Deplump for Streaming Data

Nicholas Bartlett, Frank Wood, Department of Statistics

COLUMBIA UNIVERSITY IN THE CITY OF NEW YORK

Arithmetic Encoding [6]

An arithmetic encoder uses of the output from a predictive model to create a 1-1 correspondence between input streams and sub-intervals of the unit interval [0,1). The input is then encoded using the sub-interval to which it corresponds. Input streams which are likely under the predictive model correspond to larger sub-intervals and thus require fewer bits to encode.

For example:

Context (sequence thus far)	Probability of next symbol		
	$P(\mathtt{a}) = 0.425$	$P(\mathtt{b}) = 0.425$	$P(\Box) = 0.15$
b	$P(\mathtt{a}\,\vert\,\mathtt{b}) = 0.28$	$P(\mathtt{b}\,\vert\,\mathtt{b}) = 0.57$	$P(\Box\,\vert\,\mathtt{b}) = 0.15$
bb	$P(\mathtt{a}\,\vert\,\mathtt{bb}) = 0.21$	$P(\mathtt{b}\,\vert\,\mathtt{bb}) = 0.64$	$P(\Box\,\vert\,\mathtt{bb}) = 0.15$
bbb	$P(\mathtt{a}\,\vert\,\mathtt{bbb}) = 0.17$	$P(\mathtt{b}\,\vert\,\mathtt{bbb}) = 0.68$	$P(\Box\,\vert\,\mathtt{bbb}) = 0.15$
bbba	$P(\mathtt{a}\,\vert\,\mathtt{bbba}) = 0.28$	$P(\mathtt{b}\,\vert\,\mathtt{bbba}) = 0.57$	$P(\Box\,\vert\,\mathtt{bbba}) = 0.15$

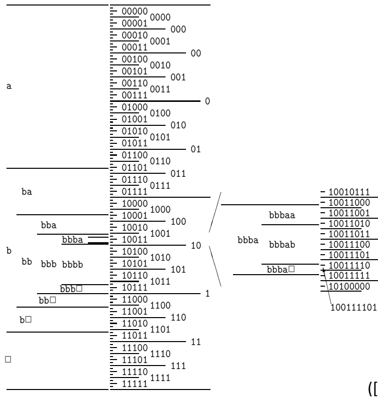


Figure 6.4. Illustration of the arithmetic coding process as the sequence bbba□ is transmitted.

([6] MacKay 2003, 114)

Batch Deplump [4] and the Sequence Memoizer [11]

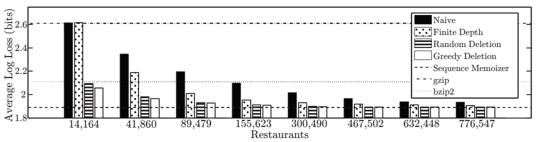
Batch deplump is an arithmetic compressor powered by a nonparametric Bayesian model called the sequence memoizer. The performance of deplump has been demonstrated on benchmark corpora, including the Calgary Corpus [2]. The aggregate performance of deplump is equal or better than that of comparable state of the art, general purpose, lossless compressors [4]. The spatial complexity of the sequence memoizer model grows unboundedly making batch deplump unrealistic for streaming data.

The performance of batch deplump on the Calgary Corpus is compared to PPM [3] and CTW [10]. Performance is measured in average bits per byte (lower is better). Bold text indicates best performance.

		DEPLUMP		PPM		CTW
File	Size	1PF	UKN	PPM*	PPMZ	CTW
bib	111261	1.73	**1.72**	1.91	1.74	1.83
book1	768771	**2.17**	2.20	2.40	2.21	2.18
book2	610856	**1.83**	1.84	2.02	1.87	1.89
geo	102400	**4.40**	4.40	4.83	4.64	4.53
news	377109	**2.20**	2.20	2.42	2.24	2.35
obj1	21504	**3.64**	3.65	4.00	3.66	3.72
obj2	246814	2.21	**2.19**	2.43	2.23	2.40
paper1	53161	2.21	**2.20**	2.37	2.22	2.29
paper2	82199	**2.18**	2.18	2.36	2.21	2.23
pic	513216	0.77	0.82	0.85	**0.76**	0.80
progc	39611	2.23	**2.21**	2.40	2.25	2.33
progl	71646	1.44	**1.43**	1.67	1.46	1.65
progp	49379	1.44	**1.42**	1.62	1.47	1.68
trans	93695	1.21	**1.20**	1.45	1.23	1.44
avg.		**2.12**	2.12	2.34	2.16	2.24
w. avg.		**1.89**	1.91	2.09	1.93	1.99

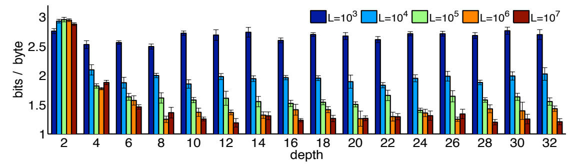
([4] Gasthaus, J.; Wood, F. and Teh, Y. W.)

Results 1



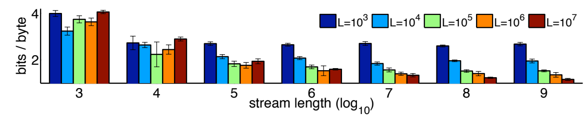
The sequence memoizer model makes use of a suffix tree data structure. To create a streaming deplump compressor it is necessary to approximate the model using a data structure which does not grow with the length of the input sequence. Forgetting (pruning) of the suffix tree is used to achieve this and was demonstrated to have excellent empirical performance [1]. Results here are measured in bits per byte and shown versus an upper limit on the number of nodes in the suffix tree. Comparisons are made to naïve and other simple strategies to obtain constant spatial complexity.

