

PPM revisited

Christian Steinruecken, Zoubin Ghahramani, David MacKay
University of Cambridge

PPM revisited

Cleary & Witten (1984)³

"It's been $2^5 - 1$ years since PPM."

And PPM is still going strong.

Three improvements to PPM

Steinruecken & al (2015)¹

We improve PPM in the following way:

- Replacing the **escape mechanism** with **blending** (full inheritance)
- Giving separate hyper-parameters to different groups of contexts
- Updating these hyper-parameters **online** using gradient information

We call the resulting model "**PPM-DP**" — PPM with dynamic parameter updates.

PPM-DP beats PPMZ, PPMII and other known PPM variants on human text.
It also beats CTW, DMC, LZ, CSE, and BWT.

Quick note

Any lossless data compression algorithm:

- Defines a reversible mapping from **inputs** to **outputs**.
- This mapping defines a **probability distribution** over the **space of input objects**.
- In this talk, our input space is the set of **finite, unbounded sequences**.

Sequence Models

Sequence models are **probability distributions** over sequences:

$$P(x_1, x_2, x_3, x_4, x_5, \dots x_N)$$

where symbols x are from some discrete alphabet \mathcal{X} .

► Note that P also defines a distribution over sequence lengths.

Aside

Any code that translates messages to binary strings implicitly defines a probability distribution P :

$$P(x_1 \dots x_N) = \frac{1}{Z} e^{-L(x_1 \dots x_N)}$$

where $x_1 \dots x_N$ = message,
and $L(x_1 \dots x_N)$ = length of encoded string.

► Any lossless compression algorithm necessarily defines a probabilistic sequence model.

მსოფლიოში, როგორც ჩვენ
ვიცით, ეს არის, რომ
შეიტვალის. თუმცა მაცდური
მზარდი რესურსების მოხაცემთა
შენახვის, მე შიგნად არ დაგვრჩა
კიდევ ერთი bit. მოტივი,
თავმდაბალი მოკრძალვებით,
პატივი თითოეული bit ზღვიდან
მონაცემები და შეინართ
თითოეული პატივისცემით ამ
ფრთხილად შერჩეული დაბალი
ენტროპია მოხყობა.



010001001111011110010011101110
110110010101001110110111011101
010001101111011010100011010000
01110010110001000110111100111011
1110100010011000011111100111011
00110111000010111110000101001010
1101110000000010100001000111001
00100001001011100101101110000
1110111001011111111110010011
1000000100000100000001100011000
110011011011010011111011101101
00001110011000101110100010101
1000001101111101000011101001001
01011100011001110010111001011011
0101000001101010110001110011011011

Let's think about distributions!

An algorithm's implicit distribution P uniquely determines the compression effectiveness of the algorithm on all inputs.*

So we could think about **distributions** rather than **algorithms**.

Advantages:

- Separates **modelling** from **coding**.
- Makes it easier to **understand** the compressor, **modify** its properties, or **use it as a module** in other compressors.

* If the algorithm's mapping is compact, i.e. leaves no gaps.

How a sequence model works

Assign probability mass to the space of finite sequences \mathcal{X}^* , for a given alphabet \mathcal{X} .

$$P(x_1, x_2, x_3, x_4, x_5, \dots x_N, N)$$

- Sequence models are inherently non-parametric.
- Want to jointly model sequence length N and content $x_1 \dots x_N$.

The probability of a sequence

All distributions over sequences can be factorised as follows:

$$P(x_1 \dots x_N | N) = \prod_{n=1}^N P(x_n | x_1 \dots x_{n-1})$$

The conditional distribution $P(x_n | x_1 \dots x_{n-1})$ may not be cheap to compute.
(But useful models make it cheap.)

The PPM family of algorithms

Any PPM-like algorithm:

- collects context-dependent symbol occurrence counts \mathcal{M}_s .
(Often using a trie data structure.)
- defines a probabilistic model that computes $P(x_n | x_1 \dots x_{n-1})$ from the collected counts.
- uses arithmetic coding to sequentially compress / decompress each symbol x_n according to $P(x_n | x_1 \dots x_{n-1})$.

