Experiments on the zero frequency problem

John G. Cleary, W. J. Teahan*

Department of Computer Science, University of Waikato, New Zealand

1 Introduction

The best algorithms for lossless compression of text are those which adapt to the text being compressed [1]. Two classes of such adaptive techniques are commonly used. One class matches the text against a dictionary of strings seen and transforms the text into a list of indices into the dictionary. These techniques are usually formulated as a variant on Ziv-Lempel (LZ) compression. While LZ compressors do not give the best compression they are widely used because of their simplicity and low execution overhead.

The best compression is obtained by another class of compressors which use adaptive statistical modelling. These split compression into two steps. The first step accumulates a statistical model of the characters seen so far in the input text. As each character is encoded this model is used to generate a probability distribution over those characters which can occur next. *Arithmetic coding* is then used to optimally encode the character which actually does occur with respect to this distribution.

The best compression has been obtained from a series of variants of PPM modelling [1]. PPM models are built up by counting the characters that have occurred following contexts of prior characters. For example, all the characters following 'a' are recorded. The next time 'a' occurs the counts associated with it are used to generate the probability distribution for the following character. The PPM techniques blend together the predictions from contexts of varying lengths to arrive at an overall probability distribution. For practical reasons of memory usage and execution time most PPM variants fix an upper bound to the lengths of the contexts, although recently a variant which uses unbounded length contexts has been very successful [2].

The focus of this paper is the problem of transforming the set of counts accumulated for a particular context into a probability distribution. To simplify our discussion and later experiments we will focus on the case when the alphabet of characters is binary with just two symbols: 0 and 1. Now in a statistical model each context will deliver two counts: C_0, the number of times a 0 has occurred, and C_1, the number of times a 1 has occurred. A naive estimate of the probability of character i could be obtained by the ratio

$$p_i = \frac{C_i}{C_0+C_1} \ .$$

A fundamental problem with this is that it will generate a zero probability if C_0 or C_1 is zero. Unfortunately, a zero probability prevents arithmetic coding from working correctly as the "optimum" code length in this case is infinite. Consequently any estimate of the probabilities must be non-zero even in the presence of zero counts. This problem is called the *zero frequency problem* [7] and was discussed at least as early as Kant.

A well known solution to the problem was formulated by Laplace and is known as Laplace's Law of Succession. It states that for the binary case the estimate for the probability of the next character being an i is given by

*email {jcleary, wjt}@waikato.ac.nz

Method	Escape probability
A	$\frac{1}{n+1}$
B	$\frac{u-t_1}{n}$
C	$\frac{u}{n+u}$
D	$\frac{u/2}{n}$
P	$\frac{t_1}{n} - \frac{t_2}{n^2} - \frac{t_3}{n^3} - \ldots$
X	$\frac{t_1}{n}$
XC	$\frac{t_1}{n}$ when $0 < t_1 < n$,
	$\frac{u}{n+u}$ otherwise

n is the number of tokens seen so far
u is the number of unique tokens seen so far
t_i is the number of unique tokens that have been seen exactly i times so far

Table 1: Different models for the escape probability.

$$p_i = \frac{C_i+1}{n+2} \quad (i = 0, 1)$$

where $n = C_0 + C_1$ is the total count of characters seen so far in the current context.

Laplace's Law of Succession can be generalized both to non-binary alphabets and to allow the added constant to be varied. The *Generalized Law of Succession* states that

$$p_i = \frac{C_i+r}{n+r} \quad (i = 0, 1)$$

where r is a fixed parameter. The probabilities generated by this law are always non-zero provided r is greater than 0. The original version with $r = 1$ is known to be optimal if the prior distribution for the p_i is uniform. However, without making some prior assumption like this there is no way to choose an optimal value for r.

For the PPM data compression scheme [1, 2] the prediction of the next character is based upon the last few characters of the context. However a problem occurs if the next character has never been seen in the current context. In this case, an escape probability is computed to escape down to shorter contexts, the escaping continuing until a context is found which predicts the next character, or until the context is of zero length, in which case a default model is used which predicts all characters in the alphabet with equal probability. Estimating the escape probabilities is just the zero-frequency problem in another guise.

