Probabilistic and Q-Coder Algorithms for Binary Source Adaptation

Glen G. Langdon, Jr.
Computer Engineering Department
University of California, Santa Cruz 95064

IBM Almaden Research Center
San Jose CA 95120-6099

1. Deterministic and Probabilistic counting for binary adapters

Familiarity with arithmetic coding is assumed. Early probability estimators for binary sources employ deterministic techniques, see [1], for a survey. One way to characterize a binary source is by the value of the more probable symbol (MPS) and less probable symbol (LPS) and by skewness of the distribution. The negative log2 of p(LPS) serves as a skewness measure. Many binary adapters quantize the skewness range to a small number of indices k, and use an index-change decision to alter the estimate.

Deterministic *Skewcount:* In one family of algorithms [2-4], a count ratio is maintained where the numerator is the LPS count, and both numerator and denominator are halved whenever the LPS count exceeds the maximum allowable count. In Skewcount, the maximum (and minimum) LPS count is 1: both counts are halved whenever an LPS occurs. In Skewcount, the denominator count is increased for each occurrence of the MPS. Since the index k is decreased to a less skewed estimate each LPS, the new MPS count required to reach the next highly skewed index k was experimentally adjusted so the estimate wandered about the index k corresponding to the probability distribution of the source. The index k is determined by the most significant bit of the denominator.

Probabilistic *Skewadap:* In the probabilistic count based binary adapter called Skewadap [5-6], the numerator is handled as in Skewcount (fixed at 1), however the larger binary count (denominator) is maintained with fewer bits and fewer updates by using the probabilistic counting technique: a Monte Carlo test outcome determines whether the count increases or remains the same. The need to counter-balance the upward (MPS path) and downward (LPS path) skewness movement is inherent in all wandering algorithms. In Skewadap, the design experimentally balanced the lower skewness movement of decrementing the index with the higher skewness movement to a lower value of $p_{MC}(k)$.

Morris [7] conceived the use of *probabilistic counting* by maintaining large counts in small registers. He assigns count value N to an event that occurs 1/N of the time, and lets the register value represent an exponent to the number base. In Skewadap, count radix 2 is used. When an LPS occurs, the denominator count is halved. The Skewadap denominator is adjusted by probabilistic counting: an MPS occurrence results in a Monte Carlo test whose answer is YES of the appropriate probability. In Skewadap the count radix is 2 and the exponent is an integer value k. A denominator, represented by an index k is 2^k,

TH0373-1/91/0000/0013/$01.00 © 1991 IEEE

remains the same if the test is NO, and the index is increased to k + 1 (representing a denominator of 2^{k+1}) if the Monte Carlo test result is YES. A key to the balancing of this wandering algorithm centering about an index appears in [6] and is stated in the form of a heuristic: *The higher the skew number the lower the probability of a yes answer given the occurrence of a most probable symbol.*

Monte Carlo techniques typically employ a pseudorandom number as the value tested; in Skewadap a simple up-counter was used, and the least significant k bits were tested for a unique combination to give a YES one every $1/2^k$ times. The Monte Carlo index-change decision was conceived in [5] as a generic method to eliminate the overhead of precise counting for each context in adaptive probability estimation algorithms for binary arithmetic coders, thus sharing the pseudorandom number generator. For Skewadap, when there is only one context, the pseudorandom counter normally shared by all contexts becomes an actual counter for the lone context.

In the multiple context case, context z determines index k, which accesses Monte Carlo test probability $p_{MC}(k)$. The random number generator provides the next such number R between 0 and 1, which is compared against p_{MC}, and if less, delivers the outcome Yes. The Yes outcome chooses the new index value (as opposed to keeping the old value) and the new index is rewritten to coding parameter memory at the location for context z.