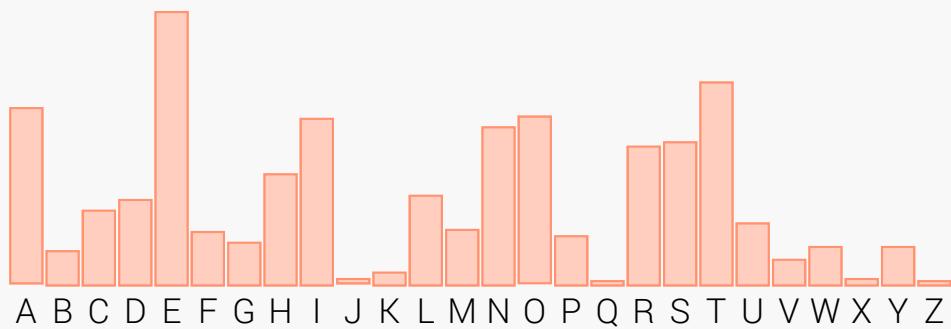
➤ PPM-like algorithms differ primarily in their choice of probabilistic model P .

A partial history of the PPM family

Method	Depth	Smoothing	Updates	References
PPM{A,B}	finite	escape	full	Cleary & Witten (1984) ³
PPMC	finite	escape	shallow	Moffat (1990) ¹⁴
PPMD	finite	escape	shallow	Howard & Vitter (1991) ¹⁵
PPM*	unbounded	escape	full	Cleary & al (1995) ¹⁶
PPM*SB	unbounded	various	various	Bunton (1997) ¹⁷
PPMII	finite	custom	custom	Шкарин (2001) ²²
SeqMem	unbounded	blending	various	Wood & al (2009) ¹⁸
Deplump 1	unbounded	blending	various	Gasthaus & al (2010) ²⁰
Deplump 2	unbounded	blending	various	Bartlett & Wood (2011) ²¹

Histogram of English text

Norvig (2013)⁴



This is a histogram of letter occurrences in English text, obtained by Norvig (2013)⁴ using millions of books from Google's n-gram corpus.

Learning a histogram

We can **learn** the histogram \mathcal{M} from the data, by counting how often we see each symbol.

$$\underbrace{x_1, x_2, x_3, x_4, x_5, \dots x_N}_{\text{summarised by } \mathcal{M}}$$

$\mathcal{M}(x)$ denotes the number of times x occurred in the sequence.

$|\mathcal{M}|$ is the total number of symbols in the sequence. (Equal to N in this case.)

Dirichlet process (DP)

$$P(x_{N+1} | \mathcal{M}, \alpha, H) = \frac{\mathcal{M}(x_{N+1})}{|\mathcal{M}| + \alpha} + \frac{\alpha}{|\mathcal{M}| + \alpha} H(x_{N+1})$$

where: α is a concentration parameter,
 \mathcal{M} the multiset of past symbol occurrences,
and H is a base distribution.

A Dirichlet process [learns](#) the symbol distribution of the sequence.

Hierarchical predictions

We can [combine histograms of different depths](#) to make improved predictions.

E.g. we can use the DP of context H as [base distribution](#) for the DP of context TH . (And also for the DPs of context RH, SH, CH etc.)

Here is such a hierarchical DP:

$$P_s(x_{N+1} | \mathcal{M}_s, \alpha, H) = \frac{\mathcal{M}_s(x_{N+1})}{|\mathcal{M}_s| + \alpha} + \frac{\alpha}{|\mathcal{M}_s| + \alpha} P_{\text{suf}(s)}(x_{N+1})$$

(For $\alpha = 1$, this hierarchical model corresponds to [PPMA](#) with blending + shallow updates.)