A similar method of escaping is used for the WORD data compression scheme [5]. The model is based upon predicting words rather than single characters. If the word is novel (has not occurred before), then the model escapes down to a character based model similar to PPM.

Methods proposed for estimating the escape probability are shown in Table 1. Methods A and B are described in [3, 8]. Method A is based upon Laplace's Law whereas method B still classifies a character as being novel even if it has occurred once before. Method C proposed in [6] uses the number of times a novel character has occurred before as the basis of its probability. Method D [4] is a minor modification to method C. Experiments with compressing files show that method D is slightly better than method C, but both methods perform better than methods A and B. Methods P, X and XC described in [8] are based upon a Poisson process model and perform better than the other methods in most cases.

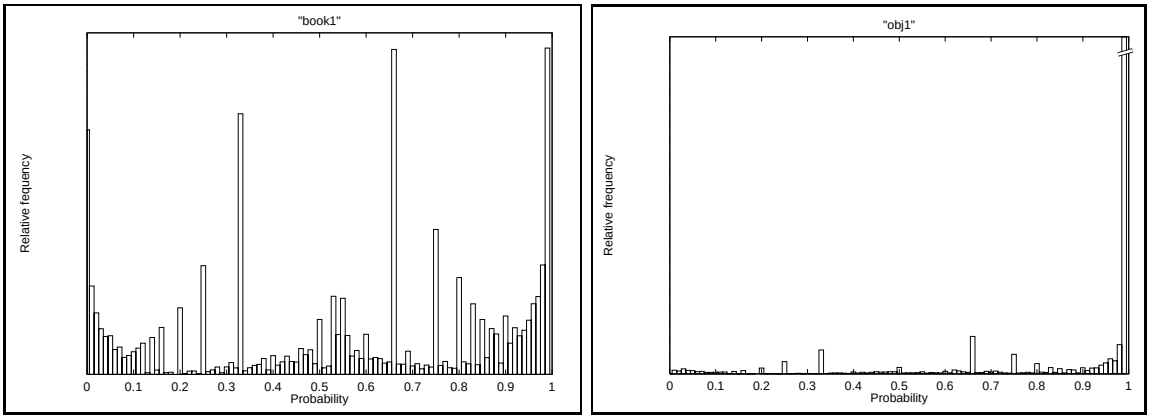

Figure 1: Histograms of context probabilities

Clearly there are many ways that the probability estimation problem can be dealt with. In this paper we report on experimental *a posteriori* measurements of character frequencies as they depend on the prior counts. The next section describes how the measurements were done. Section 3 describes the results of these measurements. Section 4 describes some results on larger alphabets. Section 5 concludes with a summary and discussion of the implications of the results for data compression.

2 Experiments with bitstrings

The context trie data structure described in [2] was used to collect statistics for contexts of unbounded length over a binary alphabet. These contexts were used for collecting statistics on the actual frequencies with which novel events occur. It is these which can be compared experimentally with the probabilities estimated by theories such as Laplace's Law of Succession.

As a first experiment, we determined the probability of a particular bit occurring based upon the C counts in all the current active contexts as the context trie data structures were being updated. The histogram of the frequencies of these observed probabilities may be used to gain insight into the nature of the zero frequency problem for bitstrings. Two such histograms for two files in the Calgary corpus [1], *book1* and *obj2*, are shown in Figure 1. In the graphs, the frequency of the probabilities of a 0 bit using Laplace's Law is plotted.

The histogram for *book1* is typical of many of the graphs for other files in the corpus. Large peaks occur for pairs of small counts. Such peaks can be seen at probabilities $0.25, 0.33, 0.66$ and 0.75 in the graph. Another discernible pattern is that probabilities near 0.5 occur relatively infrequently, most estimates are near the extreme probabilities of 0 and 1. Although not shown by the histograms, these higher frequencies are predominantly explained by the frequent occurrences of contexts where one of the counts is zero.

