Sparse Adaptive Dirichlet-Multinomial-like Processes

Marcus Hutter

Research School of Computer Science
Australian National University
Canberra, ACT, 0200, Australia
http://www.hutter1.net/

May 2013

Abstract

Online estimation and modelling of i.i.d. data for short sequences over large or complex "alphabets" is a ubiquitous (sub)problem in machine learning, information theory, data compression, statistical language processing, and document analysis. The Dirichlet-Multinomial distribution (also called Polya urn scheme) and extensions thereof are widely applied for online i.i.d. estimation. Good a-priori choices for the parameters in this regime are difficult to obtain though. I derive an optimal adaptive choice for the main parameter via tight, data-dependent redundancy bounds for a related model. The 1-line recommendation is to set the 'total mass' = 'precision' = 'concentration' parameter to $m/[2\ln\frac{n+1}{m}]$, where n is the (past) sample size and m the number of different symbols observed (so far). The resulting estimator (i) is simple, (ii) online, (iii) fast, (iv) performs well for all m, small, middle and large, (v) is independent of the base alphabet size, (vi) non-occurring symbols induce no redundancy, (vii) the constant sequence has constant redundancy, (viii) symbols that appear only finitely often have bounded/constant contribution to the redundancy, (ix) is competitive with (slow) Bayesian mixing over all sub-alphabets.

Contents

Keywords

sparse coding; adaptive parameters; Dirichlet-Multinomial; Polya urn; data-dependent redundancy bound; small/large alphabet; data compression.

1 Introduction

The problem of estimating or modelling the probability distribution of data sequences sampled from an unknown source is central in machine learning [Bis06], information theory [CT06], and data compression [Mah12]. I consider the case where the data items are complex and/or are drawn from a large space. Many approaches to language modelling and document analysis [MS99] fall into this regime, where data items are words. Typical documents comprise a small fraction of the available 100'000+ English words, and words have different length/complexity/frequency.

Online estimation of i.i.d. data. More formally, I consider i.i.d. data with base alphabet $\mathcal{X}$ much larger than the sequence length, which implies that only a small fraction of symbols (which in case of text are words) appear in the sequence. I focus on online algorithms that at any time can predict the probability of the next symbol given only the past sequence and without knowing the actually used alphabet $\mathcal{A}$ and/or symbol occurrence frequencies in advance.

While real-word data like text are often not i.i.d, i.i.d. estimators are often a key component of more sophisticated models. For instance, in n-gram models, the subsequence of words that have the same length-n context is (assumed) i.i.d. Since these subsequences can be very short, good i.i.d. estimators for short sequences and huge alphabet are even more important. The same holds for variable-order models like large-alphabet context tree weighting [TSW93], and in addition, the employed i.i.d. estimators need to be online.

Performance measures. Performance can be measured in many different ways: code length [CT06], perplexity [MS99], redundancy [Wal05], regret [Grü07], and others. The most wide-spread (across disciplines) performance measures are transformations of the (estimated) data likelihood(s). If $Q(x_{1:n})$ is the estimated probability of sequence $x_{1:n} \equiv x_1...x_n$, then $\log 1/Q(x_{1:n})$ is the optimal code length and $Q(x_{1:n})^{1/n}$ the perplexity of $x_{1:n}$. If P is some reference measure, then $\log 1/Q - \log 1/P$ is the

redundancy of Q relative to P. For log-loss, this is also its regret, though many variations are used. Many other performance measures can be upper bounded by (expected) code length [Hut03]. I therefore concentrate on $-$log-likelihood = code length and redundancy.