Phillipe Flajolet was provided with a copy of the report describing Skewadap [6], and further investigated probabilistic counting. Flajolet performed a successful mathematical analysis of the radix 2 probabilistic counting algorithm incorporated into Skewadap. He showed that after n applications, the exponent random variable k(n) yields an expected count for $2^{k(n)}$ of n + 2, with a relatively low dispersion if n is large. For greater count accuracy, Flajolet observes that using a smaller count radix has even less dispersion [8].

2. Using the MPS renormalization for adaptation

The MPS and LPS renormalization (renorm) operations control the shifting of the code string during arithmetic coding, and refer to renormalizing the A register by left-shifting both the A register and the code string as many bit positions as are needed for the A register value to lie in the range [1.0,2.0). After a communication on Skewadap, colleagues G. Goertzel and J. L. Mitchell suggested that the random number generator and Monte Carlo test be replaced, as this operation is expensive for software versions in terms of instruction executions in the MPS instruction path of the adapter. Instead, they suggested that the MPS renorm serve as the Yes answer that causes adaptation to occur in the MPS path.

The binary arithmetic coder to which the idea applied is called the *Skew coder*. During the Skew coder encoding operation, a value, called the *augend* Q, is added to the code string and subtracted from the A register. An MPS renorm occurs if: $(A\text{-}Q) < 1$, where $1 \leq A < 2$.

In treating the MPS renorm as the Yes answer to a Monte Carlo test, fortuitously, augend value Q is such that the heuristic mentioned in [6] applies: The smaller the value of Q, the less the chance of a YES answer and the more highly skewed is the current probability estimate. Making the value of Q larger makes the estimate less highly skewed while increasing the chance of an MPS renorm.

Unfortunately, neither the author's attempts nor those of Goertzel and Mitchell, reduced the idea to practice. If successful, the adapter would be entirely renormalization-driven, since the LPS path always does a renormalization. Mitchell also observed that the index need not be changed on every MPS renorm, but could (for example) be incremented on every other renorm.

3. Multi-rate Adaptive Q-Coder

A landmark development for entirely renorm-driven adapter-coder algorithms is an unpublished multi-rate Q-Coder due to Pennebaker and Mitchell [9]. This algorithm reduced to practice the Q-Coder concept, as two parallel algorithms employed only during renormalizations. The result was preceded by a theoretical analysis and experimental verification (some of which appears in [10]).

This parallel two-adapter scheme has a first-order single-rate adapter to handle the stationary strings and a second-order adapter to detect nonstationary changes and add extra jump amounts to the first-order index change. The first-order adapter employs a constant or uniform index-change rate: as in Skewadap, the index jumps and jump decisions are the *same* for each index k (*independent* of the current estimate) except at the extremes. When the normal counter-balancing of the first-order adapter fails to track a change in statistics, then an abundance of the same type of renorm occur. This out-of-balance condition is detected by the second-order adapter. The idea is described in §6 of [10], where it is attributed to an unpublished Log Coder adaptation technique. The extra jump amount of the second-order adapter depends both on the current skewness estimate k and on the value of a context-dependent counter that measures the out-of-balance condition. The larger the nonstationary change in the statistic, the higher the counter value, and the larger the extra jump. (See [10 (Table 3)] for the additional jump table).

The multi-rate algorithm, as noted in [10], uses two index-change strategies together, and requires a 12-bit to 13-bit coding parameter. The notion of multi-rate adaptation was introduced by a statement in [9] summarizing the study of single-rate adaptation. "Although the single rate adaptation experiments and modeling described in the preceding section are of interest in understanding the mechanisms of probability estimation via renormalizations, the single rate adaptation is too restricted to give good performance over a variety of real mixed state datasets."

4. A single-adapter solution: L-SQ1

Although [10] does not relate Q-Coder adaptation to Skewadap, the author's design approach in L-SQ1 views the Monte Carlo test of Skewadap as evolving through the Goertzel-Mitchell suggestion that also uses an ever smaller probability to index-change to ever higher skews, and finally to test "(A-Q) < 1" of the Q-Coder. Of course, the situation is more complicated because, as in Skewadap, a single-context makes the situation deterministic. This presentation concentrates on the assumption the A register value appears random.