Pitman–Yor process

$$G(x_{N+1} | \mathcal{M}) = \frac{\mathcal{M}(x_{N+1}) - \beta t}{|\mathcal{M}| + \alpha} + \frac{\alpha + U\beta}{|\mathcal{M}| + \alpha} H(x_{N+1})$$

where: α is a concentration parameter,
 β a discount parameter,
 \mathcal{M} the multiset of symbol occurrences,
 U the number of unique symbols in \mathcal{M} ,
and H is a base distribution.

(This model is used in the [Sequence Memoizer](#) and in [Deplump](#).)

Simplified Pitman–Yor process (1TPD)

$$G(x_{N+1} | \mathcal{M}) = \frac{\mathcal{M}(x_{N+1}) - \beta}{|\mathcal{M}| + \alpha} \mathbf{1}_{[x \in \mathcal{M}]} + \frac{\alpha + U\beta}{|\mathcal{M}| + \alpha} H(x_{N+1})$$

where: α is a concentration parameter,
 β a discount parameter,
 \mathcal{M} the multiset of symbol occurrences,
 U the number of unique symbols in \mathcal{M} ,
and H is a base distribution.

Inference in this process is much cheaper than in standard Pitman–Yor, with very similar predictive power.
This process is used in PPM-DP.

Making things even better

Some things are quite good now:

- ✓ The hierarchical architecture allows effective sharing of information for human text.
- ✓ The simplified Pitman–Yor process models the histograms pretty well.

But a few things could be improved:

- How to set the parameter of each histogram learner?
(Answer: with global parameters, set $\alpha = 0.5$ and $\beta = 0.85$)
- Some histograms seem rather different from others, and maybe they should be learned differently!

Grouping histograms into classes

- Histograms get sparser the deeper the context s .
Idea 1: Make the parameters (α_d, β_d) depend on context depth.
- Contexts in which only 1 or 2 unique symbol occur seem different from those in which many different symbols occur.
Idea 2: Make the parameters (α_f, β_f) depend on the number of unique symbols observed so far.
- **Idea 3:** Do both. So we have a matrix of parameters $(\alpha_{df}, \beta_{df})$.

How to set these parameters?

So let's use depth-dependent and fanout-dependent parameters (α_{df} , β_{df}) to make hierarchical predictions from our histograms.

But how should we set these parameters?

(Spoiler alert: gradients come to the rescue.)

Parameter gradients

We want to minimise the [output length](#), i.e. the [Shannon information content](#) of the sequence given P .

$$\nabla \log_2 \frac{1}{P(x_1 \dots x_N)} = \sum_{n=1}^N \nabla \log_2 \frac{1}{P(x_n | x_1 \dots x_{n-1})}$$

Parameter gradients

For any given sequence $x_1 \dots x_N$, the [parameter gradients](#)

$$\frac{\partial P(x_1 \dots x_N)}{\partial \beta_{df}} \quad \frac{\partial P(x_1 \dots x_N)}{\partial \alpha_{df}}$$

can be computed analytically, at (nearly) [the same cost](#) as evaluating $P(x_1 \dots x_N)$.

We can use a conjugate-gradient optimiser to find good settings for the β_{df} , α_{df} .

Parameter gradients

$$\begin{aligned}\frac{\partial P_{\textcolor{teal}{s}}(x_{N+1} \mid x_1 \dots x_N)}{\partial \beta_{df}} &= \frac{-1}{|\mathcal{M}_{\textcolor{teal}{s}}| + \alpha_{df}} \\ &+ \frac{U_{\textcolor{teal}{s}}}{|\mathcal{M}_{\textcolor{teal}{s}}| + \alpha_{df}} P_{\text{suf}(\textcolor{teal}{s})}(x_{N+1} \mid x_1 \dots x_N) \\ &+ \frac{\alpha_{df} + U_{\textcolor{teal}{s}} \beta_{df}}{|\mathcal{M}_{\textcolor{teal}{s}}| + \alpha_{df}} \frac{\partial P_{\text{suf}(\textcolor{teal}{s})}(x_{N+1} \mid x_1 \dots x_N)}{\partial \beta_{df}}\end{aligned}$$