Dirichlet-multinomial and parameter choice. The Dirichlet-multinomial distribution is defined as $\text{DirM}(x_{n+1}=i|x_{1:n}) = \frac{n_i+\alpha_i}{n+\alpha_+}$, which can be motivated in many ways, e.g. by the Polya urn scheme or as below. This process and extensions thereof like the Pitman-Yor process are widely studied and applied [BH10], in particular for language processing and document analysis. Theoretically motivated choices for the Dirichlet parameters α_i are $\alpha_i = 1/2$ for the Krichevsky-Trofimov (KT) estimator [KT81] and Jeffreys/Bernardo/MDL/MML prior [Jef46, Jef61, Ber79, Grü07, Wal05], $\alpha_i=0$ for Frequentist and Haldane's prior [Hal48], $\alpha_i=1$ for the uniform/indifference/Bayes/Laplace prior [Bay63, Lap12], and $\alpha_i=1/|\mathcal{X}|$ for Perks' prior [Per47]. They are all problematic for large base alphabet $\mathcal{X}$, so α_+ is sometimes optimized or sampled experimentally or averaged with a hyper-prior. The following table summarizes these choices:

Dirichlet	Laplace	KT&others	Perks	Haldane	Hutter
$\alpha_i = \frac{\alpha_+}{\lvert\mathcal{X}\rvert}$	1	$\frac{1}{2}$	$\frac{1}{\lvert\mathcal{X}\rvert}$	0	$\frac{m}{2\lvert\mathcal{X}\rvert \ln \frac{n+1}{m}}$

(1)

The last column is a glimpse of the results in this paper, where m is the number of different symbols that appear in $x_{1:n}$. For continuous spaces $\mathcal{X}$, the Dirichlet process is usually parameterized by a base distribution $H()$ and a critical concentration parameter $\beta\widehat{=}\alpha_+$.

Main contribution. In this paper I introduce an estimator S [Eq.(2)], which essentially estimates the probability of the next symbol by its past relative frequency, but reserves a small (or large!) "escape" probability to new symbols that have not appeared so far. Such escape mechanisms are well-known and used in data compression such as prediction by partial match (PPM) [CW84, Mah12]. This is (somewhat) different from how the Dirichlet-multinomial regularizes zero frequency with $\alpha_i>0$ or $\beta>0$.

The main contribution is to derive an "optimal" escape parameter β^* [Eq.(16) offline and Eq.(21) online]. The key to improve upon existing estimators like the minimax optimal KT estimator is to consider data-dependent redundancy bounds, rather than expected or worst-case redundancy, and find its minimizing β. While the KT estimator and many of its companions have $\frac{1}{2}\log n$ redundancy per symbol in $\mathcal{X}$, whether the symbol occurs in the sequence or not, our new estimator S^{β^*} suffers zero redundancy for non-occurring symbols, and essentially only $\frac{1}{2}\log n_i + O(1)$ for symbols i appearing n_i times. This is never much worse and often significantly better than KT. This also leads to an "optimal" variable Dirichlet parameter $\vec{\beta}^*$. While knowing $\vec{\beta}^*$ is practically useful, the derived redundancy bounds themselves are of theoretical interest.

Contents. After establishing notation in Section 2, I motivate and state my primary model S^β in Section 3. I derive exact expressions and upper and lower bounds for the redundancy of S^β for general constant β in Section 4, and show how they improve upon the minimax redundancy. I approximately minimize the redundancy w.r.t. β in Section 5. There are various regimes for the optimal β^* and the used alphabet size $|\mathcal{A}|$, even with negative redundancy. To convert this into an online model, I make β^* time-dependent in Section 6, causing very little extra redundancy. In Section 7 I theoretically compare my models to the Dirichlet-multinomial distribution, and Bayesian sub-alphabet weighting. Section 8 concludes.

Proofs of the lower and two upper bounds can be found in Appendices B, D, and E, a derivation of β^* in Appendix C with improvements in Appendix F, details of Bayesian subset-alphabet weighting in Appendix G, algorithmic considerations in Appendix H, and an experimental evaluation in Appendix I. Used properties of the (di)Gamma functions can be found in Appendix A, and a list of used notation in Appendix J.

2 Preliminaries

All global notation is introduced in this section and summarized in Appendix J.