The algorithm credited to Langdon in footnote 2 of [10] is called L-SQ1. This algorithm uses a 6-bit coding parameter. L-SQ1 reduced to practice the concept of a single-adapter Q-Coder [11]. Success was due to [9] and the multi-rate Q-Coder, the ability to select the number and value of the augends Q(k), and an index-change strategy based on Skewadap. The now familiar (Skew count, Skewadap) experimental task was to counter-balance the downward skewness movement by the skewness change on the MPS path for best compression. Since only the MPS renorms and LPS outcome drive the adapter, the former to increase skewness and the latter to decrease it, these movements should counter-balance themselves in the vicinity of the best index k and augend Q(k) for the input distribution q_{bpd}. In the algorithm of [9], a second order algorithm is needed when compressing real data in a multiple-context environment.

An important new challenge for single-adapter Q-Coders was to accommodate the relative frequency of the Yes answer, denoted p(Mrn|k), instead of simply prescribing the value p(Yes|MPS) as in [5]. The counter-balancing problem in the MPS path was solved in L-SQ1 by the fortuitously simple idea of slowing the upward skewness movement of the MPS path by increasing the skewness only on *every other* MPS renorm, an idea suggested by Mitchell [12] during the MPS renorm investigation described in Section 2. The consequence of waiting for two MPS renorms is that the probability of an index change is $p(Mrn|k)^2$, the *square* of probability value p(Mrn|k).

A single-adapter Q-Coder is most conveniently viewed as a single algorithm, but adjusts the index-change strategy according to the index value. In L-SQ1, at lower skew indices the rate is slower (increase the index on the second, fourth, etc., MPS renorm) and at higher skews is higher (increase the index on the first, third, etc., MPS renorm). In [9] only index-change strategies that used a single rate for all indices were considered.

The resulting L-SQ1 outperforms the earlier pair of Skew coder and Skewadap. Moreover, L-SQ1 was within a few percent of the performance of the Pennebaker and Mitchell multi-rate Q-Coder [9], while possessing an estimated hardware implementation that performed at twice the speed with half the silicon area [13].

5. Evolution of the Q-Coder from L-SQ1

Following L-SQ1, Pennebaker and Mitchell together with the author, improved the single-adapter approach to the point where it competed favorably with the multi-rate approach. Table 1 above traces the checkpoints along the way. The Q-Coder adaptation algorithms corresponding to column headings with SJ are due to the author working in San Jose, CA, and the column headings YKT are due to Pennebaker and Michell working in Yorktown Heights, NY. The notation "Com" is a splicing together of the SJ higher skew strategy and

YKT lower skew strategy into a common algorithm. Except for minor changes not related to index-change strategy, the published algorithm [10] is ComboX.

Table 1. Evolution and Performance of Single-Adapter Q-Coders

	L-SQ1	YKT	SJ	SJ Com	YKTmodC	SJmodC	ComboX
	RN@1	RN@3/4	RN@1	RN@3/4	RN@3/4	RN@1	RN@1
Test	3/21/85	4/23/85	5/5/85	5/5/85	5/22/85	5/25/85	6/7/85
File	16 Qval	32 Qval	16 Qval	24 Qval	32 Qval	32 Qval	30 Qval
Bilevel							
CCITT1	121219	121472	119927	120584	120451	120020	119802
CCITT2	72502	72016	72168	71914	71796	71471	71409
CCITT3	189908	190616	189215	189172	188865	188617	188000
CCITT4	451783	453504	447829	449360	448077	447296	446584
Budking	971753	990896	966988	969394	971151	968230	966040
Gray-level							
Courierf	97209	94816	94853	94553	94606	94405	94366
Marcosf	111840	110632	110452	110425	110456	110412	110412

Subsequent to ComboX, the multi-rate approach in [10] replaced the single-rate first-order algorithm in [9] with a single-adapter algorithm, and made other adjustments. Except in special cases, the multi-rate approach is not a significant improvement over the single-adapter Q-Coder.