The histogram for *obj1* is noticeably different from *book1*. Although the same pattern of low frequencies for the mid-range probabilities can be discerned, there is no peak near a probability of zero and a very high peak close to a probability of 1 (this peak is actually ten times higher than it appears as it has been truncated to fit it into the diagram). This peak can be explained by the long runs of 0 bits which occur in the file.

Laplace's Law (with $r = 1$) is optimal if the prior distribution of probabilities is uniform, clearly from Figure 1 the probabilities are very far from this. Although such histograms are useful for demonstrating that the assumption of a uniform prior distribution is invalid,

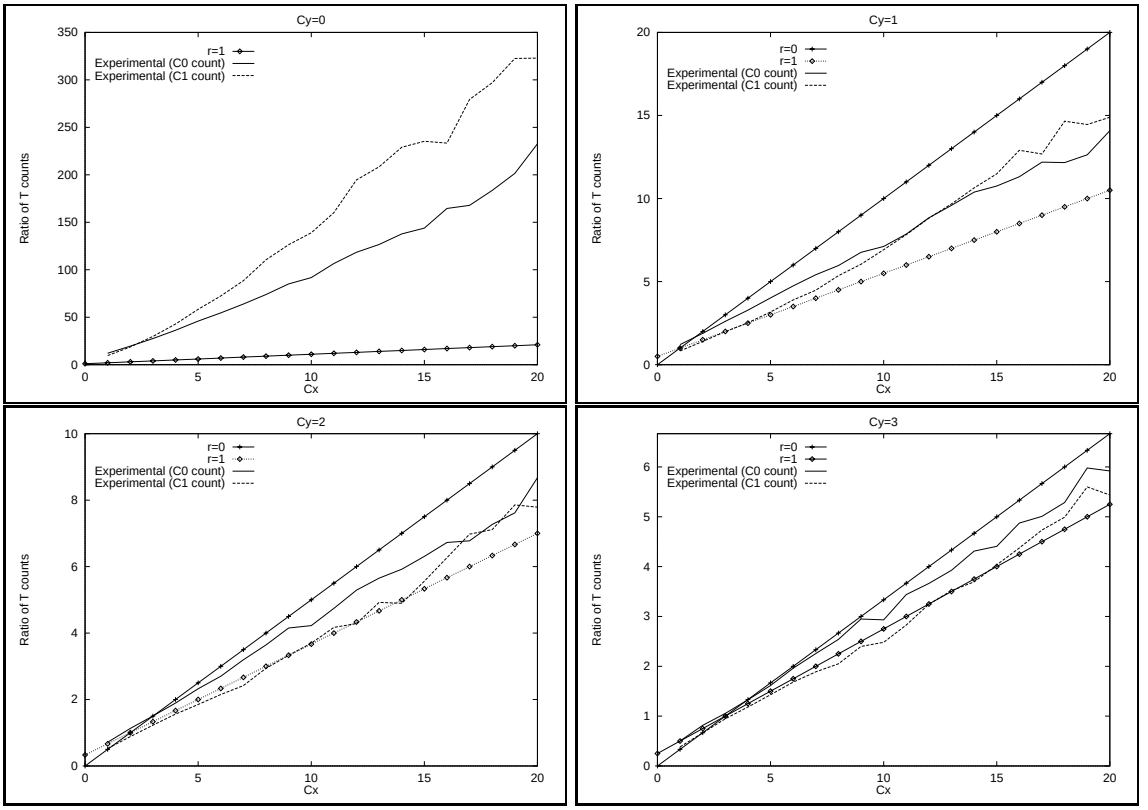

Figure 2: Ratio of T counts for the file *book1*

it doesn't give a real insight into what is going on or what other probability estimators should be used. Another problem is that in deciding which probability to plot, we have to choose some method to overcome the zero frequency problem (Laplace's Law was chosen in this case) even though the whole point of the experiment was to investigate this in the first place.