Base alphabet ($\mathcal{X}$, D). Let $\mathcal{X}$ be the *base* alphabet of size $D=|\mathcal{X}|$ from which a sequence of symbols is drawn. If not otherwise mentioned, I assume $\mathcal{X}$ to be finite. I have a large base alphabet in mind, but this is not a technical requirement. The alphabet could literally consist of e.g. ASCII symbols, could be the set of (over 100'000) English words, or just bits $\{0,1\}$. Indeed, even finiteness of $\mathcal{X}$ is nowhere crucially used and all results generalize easily to countable and even continuous $\mathcal{X}$ as we will see.

Total sequence (n, $x_{1:n}$, n_i). I consider sequences $x_{1:n}\equiv(x_1,...,x_n)\in\mathcal{X}^n$ of length n drawn from $\mathcal{X}$. Let n_i be the number of times, i appears in $x_{1:n}$. I have in mind that the sequences are sampled independent and identically distributed (i.i.d), but I actually never use this assumption. All results in this paper hold for any individual fixed sequence $x_{1:n}$, and only depend on the order statistics $\boldsymbol{n}=(n_i)_{i\in\mathcal{X}}$. The crucial parameters are n, D, the number m of non-zero counts, and model parameter β introduced later, which induces several different regimes, second by the counts n_i.

Used alphabet (m, $\mathcal{A}$, i,j,k, ν, $\bar{\nu}$). Only a subset of symbols $\mathcal{A}:=\{x_1,...,x_n\}\subseteq\mathcal{X}$ may actually appear in a sequence $x_{1:n}$. Our model is primarily motivated for the regime where the number $m=|\mathcal{A}|$ of *used* symbols is much smaller than $D=|\mathcal{X}|$, as e.g. any English text uses only a small fraction of all possible words. It turns out that our model can be tuned to actually perform very well for all possible $1\leq m\leq\min\{n,D\}$: constant sequences ($m=1$), every symbol appearing only once ($m=n$), and all available symbols appear ($m=D$). Indices i,j,k are understood to range respectively over symbols in $\mathcal{X}$, $\mathcal{A}$, and $\mathcal{X}\backslash\mathcal{A}$. Without loss of generality I can

assume $i \in \mathcal{X} = \{1,...,D\}$, $j \in \mathcal{A} = \{1,...,m\}$, and $k \in \mathcal{X} \setminus \mathcal{A} = \{m+1,...,n\}$. I also use $\bar{\nu} := n/m$ for the average multiplicity of symbols in $x_{1:n}$, and $\nu := m/n$ is its inverse.

Current sequence and observed alphabet (t, $x_{1:t}$, $\mathcal{A}_t$, m_t, x_{t+1}, n_i^t, $\mathcal{N}ew$, $\mathcal{O}ld$). Let t be the current time ranging from 0 to $n-1$, with $x_{1:t}$, $\mathcal{A}_t := \{x_1,...,x_t\}$ and $m_t = |\mathcal{A}_t|$ being respectively, the sequence, symbols, and number of different symbols observed so far, and as usual $x_{1:0} = \epsilon$ is the empty string and $\mathcal{A}_0 = \{\}$ the empty set. The next symbol to be predicted or coded is $x_{t+1} = i$. Either x_{t+1} is a new symbol or an "old" symbol. Let $\mathcal{N}ew := \{t = 0...n-1 : x_{t+1} \notin \mathcal{A}_t\}$ and $\mathcal{O}ld := \{t = 0...n-1 : x_{t+1} \in \mathcal{A}_t\}$ be the sets of times for which the next symbol is new/old. Note that $|\mathcal{N}ew| = |\mathcal{A}| = m$. Finally, let n_i^t be the number of times, i appears in $x_{1:t}$. Note that most inroduced quantities $*_t$ depend on $x_{1:t}$, but since I consider an (arbitray but) fixed sequence $x_{1:n}$ it is safe to suppress this dependence in the notation.