6. Prior analytical work

In [9] and [10], a mathematical analysis of single-rate Q-Coders for the single-context case is performed, as well as an analysis of the assumption that the A-register value is uniformly distributed within its allowable range of values. A notion of "balance" is introduced that suggests the adapter should place augend Q close to value q_{bpd}, when the renorm point is 0.75. Our new analysis shows augend Q and input q_{bpd} are not quite related by a simple scale factor.

In [10], the system is in balance at the proper index when, in state k corresponding to the input q_{bpd} or augend $Q = 1.5q_{\text{bpd}}$, the probability of an index change to the higher skew is the same as the probability the next index change will be to a lower skew. Also employed in the analysis [9,10] is the probabilistic automaton notion of state transition probabilities, where each adaptation index represents a state. A balancing condition in [9] states that the sum of transition probabilities from state k equals the sum of probabilities to state k (where probabilities are per symbol encoded).

Researchers may experience some confusion with the 0.75 scale factor in [10], whereas the paper [14] in the same issue provides the correct nonlinear relationship between most efficient augend value Q for a given distribution q_{bpd}. This work presents a theory that relates MPS renorm (Mrn) frequencies to LPS frequencies at the most efficient coding relationship, using the more precisely derived relationship from [14].

An important contribution in [9,10] is the distinction between the single-context (memoryless source) case, and the multiple-context case. The single-context case noted a range of augend values Q provided the same coding performance, which the author related to the Golomb code in [15].

For the multiple-context case, a *random augend* model is defined in [9], and is called *random interval model* in [10]. The random augend model assumes the

A register value to be uniformly distributed, because (in simplistic terms) as the context changes so does the value being subtracted from the A register.

7. New analysis of Q-Coder renorm relationships

A study of the Q-Coder or A-based coding operations and the Golomb code appears in [15]. From this work a best relationship between the augend value and input distribution q_{bpd} is defined at points between ranges of augend values that provide the best performance. The relationship is shown in **Figure 1** as discrete points. The Y-axis represents the ratio Q/q_{bpd}, and the Y axis represents q_{bpd} as the skewness measured as the negative log2.

For the random augend case, [14], Table 2, provides the ratio Q/q_{bpd} for the most efficient coding. Augend Q and q_{bpd} are ***matched*** when augend value Q yields the most efficient coding of input distribution q_{bpd}. The derivation of this relationship employs the random interval model, and the result appears as the smooth curve in Figure 1. Thus, Figure 1 proves that Q and q_{bpd} are not governed by a simple scale-factor relationship as in [10].

The correct ***matching*** Q/q_{bpd} relationship is employed to determine the expected ratio of the probability of an MPS renorm to the probability of an LPS. Let Mrn2L(Q,q) denote the expected ratio of MPS renorms to LPS events when the augend value is Q and the probability distribution parameter is q_{bpd}. The notion of ***equilibrium point*** for value Mrn2L as a guide to the index-change strategy is introduced. Let equilibrium point Mrn2L(Q) denote this ratio for augend value Q when the input distribution q_{bpd} is matched to Q. Similarly, and let equilibrium point Mrn2L(q_{bpd}) denote the ratio when augend Q is matched to input distribution q_{bpd}.

If the MPS renorms occur more frequently than its equilibrium value for a given augend Q, then the LPS symbol is less frequent than expected, and augend Q is too large. However, the extra MPS renorms drive the system to a higher skewness index, making the augend smaller. On the other hand, if the MPS renorms occur less frequently than Mrn2L, then the augend is probably too small. However, the extra LPS outcomes move the system to lower skewness indices, making the augend larger. When Q(k) is matched to the input distribution, then Mrn2L(Q) defines the normal expectation.