As another experiment we can tabulate the actual occurrences of bits that follow a particular context. To accumulate the actual observed counts we used two arrays, T_0 and T_1. Whenever a context occurred with associated counts C_0 and C_1, then if a 0 bit actually followed we incremented the count in $T_0[C_0, C_1]$, otherwise we incremented the count in $T_1[C_0, C_1]$. After the fact these counts reflect the actual frequency with which the predicted characters appeared in this context, which can then be compared with the probability estimated using the counts C_0 and C_1.

Figure 2 displays the results for the file *book1*. In each of the diagrams (which we call T plots), the ratio of the T counts ($T_0[C_0, C_1]/T_1[C_0, C_1]$) is plotted against each of the C counts, with one of them (C_0 or C_1) varying along the horizontal axis while the other is constant. These have been labelled "Experimental (C_0 count)" and "Experimental (C_1 count)" in the diagrams. These plots are a convenient way of displaying the results as estimators using the generalized Laplace's Law form straight lines. For comparison, two line plots for ratios predicted by Laplace's Law with $r = 1, (C_x + 1)/(C_x + C_y + 2)$ and $r = 0, C_x/(C_x + C_y + 2)$ are included for comparison where C_x is the count plotted on the horizontal axis, and C_y is the other count which is constant. ($r = 0$ is not plotted when $C_y = 0$ as it then gives invalid zero probabilities).

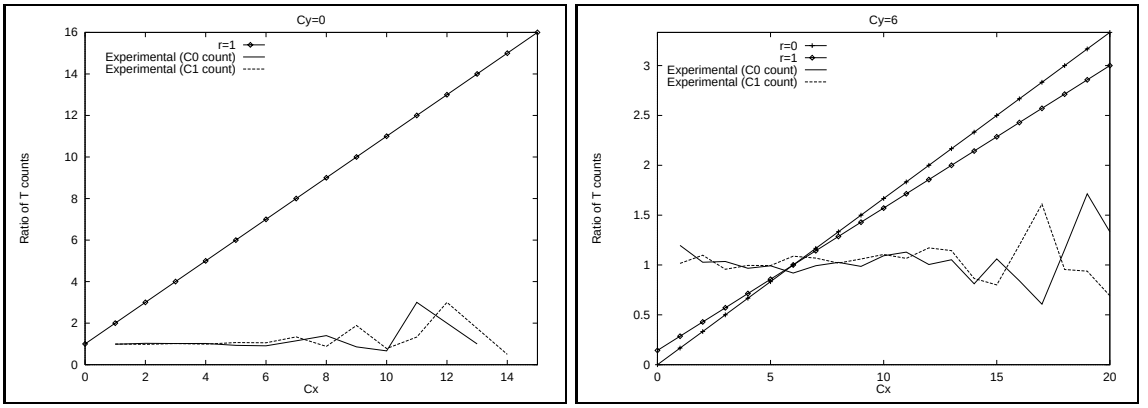

Figure 3: Ratio of T counts for a random file

The four graphs shown in Figure 2 are for the four cases when one of the C counts is either $0, 1, 2$ or 3. C_x on the horizontal axis (C_1 or C_0) varies from 0 to 20. The maximum C counts which actually occurred will quite likely be greater then 20, however, their frequency is low and they do not materially affect the results. There are of course two plots of observed values in each graph, corresponding to predictions when $C_x = C_0$ and $C_y = C_1$ and the other case when $C_x = C_1$ and $C_y = C_0$.

3 Results

The graphs demonstrate a striking dichotomy in the results. For the three graphs where C_y equals 1, 2 and 3 the plotted values lie between the lines for $r = 0$ and $r = 1$. At least for this file Laplace's Law, perhaps with a generalized coefficient of $r = 0.5$ or so, seems to provide an excellent estimator for the true probabilities.

For the deterministic case where $C_y = 0$ the results are strikingly different. The experimental values lie between the lines for r of roughly 0.01 to 0.05. There is also an asymmetry between the results for the two different curves (although they are both within the same order of magnitude). This pattern has been observed for all files in the Calgary corpus.

To gain some insight into the meaning of these results we constructed the T plots for a random bitstring in Figure 3. As expected, since both a 0 or 1 is equally likely in all contexts in a random bitstring, the resulting T plots for all counts approximate a horizontal line at a ratio of the T counts equal to 1.0. Clearly such T plots sharply distinguish the actual files in the corpus from random data. One feature that is common to all these plots is that as the count increases, the amount of noise in the data also increases. This is a simple consequence of the lower number of occurrences of high counts.