$$\begin{aligned}\frac{\partial P_{\textcolor{teal}{s}}(x_{N+1} \mid x_1 \dots x_N)}{\partial \alpha_{df}} &= \frac{\beta_{df} - \mathcal{M}_{\textcolor{teal}{s}}(x)}{(|\mathcal{M}_{\textcolor{teal}{s}}| + \alpha_{df})^2} \\ &+ \frac{|\mathcal{M}_{\textcolor{teal}{s}}| - U_{\textcolor{teal}{s}} \beta_{df}}{(|\mathcal{M}_{\textcolor{teal}{s}}| + \alpha_{df})^2} P_{\text{suf}(\textcolor{teal}{s})}(x_{N+1} \mid x_1 \dots x_N) \\ &+ \frac{\alpha_{df} + U_{\textcolor{teal}{s}} \beta_{df}}{|\mathcal{M}_{\textcolor{teal}{s}}| + \alpha_{df}} \frac{\partial P_{\text{suf}(\textcolor{teal}{s})}(x_{N+1} \mid x_1 \dots x_N)}{\partial \alpha_{df}}\end{aligned}$$

PPM-DP: online optimisation of the parameters

Online optimisation:

We can use the gradient information to adjust the parameters online, just like we learn the histograms online.

Results

	gzip	bzip2	CSE	Izip	CTW	ppmz2	PPMII	N8	N16
alice29.txt	2.850	2.272	2.192	2.551	2.075	2.059	2.033	2.018	2.015
asyoulik.txt	3.120	2.529	2.493	2.848	2.322	2.309	2.308	2.284	2.280
cp.html	2.593	2.479	2.555	2.478	2.307	2.158	2.139	2.121	2.113
fields.c	2.244	2.180	2.276	2.152	1.990	1.896	1.845	1.820	1.799
grammar.lsp	2.653	2.758	2.750	2.709	2.384	2.300	2.268	2.210	2.199
kennedy.xls	1.629	1.012	0.891	0.409	1.009	1.373	1.168	1.547	1.519
lcet10.txt	2.707	2.019	1.928	2.233	1.832	1.794	1.791	1.783	1.773
plrabn12.txt	3.225	2.417	2.283	2.746	2.185	2.194	2.202	2.172	2.171
ptt5	0.816	0.776	0.772	0.618	0.796	0.754	0.757	0.767	0.768
sum	2.671	2.701	3.024	1.982	2.571	2.538	2.327	2.448	2.399
xargs.1	3.308	3.335	3.494	3.369	2.962	2.850	2.852	2.775	2.771
bib	2.509	1.975	1.975	2.199	1.833	1.718	1.726	1.715	1.697
book1	3.250	2.420	2.268	2.717	2.180	2.188	2.185	2.165	2.166
book2	2.700	2.062	1.977	2.224	1.891	1.839	1.827	1.819	1.809
geo	5.345	4.447	5.354	4.185	4.532	4.578	4.317	4.383	4.379
news	3.063	2.516	2.525	2.521	2.350	2.205	2.188	2.196	2.177
obj1	3.837	4.013	4.462	3.506	3.721	3.667	3.506	3.577	3.574
obj2	2.628	2.478	2.711	1.991	2.398	2.241	2.160	2.213	2.173
paper1	2.789	2.492	2.540	2.598	2.291	2.212	2.190	2.179	2.170
paper2	2.887	2.437	2.412	2.655	2.229	2.185	2.173	2.162	2.158
progC	2.677	2.533	2.604	2.532	2.337	2.257	2.198	2.207	2.192
progl	1.804	1.740	1.712	1.666	1.647	1.447	1.437	1.459	1.415
progp	1.811	1.735	1.778	1.671	1.679	1.449	1.445	1.513	1.432
trans	1.610	1.528	1.598	1.420	1.443	1.214	1.222	1.241	1.195

Summary

We talked about...

- **PPM-DP**: Three improvements that improve the compression effectiveness of **PPM** on human text and other common data.
- New state-of-the-art compression results.

Paper

Improving PPM with dynamic parameter updates

Christian Steinruecken, Zoubin Ghahramani, David MacKay

Data Compression Conference 2015

Maybe we also care about speed etc.