Results 2



Streaming deplump was evaluated on a complete Wikipedia .xml dump [9]. For this result 10 100MB chunks of text were sampled with replacement and compressed using models limited to depths varying from 2 to 32. Performance is shown in bits per byte and each group contains results for models with a different upper limit on the node count of the data structure. As expected, larger models generally perform better. Using a larger depth appears advantageous up to ≈ 16.

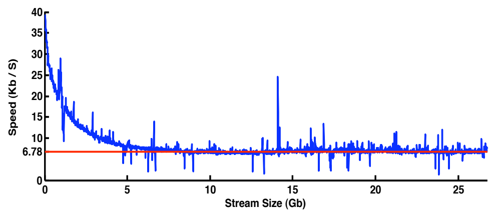
Results 3



The performance of streaming deplump as a function of stream length was also evaluated using the Wikipedia .xml dump. For this result stream lengths ranging from 10^3 to 10^9 bytes were compressed using models of varying size. Performance is again measured in bits per byte. The performance clearly improves as the length of the sequence increases.

Linear Time Verification

Streaming deplump has asymptotic properties appropriate for streaming data. Shown here is the speed of the compressor on the entire Wikipedia corpus. The speed of the compressor is plotted with the size of the input stream. After an initial period the speed remains constant as the stream length increases. Streaming deplump compresses the 26.8Gb corpus to 4.0Gb, compared to 7.8Gb with gzip and 3.8Gb with paq9a.

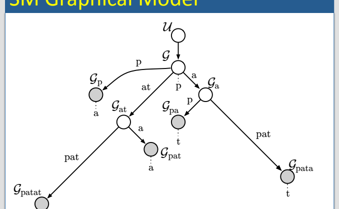


Hierarchical Pitman-Yor Processes (HPYP) [7,8]

The PYP is a distribution over distributions and is a generalization of the Dirichlet process [7]. Hierarchically composing PY processes is a way to smooth distribution estimates. The parameters of the process are known as discounts and concentrations and control the amount of smoothing in the model.

$$
\begin{aligned}
\mathcal{G}_1 | d_1, c_1, \mathcal{G}_0 &\sim \mathcal{PY}(d_1, c_1, \mathcal{G}_0) \\
\mathcal{G}_2 | d_2, c_2, \mathcal{G}_1 &\sim \mathcal{PY}(d_2, c_2, \mathcal{G}_1) \\
\theta_i | \mathcal{G}_2 &\sim \mathcal{G}_2 \quad i = 1, \ldots, N
\end{aligned}
$$

SM Graphical Model



Additional Approximations

Node Representation

- Each node can be represented in a constant amount of space [5].
- Each node is associated with a context observed in the input sequence.
- The number of times a given context has been observed in the input sequence grows unboundedly with the length of the input sequence.
- Operations on the suffix tree necessary for incremental construction and estimation of the model require space and time which grows with the number of times contexts have been observed in the input sequence [5].
- Node representations are approximated by placing an upper bound on the recorded number of occurrences in the input sequence of any given context.

Suffix Tree

- Suffix tree data structures label edges using pointers into the input sequence.
- The input sequence grows linearly.
- The removal of nodes through the pruning mechanism does not guarantee that the edges will be labeled using only a small subset of the original input sequence.
- The suffix tree data structure is approximated by only allowing edges to be labeled by a fixed length suffix of the input sequence.
- Edges which cannot be labeled are removed from the model along with the descending sub tree.
- To minimize the impact of this approximation edges are updated incrementally to point to later sections of the input sequence.

References

[1] Bartlett, N.; Pfau, D. & Wood, F. Forgetting Counts: Constant Memory Inference for a Dependent Hierarchical Pitman-Yor Process. Proceedings of the 27th International Conference on Machine Learning, 2010, 63-70.

[2] Bell, T., Witten, I.H., and Cleary, J.G. Modeling for text compression. ACM Computing Surveys (CSUR), 1989, 21(4) 557–591.

[3] Cleary, J. G. and Teahan, W. J. Unbounded length contexts for PPM. The Computer Journal, 1997, 40:67–75.

[4] Gasthaus, J.; Wood, F. and Teh, Y. W. Lossless compression based on the Sequence Memoizer. Data Compression Conference, 2010, 337-345.

[5] Gasthaus, J. and Teh, Y. W. Improvements to the Sequence Memoizer. Proceedings of Neural Information Processing Systems, 2011, 685-693.

[6] MacKay, D. Information theory, inference, and learning algorithms. Cambridge University Press, 2003.

[7] Pitman, J. and Yor, M. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. Annals of Probability, 1997, 25:855–900.

[8] Teh, Y. W. A hierarchical Bayesian language model basedon Pitman-Yor processes. In Proceedings of the Association for Computational Linguistics ,2006, 985-992.

[9] Wikipedia, 2010. URL: http://download.wikimedia.org/enwiki/.

[10] Willems, F. M. J. , 2009. CTW website. URL: http://www.ele.tue.nl/ctw/.

[11] Wood, F., Archambeau, C., Gasthaus, J., James, L., and Teh, Y. W. A stochastic memoizer for sequence data. In Proceedings of the 26th International Conference on Machine Learning, 2009, 1129-1136.