The T plots for deterministic cases for the other files in the Calgary corpus are shown in Figure 4. The plots for the files *book2*, *paper2*, and *progc* show a similar pattern to that observed for *book1*. The graphs are linear for low counts, and all fit between the lines for $r = 0.1$ and $r = 0.05$. In all cases the data at higher counts is very noisy. This tendency is more marked for smaller files such as *paper2* and *progc*.

The T plots for the other files shown in the diagram are more exotic. The plots for the files *news*, *obj2*, both show large fluctuations. *news* contains peaks which occur at the same point for both of the C counts (at $C_x = 9$). It is hard to know what to make of these large abrupt changes as C_x increases. One possibility in *news* is that there is some pattern which repeats 9 times through the file leading to many contexts repeating 9 times.

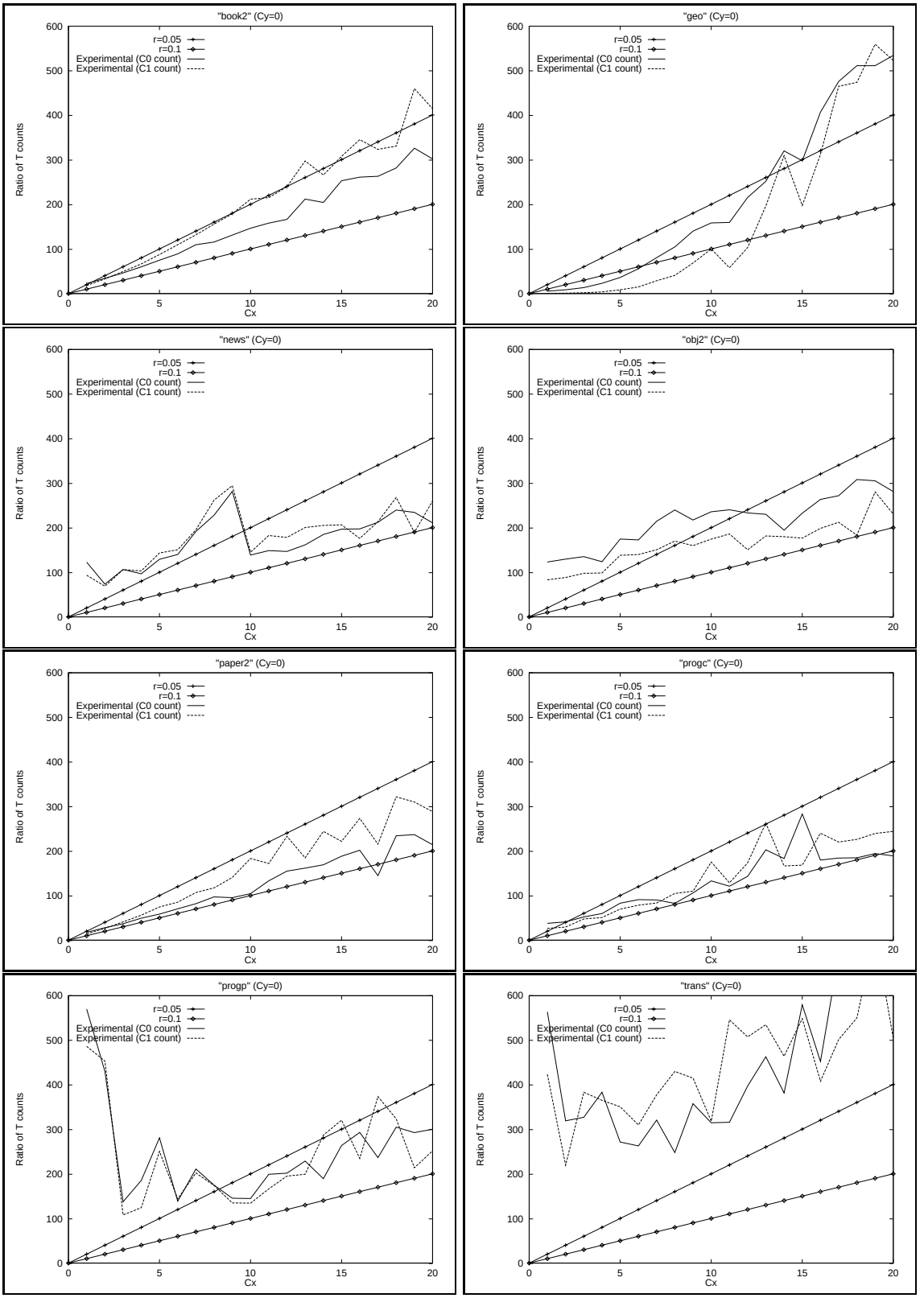

Figure 4: Ratio of T counts for deterministic contexts

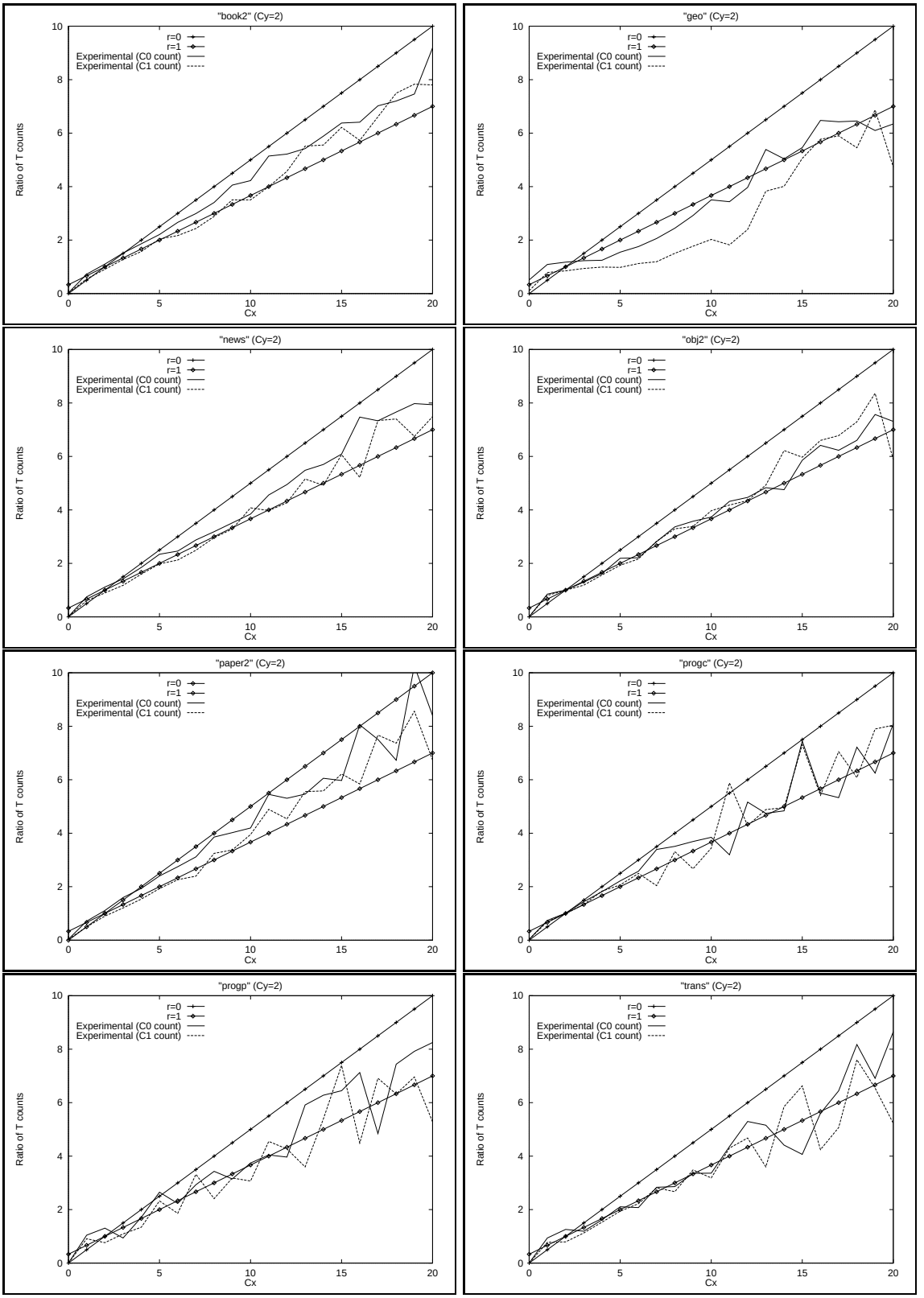

Figure 5: Ratio of T counts for non-deterministic contexts

Examining *news* by eye shows no such obvious 9 fold repetition. The file does have a clear repetitive structure consisting of a number of articles from a news group with stereotyped headers at the beginning of each, however, there are not 9 articles! The largest deviation from Laplace's Law is seen for *trans* where the ratio of the T counts is over 200 for all of the counts and nearly reaches 900 for $C_x = 20$.