Other things we might care about:

- Speed (compression efficiency)
- Simplicity (implementation efficiency)
- Interpretability
- Parallelisation

What about PAQ?

Compressors in the PAQ family combine predictions from many probabilistic models (including PPM-like models) to form a single prediction.

The PAQ family of compressors was developed by Matt Mahoney (2000²⁴, 2002²⁵, 2005²⁶).

References

- [1] Christian Steinruecken; Zoubin Ghahramani; David MacKay. *Improving PPM with dynamic parameter updates*, 2015-04. In Proceedings of the Data Compression Conference, pp. 193-202, ed. Ali Bilgin; Michael W. Marcellin; Joan Serra-Sagristà; James A. Storer. IEEE Computer Society. ISSN 1068-0314.
- [2] Claude Elwood Shannon. *A Mathematical Theory of Communication*, 1948. In The Bell System Technical Journal, Vol. 27, pp. 379-423, 623-656. ISSN 0005-8580. [[PDF: http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf](http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf)]
- [3] John Gerald Cleary; Ian Hugh Witten. *Data compression using adaptive coding and partial string matching*, 1984-04. In IEEE Transactions on Communications, Vol. 32, No. 4, pp. 396-402. ISSN 0090-6778. [[download: http://ieeexplore.ieee.org/ielx5/26/23982/01096090.pdf?tp=&arnumber=1096090&isnumber=23982](http://ieeexplore.ieee.org/ielx5/26/23982/01096090.pdf?tp=&arnumber=1096090&isnumber=23982)]
- [4] Peter Norvig. *English Letter Frequency Counts: Mayzner Revisited*, 2013. Modern version of the letter frequency analysis by Mayzner and Tresselt (1965)⁵, based on data from the Google books N-gram corpus (Michel et al, 2011)⁶. [[HTML: http://norvig.com/mayzner.html](http://norvig.com/mayzner.html)]
- [5] Mark S. Mayzner; Margaret Elizabeth Tresselt. *Tables of single-letter and digram frequency counts for various word-length and letter-position combinations*, 1965. In Psychonomic Monograph Supplements. Psychonomic Press.

