequences, giving

Theorem 3: Given a rate pair (R_x, R_y), such that $R_x > H(p_{x|y})$, $R_y > H(p_{y|x})$, $R_x + R_y > H(p_{xy})$, then for all $E < E_r(R_x, R_y)$ there is a constant $K > 0$, such that $\Pr[(\hat{x}^{n-\Delta}, \hat{y}^{n-\Delta}) \neq (x^{n-\Delta}, y^{n-\Delta})] \leq K\exp\{-\Delta E\}$ for all $n, \Delta \geq 0$ where $E_r(R_x, R_y) =$

$$\min\left\{\inf_{\lambda\in[0,1]}\sup_{\rho\in(0,1)} G_{x_\lambda}(\rho), \inf_{\lambda\in[0,1]}\sup_{\rho\in(0,1)} G_{y_\lambda}(\rho)\right\} > 0,$$

and where the probability is taken over both random encoders and the random source. The functions $G_{x_\lambda}(\rho) = \lambda G_x(\rho) + (1-\lambda)G_0(\rho)$, and $G_{y_\lambda}(\rho) = \lambda G_y(\rho) + (1-\lambda)G_0(\rho)$, where

$$G_x(\rho) = \rho R_x - \log_e\left[\sum_{y\in\mathcal{Y}}\left(\sum_{x\in\mathcal{X}} p_{x,y}(x,y)^{\frac{1}{1+\rho}}\right)^{1+\rho}\right],$$

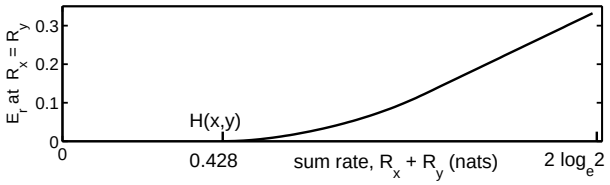

Fig. 2. Error exponent $E_r(R_x, R_y)$ for a binary-asymmetric example with $H(x,y) = 0.428$, evaluated along the symmetric-rate line $R_x = R_y$. The source statistics are $p_{x,y}(0,0) = 0.05$, $p_{x,y}(0,1) = 0.03$, $p_{x,y}(1,0) = 0.02$, and $p_{x,y}(1,1) = 0.9$.

$$G_y(\rho) = \rho R_y - \log_e \left[\sum_{x \in \mathcal{X}} \left(\sum_{y \in \mathcal{Y}} p_{x,y}(x,y)^{\frac{1}{1+\rho}} \right)^{1+\rho} \right],$$

$$G_0(\rho) = \rho(R_x + R_y) - (1+\rho) \log_e \left[\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{x,y}(x,y)^{\frac{1}{1+\rho}} \right].$$

The exponents of Theorems 2 and 3 are generally smaller than their block-coding counterparts [1], [3]. However, the difference disappears when one of the sources is observed at the decoder. We also believe that the exponents of Theorems 2 and 3 can be shown to be equal using the same techniques that work for other comparisons of universal and ML error exponents.

In Figure 2 we plot the ML error exponent for a binary-asymmetric example along the symmetric rate line $R_x = R_y$. The maximum difference between this exponent and Gallager's [3] occurs at $R_x + R_y = 0.6222$, where the difference is 0.0011. The maximum difference at any rate pair for this example is 0.003. We do not claim that our error exponents are optimal, but it is important to realize that a consequence of our theorems is that *every source symbol* is eventually decoded correctly with probability 1. Atypical bursts of source symbols result in increased delay till correct decoding, but as long as the rate pair lies within the achievable region, everything is decoded correctly eventually. This is impossible to achieve with block-codes.

IV. Discussion and Example

In this paper we have derived error exponents for sequential binning schemes as applied to entropy and distributed source coding. In the general Slepian-Wolf setting of two remote encoders, the exponents of Theorems 2 and 3 are generally smaller than their block-coding counterparts, although, as in the example of Figure 2, the difference is usually quite small.

We have not yet shown upper bounds on the error exponents of streaming systems. The next example illustrates that such error exponents are not necessarily upper-bounded by their block-coding counterparts. Consider a system observing a stream of tertiary i. i. d. symbols with distribution $p_x(a) = 0.9$, $p_x(b) = 0.05$ and $p_x(c) = 0.05$. At each time, the encoder observes a pair of these symbols, and can send 3 bits to the decoder. The best possible block-coding error exponent for this system can be shown by using [1], Theorem 2.15, to equal 1.474 bits per symbol pair. Now consider the following ad-hoc streaming strategy, which we show results in a much higher exponent.

The encoder takes each pair of source symbols and maps them into either two or four bits as follows. If the ith source-pair $\mathbf{x}_i = (a,a)$, the encoder outputs 00. For any other pair the encoder maps it into a four-bit prefix code, i.e., the bit-tuplets $1000, 1001, \ldots, 1111$. The encoder output is fed into an infinite-length first-in-first-out buffer, the oldest three bits of which are sent to the decoder at each time step, padded by zeros if the buffer is empty. Note that to stay synchronized the decoder can count symbols to tell when the buffer is empty. Denote the number of bits in the buffer at time i by b_i. If $\mathbf{x}_i = (a,a)$, then $b_{i+1} = b_i - 1$, else $b_{i+1} = b_i + 1$. The stationary distribution of b_i exists, and is $\lim_{i \to \infty} \Pr[b_i = l] = \mu_l = 0.7654(0.19/0.81)^l$.

If $b_{n_0} \leq R_x \Delta - 4$, then by time $n_0 + \Delta$, the encoding of symbol pair $\mathbf{x}_{n_0}$ will certainly have been received by the decoder. Thus, in steady-state, $\Pr[\hat{\mathbf{x}}^{n-\Delta} \neq \mathbf{x}^{n-\Delta}] \leq \Pr[b_{n-\Delta} \geq R_x \Delta - 4] = \sum_{l=R_x\Delta-4}^{\infty} \mu_l \leq 2^9 2^{-2.09 R_x \Delta}$. For $R_x = 3$ bits per symbol pair, this gives an error exponent $E_r(3) = 6.27$, far larger than the block-coding exponent of 1.474. The streaming exponent can be further increased by mapping $\mathbf{x}_i = (a,a)$ to 0 instead of to 00. A related study of buffer overflow and variable length coding was made by Jelinek in [4].

Acknowledgment

The authors wish to acknowledge a desire expressed by Zixiang Xiong and subsequent hallway discussions during ITW 2004 that helped precipitate the current line of research. This work was supported in part by the NSF Grant No. CNS-0326503.

References

[1] I. Csiszár and J. Körner. *Information Theory, Coding Theorems for Discrete Memoryless Systems.* Akadémiai Kiadó, 1981.

[2] S. C. Draper, C. Chang, and A. Sahai. Sequential random binning for streaming distributed source coding. *In preparation.*

[3] R. G. Gallager. Source coding with side information and universal coding. Technical Report LIDS-P-937, Mass. Instit. Tech., 1976.

[4] F. Jelinek. Buffer overflow in variable length coding of fixed rate sources. *IEEE Trans. Inform. Theory*, 14:490–501, May 1968.

[5] A. Sahai. *Anytime Information Theory*. PhD thesis, Mass. Instit. of Tech., 2001.

[6] D. Slepian and J. K. Wolf. Noiseless coding of correlated information sources. *IEEE Trans. Inform. Theory*, 19:471–480, July 1973.