The ratio Mrn2L was employed in a study [9] and was one of several conditional renormalization ratios studied as a heuristic for changing the index that represents the current probability estimate of the binary input distribution. In [9], Mrn2L was denoted Pm/Q, and signified that each MPS renorm causes an index change, and each LPS causes an index-change. The index-change heuristic was used in conjunction with an amount-of-change heuristic; for example if the index jumps twice as much in the LPS direction as the MPS direction, then in [9] the ratio Pm/Q needed to be 2 for balancing.

To develop the curves that describe the equilibrium Mrn2L relationship in the random augend case, define p(Mrn|k) as the probability of an MPS renorm given an MPS event and current index k. Thus, p(Mrn|k) is the probability that the difference A-Q(k) is less than 1, see Eq 1a below. Note p(Mrn|k) is defined independently of the relative frequency of LPS events and of the type of previous renorm. Let A be the sum of fixed value 1 and a random number R uniformly distributed between 0 and 1, which is the ***random interval model*** of [10]. The probability of an MPS renorm is the probability of R-Q(k) < 0. The Yes/No truth function represented by inequality R-Q(k) < 0 is the familiar Monte Carlo test for simulating binary events whose outcome is Yes with probability Q(k), see Eq 1b. For the random interval model, the A register value is uniformly distributed, and the ***value Q(k) is the probability of an MPS renorm***, given an MPS event; see Eq 1c. This derivation shows that augend value Q(k) corresponds directly to the Monte Carlo test probability p_{MC} when the A register is uniformly distributed. Eqs 1a and 1b are valid for any distribution of A and R respectively.

p(Mrn|k) = p((A-Q(k)) < 1) (1a)
p(Mrn|k) = p((R-Q(k)) < 0), 0 < R < 1 (1b)
For R uniformly distributed: p(Mrn|k) = Q(k) (1c)

Next, note that at q_{bpd} there is one LPS outcome per q^{-1} events on average, so the expected number of MPS outcomes per LPS is p/q, or $(q^{-1}$-1). If A is uniformly distributed, then from Eq 1c, value p(MPS|k) is Q(k). What observable random variable does the MPS renorm correspond to? The random interval model makes a pseudorandom number generator out of the A register, and as learned from L-SQ1 the MPS renorm becomes the outcome of a Monte Carlo test, which per Eq 1 has relative frequency p(Mrn|k). The MPS renorm, as a probabilistic outcome associated with the total number of times augend Q(k) is tested against the A register value, signifies that, assuming each MPS renorm or LPS changes the index (as in the Q-Coder), on average 1/p(Mrn|k) MPS consecutive events have taken place since arriving at current index k. Note that value p(Mrn|k) has been defined *independent* of the input q_{bpd}. Note also that the average number of symbols before the LPS symbol occurs depends *only* on q^{-1}, and is independent of augend value Q(k). Also note that under the random interval model, value p(Mrn|k) is Q^{-1}.

Theorem. In the random augend case, with an A-based code of augend Q(k) and input sequence parameter q, the average ratio of MPS renorms per LPS event ($Mrn2L_R$(Q(k),q)) is p(Mrn|k)×(q^{-1}-1). If the A register is uniformly distributed then $Mrn2L_R$(Q(k),q) is Q(k)×(q^{-1}-1).

Proof. Since q^{-1} is the average number of input symbols per LPS event, then q^{-1}-1 is the average number of MPS events per LPS event. Since value 1/p(Mrn|k) is the average number of MPS events per MPS renorm, we have:

Mean $Mrn2L_R$(Q(k),q) = (q^{-1}-1)MPS events/LPS)/($p(Mrn|k)^{-1}$ MPS events/MPSrn) = p(Mrn|k)×(q^{-1}-1) (2)

For the A register uniformly distributed, Eq 1c substitutes Q(k) for p(Mrn|k), and the equilibrium value of $Mrn2L_R$ occurs when augend value Q is Q(q_{opt}), or $Mrn2L_R$ is (Q/q_{opt})-Q. As Q becomes small, value $Mrn2L_R$(Q(k),q) approaches the value of the dominant term Q(k)/q.