- [6] Jean-Baptiste Michel; Yuan Kui Shen; Aviva Presser Aiden; Adrian Veres; Matthew K. Gray; The Google Books Team; Joseph P. Pickett; Dale Hoiberg; Dan Clancy; Peter Norvig; Jon Orwant; Steven Pinker; Martin A. Nowak; Erez Lieberman Aiden. *Quantitative Analysis of Culture Using Millions of Digitized Books*, 2011. In Science, Vol. 331, No. 6014, pp. 176-182. American Association for the Advancement of Science. ISSN 0036-8075. [[PDF: http://www.sciencemag.org/content/331/6014/176.full.pdf](http://www.sciencemag.org/content/331/6014/176.full.pdf)]
- [7] Terry A. Welch. *A Technique for High-Performance Data Compression*, 1984-06. In IEEE Computer, Vol. 17, No. 6, pp. 8-16. IEEE Computer Society. ISSN 0018-9162. [[download: http://ieeexplore.ieee.org/ielx5/2/34743/01659158.pdf?tp=&arnumber=1659158&isnumber=34743](http://ieeexplore.ieee.org/ielx5/2/34743/01659158.pdf?tp=&arnumber=1659158&isnumber=34743)]
- [8] Jacob Ziv; Abraham Lempel. *A Universal Algorithm for Sequential Data Compression*, 1977-03. In IEEE Transactions on Information Theory, Vol. IT-23, No. 3, pp. 337-343. IEEE. ISSN 0018-9448. [[PDF: http://www.cs.duke.edu/courses/spring03/cps296.5/papers/ziv_lempel_1977_universal_algorithm.pdf](http://www.cs.duke.edu/courses/spring03/cps296.5/papers/ziv_lempel_1977_universal_algorithm.pdf)]
- [9] Phil Katz. *APPNOTE.TXT - .ZIP File Format Specification*, 1989. PKWARE Inc. (Milwaukee, Wisconsin, USA). Distributed by PKWARE Inc. (Milwaukee, Wisconsin, USA).
- [10] Peter Deutsch. *DEFLATE Compressed Data Format Specification version 1.3*, 1996-05. Request for Comments, No. 1951. Internet Engineering Task Force. IETF. RFC 1951 (Informational). [[TXT: http://www.rfc-editor.org/rfc/rfc1951.txt](http://www.rfc-editor.org/rfc/rfc1951.txt)]
- [11] Michael Burrows; David John Wheeler. *A block-sorting lossless data compression algorithm*, 1994-05-10. Technical Report, No. SRC Research Report 124. Digital Equipment Corporation, Palo Alto, California. [[PDF: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.141.5254&rep=rep1&type=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.141.5254&rep=rep1&type=pdf)]
- [12] Julian Seward. *bzip2, version 0.1*, 1997-08. A compressor based on the block-sorting transform of Burrows and Wheeler (1994)¹¹. Source code, first published version.
- [13] Igor Pavlov. *7-Zip, Version 3.13*, 2003-12-11. Source Code, first published version. [[download: http://downloads.sourceforge.net/project/sevenzip/7-Zip/3.13/7z313.tar.bz2](http://downloads.sourceforge.net/project/sevenzip/7-Zip/3.13/7z313.tar.bz2)]
- [14] Alistair Moffat. *Implementing the PPM data compression scheme*, 1990-11. In IEEE Transactions on Communications, Vol. 38, pp. 1917-1921. ISSN 0090-6778. [[download: http://ieeexplore.ieee.org/ielx1/26/2236/00061469.pdf?tp=&arnumber=61469&isnumber=2236](http://ieeexplore.ieee.org/ielx1/26/2236/00061469.pdf?tp=&arnumber=61469&isnumber=2236)]
- [15] Paul Glor Howard; Jeffrey Scott Vitter. *Practical Implementations of Arithmetic Coding*, 1991-07. Technical Report, No. CS-91-45. Department of Computer Science, Brown University, Providence, Rhode Island, USA. [[PDF: ftp://ftp.cs.brown.edu/pub/techreports/91/cs91-45.pdf](ftp://ftp.cs.brown.edu/pub/techreports/91/cs91-45.pdf)]
- [16] John Gerald Cleary; William John Teahan; Ian Hugh Witten. *Unbounded length contexts for PPM*, 1995. In Proceedings of the Data Compression Conference, pp. 52-61, ed. James A. Storer; Martin Cohn. IEEE Computer Society. ISSN 1068-0314. ISBN 978-0-8186-7012-1. [[PDF: http://citeseer.ist.psu.edu/viewdoc/download?doi=10.1.1.14.1356&rep=rep1&type=pdf](http://citeseer.ist.psu.edu/viewdoc/download?doi=10.1.1.14.1356&rep=rep1&type=pdf)]
- [17] Suzanne Bunton. *Semantically Motivated Improvements for PPM Variants*, 1997. In The Computer Journal, Vol. 40, No. 2, pp. 76-93. Department of Computer Science and Engineering, University of Washington. Oxford University Press, Box 352350, Seattle, WA 98195-2350, USA. ISSN 0010-4620. [[PDF: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.2447&rep=rep1&type=pdf](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.103.2447&rep=rep1&type=pdf)]
- [18] Frank Wood; Cédric Archambeau; Jan Gasthaus; Lancelot James; Yee Whye Teh. *A stochastic memoizer for sequence data*, 2009-06. In ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning, ACM International Conference Proceeding Series, Vol. 382, pp. 1129-1136, ed. Léon Bottou; Michael L. Littman. ISBN 978-1-60558-516-1. [[PDF: http://eprints.pascal-network.org/archive/00005872/01/icml_fw09.pdf](http://eprints.pascal-network.org/archive/00005872/01/icml_fw09.pdf)]
- [19] Frank Wood; Jan Gasthaus; Cédric Archambeau; Lancelot James; Yee Whye Teh. *The Sequence Memoizer*, 2011. In Communications of the ACM, Vol. 52, No. 2, pp. 91-98. ACM Press, New York, NY, USA. ISSN 0001-0782. [[PDF: http://www.gatsby.ucl.ac.uk/~ywteh/research/compling/WooGasArc2011a.pdf](http://www.gatsby.ucl.ac.uk/~ywteh/research/compling/WooGasArc2011a.pdf)]