The plot for the file *geo* is also interesting. Although the later part of this plot is similar to *book1*, there is a noticeable drop in the curve for lower counts. This is an indication that the *geo* is relatively random, with the file becoming increasingly more deterministic as the counts increase. It is noteworthy that *geo* is one of the harder files in the corpus to compress and that techniques that compress the other files in the corpus poorly sometimes do very well on this file [1]. It's anomalous behavior may be because it is more random and has less structure than the other files.

Figure 5 shows T plots for the other files in the Calgary corpus with $C_y = 2$. In all cases except *geo* these verify our earlier observation that Laplace's Law provides a good approximation to the observed ratios. Emphasizing its anomalous behavior the plots for non-zero counts for *geo* noticeably deviate from Laplace's Law, with high counts (6 and above) having plots similar to those observed for the random file.

4 Experiments with character strings

Another possibility for investigating the zero frequency problem is to examine 8-bit character string contexts as opposed to bitstring contexts. A similar approach to that taken for bitstrings can be taken here. We can count the number of times novel characters occur. However, the increased size of the alphabet complicates the collection of the data, and at the same time the frequency counts are much lower, hence there is more noise. We can, however, examine the distribution of novel events that occur in deterministic contexts where only a single character has been seen before. In these contexts, we can count both the number of novel characters and the number of times the deterministic character (the one that has been seen before) follow. These counts can be collected for all deterministic contexts that have the same prior count for each character, regardless of the length of the context. The results we obtain can be compared with the algorithms proposed for computing the escape probability shown in Table 1.

Taking the various estimators from Table 1 and applying them to this case, we have $u = 1$ and $t_n = 1$ with all other $t_i = 0$. So for methods A and C the estimator for the escape probability is $\frac{1}{n+1}$ which is just Laplace's Law with $r = 1$. The estimator for B is $\frac{1}{n}$ and for D $\frac{1}{2n}$. P, X and XC break down in this case (because most of the t_i are zero) and default to the estimator $\frac{1}{n+1}$.

Figure 6 shows the results for the file *book1*. Four plots are shown for the letters 'a', 'b' and 'e' and for the space character. The count plotted on the horizontal axis is the number of times the character had been observed in the context before. (Remember that the contexts for which data is being collected are deterministic, so only the character that is plotted has been observed before). The "escape ratio" shown on the vertical axis is the ratio of the deterministic character count divided by the novel character count. (These plots are identical to our earlier T plots in the binary case).