Note that the equilibrium point Mrn2L and the adaptation inertia are independent concepts. The equilibrium value is derived analytically from reasonable assumptions and first principles. In contrast the adaptation inertia or speed is a design choice determined by algorithm design decisions.

Figure 2 shows value $Mrn2L_R(q_{bpd})$ as the solid curve, where q_{bpd} is represented on the X-axis by its skewness. The saw tooth curve is not derived here (see [15]) and corresponds to Mrn2L for the single-context case (called fixed augend case in [15]. The points on the solid curve are calculated as follows. A value for augend Q is selected, and the X-axis point is $-\log_2 q_{opt}(Q)$, where $q_{opt}(Q)$ is calculated as in [19(Table 2)]. See also the description in [4]. Given q_{opt} and Q, $Mrn2L_R$ is calculated according to Eq 2.

A balance point for renorm-based adaptation is defined in [10 (Eq 4)]. A substitution and transposition on this equation reveals that at this balance point, the probability of an MPS renorm occurring before the LPS renorm is ½. Plotting this consequence of [10(eq 4)] on Figure 2 yields a horizontal line intersecting the Y-axis at point 1.0. Thus by employing the coding efficiency relationship in [15] for the fixed augend case, and an efficiency relationship derived in [14] for the random augend case, a more accurate analysis of the relationship between MPS renorms and LPS renorms is obtained. The L-SQ1 design decision that slows down the counter-balancing effect on the MPS renorm above the midpoint was made experimentally. L-SQ1 treated the higher skewness estimates as if MPS renorms were more frequent, which Figure 2 now confirms. Thus a single-adapter Q-Coder index-change strategy should not confine itself to a single rate.

8. Conclusions

We report developments related to the Q-Coder approach to adaptive binary arithmetic coding. Theorem 1 and Figure 2 are the main results, in addition to providing greater details on the Q-Coder evolution into the more easily implemented single-adapter version. For more details, including the extension of these results to adaptive coders that use a P-based coding operation, see [16].

Acknowledgements

This work was supported by the Almaden Research Center of IBM, and the author is grateful to colleagues at IBM, including Ron Arps and Robin Williams. The keynote acknowledgement belongs to colleagues Bill Pennebaker and Joan Mitchell, whose multi-rate Q-Coder was a significant breakthrough. The simpler family of single-adapter Q-Coders enlarges this breakthrough, and owes a debt to colleagues Dan Helman and Stephen Todd for the experience with the Monte Carlo approach of Skewadap. The joint work that interactively improved the single-adapter Q-Coder was an enjoyable design experience shared with Pennebaker and Mitchell.

References

1. G. Langdon, "On Adaptation to Multiple-context Binary Sources", IBM Almaden Research Center, Research Report RJ7845, November 1990.

2. G. Langdon and J. Rissanen, "Method for Converting Counts to Coding Parameters", *IBM Technical Disclosure Bul.*, v. **22**, No. 7, December 1979.

3. D. Helman, G. Langdon, and J. Rissanen, "Arithmetic Compression Code Control Parameter Approximation", *IBM Technical Disclosure Bul.*, v. **23**, No. 11, April 1981, 5112-4.

4. G. Langdon and J. J. Rissanen, "A Double-Adaptive File Compression Algorithm", *IEEE Trans. Commun.* **COM-31**, 1253 (1983).

5. S. J. P. Todd, G. G. Langdon, D. R. Helman, and N. Martin, "Statistics Collection for Compression Coding with Randomizing Feature", *IBM Research Report RJ-6414*, IBM Almaden Research Center, San Jose, CA, July 1988 (1981 memo with reference updated).

6. D. R. Helman, G. Langdon, N. Martin, and S. J. P. Todd, "Statistics Collection for Compression Coding with Randomizing Feature," *IBM Tech. Disclosure Bull.* **24**, 4917 (1982).