- [20] Jan Gasthaus; Yee Whye Teh. *Improvements to the Sequence Memoizer*, 2010. In Advances in Neural Information Processing Systems 23, pp. 685-693, ed. J. Lafferty; C. K. I. Williams; J. Shawe-Taylor; R. Zemel; A. Culotta. ISBN 978-1-61782-380-0. [[PDF: http://books.nips.cc/papers/files/nips23/NIPS2010_0967.pdf](http://books.nips.cc/papers/files/nips23/NIPS2010_0967.pdf)]
- [21] Nicholas Bartlett; Frank Wood. *Deplump for Streaming Data*, 2011. In Proceedings of the Data Compression Conference, pp. 363-372, ed. James A. Storer; Michael W. Marcellin. IEEE Computer Society. ISSN 1068-0314. ISBN 978-1-61284-279-0. [[PDF: http://www.stat.columbia.edu/~fwood/Papers/Bartlett-DCC-2011.pdf](http://www.stat.columbia.edu/~fwood/Papers/Bartlett-DCC-2011.pdf)]
- [22] Дмитрий А. Шкарин. *Повышение Эффективности Алгоритма PPM*, 2001. In Проблемы Передачи Информации, Vol. 37, No. 3, pp. 44-54. Наука. ISSN 0555-2923. [[PDF: http://www.mathnet.ru/links/fc3f3655dedbe86cddba9d29b9506c63/ppi526.pdf](http://www.mathnet.ru/links/fc3f3655dedbe86cddba9d29b9506c63/ppi526.pdf)]
English translation²³.
- [23] Dmitry A. Shkarin. *Improving the efficiency of the PPM algorithm*, 2001. In Problems of Information Transmission, Vol. 37, No. 3, pp. 226-235. Springer. ISSN 1608-3253. [[PDF: http://www.springerlink.com/content/k7662t777g515w41/fulltext.pdf](http://www.springerlink.com/content/k7662t777g515w41/fulltext.pdf)]
Russian original²².
- [24] Matthew Vincent Mahoney. *Fast Text Compression with Neural Networks*, 2000. In Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2000), pp. 230-234, ed. James N. Etheredge; Bill Z. Manaris. AAAI Press. ISBN 978-1-57735-113-9. [[PDF: http://mattmahoney.net/dc/mmahoney00.pdf](http://mattmahoney.net/dc/mmahoney00.pdf)]
- [25] Matthew Vincent Mahoney. *The PAQ1 data compression program*, 2002. Department of Computer Science, Florida Institute of Technology, Melbourne, FL, USA.. [[PDF: http://cs.fit.edu/~mmahoney/compression/paq1.pdf](http://cs.fit.edu/~mmahoney/compression/paq1.pdf)]
- [26] Matthew Vincent Mahoney. *Adaptive Weighing of Context Models for Lossless Data Compression*, 2005. Technical Report, No. CS-2005-16. Department of Computer Science, Florida Institute of Technology, Melbourne, FL, USA.. [[download: https://repository.lib.fit.edu/bitstream/handle/11141/154/cs-2005-16.pdf?sequence=1&isAllowed=y](https://repository.lib.fit.edu/bitstream/handle/11141/154/cs-2005-16.pdf?sequence=1&isAllowed=y)]

How to cite this presentation

```
@misc{steinruecken2015d,
  title = {PPM revisited},
  author = {Christian Steinruecken},
  year = {2015},
  month = apr,
  url = {http://cstein.kings.cam.ac.uk/~chris/talks/ppm-dp.html},
  howpublished = {Slide presentation. Data Compression Conference 2015, Snowbird, Utah.},
}
```