Also shown on the graph is the estimator $\frac{1}{n+1}$ for the escape probability. The results here are somewhat more mixed and inconsistent than in the earlier binary T plots. The plots for 'a', 'b' and 'e' lie well above the $r = 1$ line and are roughly in the range $r = 0.5$ to 0.25. The plot for space is completely different as it lies well below the $r = 1$ line, perhaps

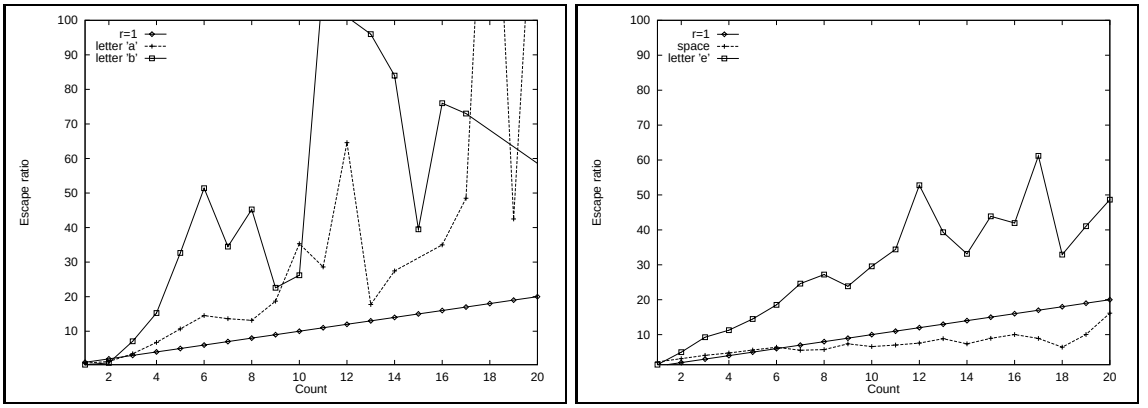

Figure 6: Escape ratio for the file *book1*

more closely approximating random input. Why this is so is not clear.

5 Conclusions

For a binary alphabet we have found experimentally that probability estimates using Laplace's Law are accurate only if both the characters have occurred at least once before. If one of the counts is zero then the estimates diverge widely from Laplace's predictions. The implications for compression are even stronger than this data indicate as the expected entropy is insensitive to the estimator used when both counts are non-zero, however, the dependence is very sharp in the deterministic case. For example, the results indicate that the parameter r in the generalized Laplace's Law should be in the region of 0.05 for the deterministic case. If instead an r of 1 were used the expected compression is reduced by a factor of 3. In contrast for non-deterministic contexts the expected compression never varies by more than 5% over the same range for r.

So, even if deterministic contexts are used only relatively infrequently during compression, they could still have a very large effect on the overall compression. While more difficult to observe and interpret, a few results for a non-binary alphabet indicate that it is still true that in deterministic contexts there can be large deviations from commonly used estimators.

These results indicate that in constructing a statistical text compression system deterministic contexts should be treated as special cases. Also given the wide variations between the measurements for the deterministic case on different files it is probably worthwhile adapting the estimator to the actual data.

References

[1] Bell, T.C., Cleary, J.G. and Witten, I.H. (1990) *Text compression.* Prentice Hall, Englewood Cliffs, NJ.

[2] Cleary, J.G., Teahan, W.J. and Witten, I.H. (1995) "Unbounded length contexts for PPM," submitted to DCC'95.

[3] Cleary, J.G. and Witten, I.H. (1984) "A comparison of enumerative and adaptive codes," *IEEE Transactions on Information Theory*, **30**(2), 306–315.

[4] Howard, P.G. (1993) "The design and analysis of efficient lossless data compression systems," Technical Report No. CS–93–28, Department of Computer Science, Brown University, Providence, Rhode Island.

[5] Moffat, A. (1987) "Word based text compression," Research Report, Department of Computer Science, University of Melbourne, Parkville, Victoria 3052, Australia.

[6] Moffat, A. (1990) "Implementing the PPM data compression scheme," *IEEE Transactions on Communications*, **38**(11), 1917–1921.

[7] Roberts, M.G. (1982) "Local order estimating Markovian analysis for noiseless source coding and authorship identification," Ph.D. thesis, Stanford University, Stanford, California.

[8] Witten, I.H. and Bell, T.C. (1991) "The zero-frequency problem: estimating the probabilities of novel events in adaptive text compression," *IEEE Transactions on Informaton Theory*, **37**(4), 1085–1094.