7. R. Morris, "Counting Large Numbers of Events in Small Registers", *Comm ACM*, vol 21, n 10, October 1978, 840-842.

8. P. Flajolet, "Approximate Counting: A Detailed Analysis," *BIT* **25**, 113 (1985).

9. W. Pennebaker and J. Mitchell, "Probability Estimation for the Q-Coder", unpublished report. (Communicated privately to G. Langdon, March 8, 1985.)

10. W. B. Pennebaker and J. L. Mitchell, "Probability Estimation for the Q-Coder," *IBM J. Res. Develop.*, **32**, 6 (1988) 727-736.

11. G. Langdon, "Some Further Results on Q-Coder Adaptation on Renorms", private communication to W. Pennebaker and J. Mitchell, March 21, 1985, 2 pages.

12. J. Mitchell, private electronic mail to G. Langdon describing use of MPS renorm for adapting with skew coder, 4/16/83.

13. Tom Truong, "Q-Coder Adaptation", unpublished private communication, 4/9/85, 10 pages.

14. R. B. Arps, T. K. Truong, D. J. Lu, R. C. Pasco, and T. D. Friedman, "A Multi-Purpose VLSI Chip for Adaptive Data Compression of Bilevel Images," *IBM J. Res. Develop.* **32**, 6 (1988) 775-795.

15. G. Langdon, "Augend-based Arithmetic Codes and the Golomb Code", IBM Almaden Research Center, Research Report RJ7844, November 1990.

16. G. Langdon, "Adaptation to Binary Sources: Monte Carlo and Q-Coder Algorithms", IBM Almaden Research Center, Research Report RJ7846, November 1990.

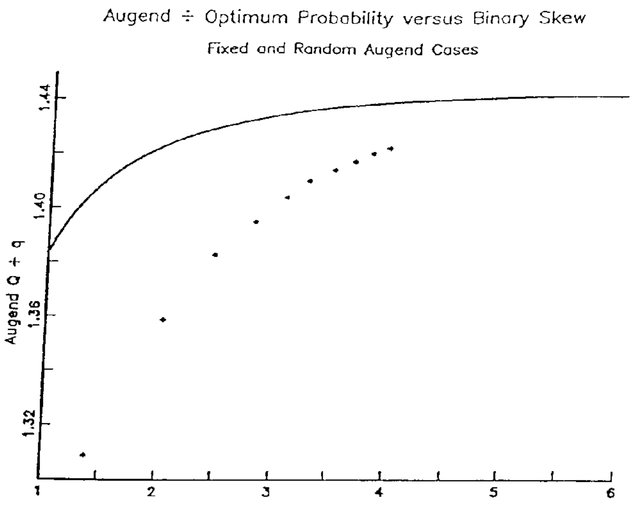

Figure 1. The nonlinear scaling between augend value Q and its associated source parameter q for most efficient coding, plotted versus binary skewness.

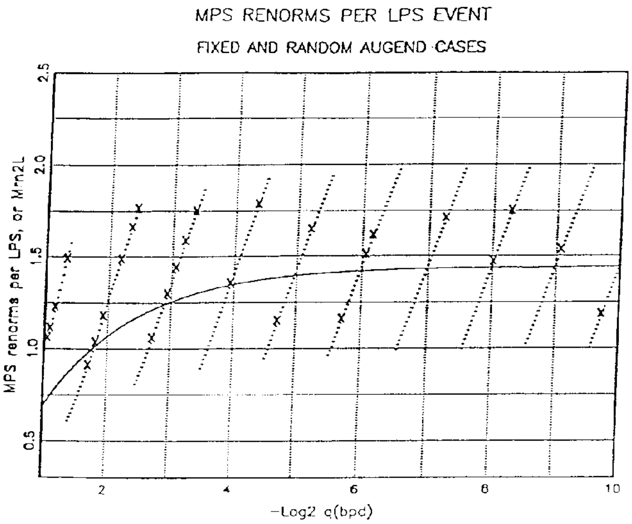

Figure 2. MPS renorms per LPS at the equilibrium point.