Probability and exchangeability and logarithms (P, Q, $P^{\text{param}}_{\text{name}}$, ln). P and Q will denote generic probability distributions over sequences, and $P^{\text{param}}_{\text{name}}$ specific parameterized and named ones. For instance, $P^{\boldsymbol{\theta}}_{iid}$ denotes the model in which symbols are i.i.d. with $P(x_t = i) = \theta_i$. Our primary prediction/compression models defined below are S, S^{β^*}, and $S^{\tilde{\beta}^*}$. A distribution $P(x_{1:n})$ is called exchangeable if it is independent of the order of the symbols in a sequence $x_{1:n}$. Many distributions have this desirable property [Fin74]. Since the natural logarithm is mathematically more convenient, I express all results in 'nits' rather than bits. Conversion to bits is trivial by dividing results by ln2.

3 The Main Model

I am now ready to motivate and formally state our primary model.

Derivation of my main model. My main model is defined via predictive distributions $S(x_{t+1}|x_{1:t})$ for $t = 0...n-1$. If i has appeared n_i^t times in $x_{1:t}$, it is natural to use the past relative frequency n_i^t/t as the predictive probability that the next symbol x_{t+1} is i. The problems with this are well-known and obvious: It assigns probability zero and hence infinite log-loss or code length to any symbol that has not yet been observed. This problem can be solved by reserving some small (or not so small) "escape" probability α_t that the next symbol x_{t+1} is new, taken from n_i^t/t by lowering it to $(1-\alpha_t)n_i^t/t$. I have to somehow distribute the probability α_t among the new symbols $x_{t+1} \in \mathcal{X} \setminus \mathcal{A}_t$. The simplest choice would be uniform. More generally assign probability $\alpha_t w_k^t$ to $k = x_{t+1} \in \mathcal{X} \setminus \mathcal{A}_t$ with $\sum_{k \in \mathcal{X} \setminus \mathcal{A}_t} w_k^t \leq 1$ and $w_k^t > 0$.

One can show that the ansatz above for time-independent weights leads to an exchangeable distribution if and only if $\alpha_t = \beta/(t+\beta)$ for some constant $\beta \geq 0$.

Main model. This motivates our main model

$$S(x_{t+1}=i|x_{1:t}) \;:=\; \begin{cases} \dfrac{n_i^t}{t+\beta} & \text{for} \quad n_i^t>0 \\ \dfrac{\beta w_i^t}{t+\beta} & \text{for} \quad n_i^t=0 \end{cases} \tag{2}$$

for $t=0...n-1$. Note that $S(x_1=i)=w_i^0$ is independent of $\beta>0$. The case conditions can also be written as $[n^t_{x_{t+1}}>0] \equiv [x_{t+1}\in\mathcal{A}_t] \equiv [t\in\mathcal{O}ld]$ and $[n^t_{x_{t+1}}=0] \equiv [x_{t+1}\notin \mathcal{A}_t] \equiv [t\in\mathcal{N}ew]$. Other motivations and relations to other estimators are given in Section 7.

Sub-probability. In general, $\sum_{i\in\mathcal{X}} S(x_{t+1}=i|x_{1:t})\leq 1$, but not necessarily $=1$. Such sub-probabilities are benign extensions for many purposes including ours. It is always possible to increase sub-probabilities to proper probabilities. For S we could replace w_i^t by $w_i^t/\sum_{k\in\mathcal{X}\setminus\mathcal{A}_t} w_k^t$ as long as $\mathcal{X}\setminus\mathcal{A}_t$ is not empty, and replace β by 0 if ever all base symbols ($m_t=D$) have appeared. Note that unless $m_t=D$, we have to assume $\beta>0$ to avoid the problems of frequentist estimation.

Sequence probability. The probability our model assigns to sequence $x_{1:n}$ is

$$S(x_{1:n}) \;=\; \prod_{t=0}^{n-1} S(x_{t+1}|x_{1:t}) \;=\; \prod_{t=0}^{n-1} \frac{1}{t+\beta} \prod_{t\in\mathcal{O}ld} n^t_{x_{t+1}} \prod_{t\in\mathcal{N}ew} \beta w^t_{x_{t+1}} \tag{3}$$

$$\;=\; \beta^{|\mathcal{A}|}\frac{\Gamma(\beta)}{\Gamma(n+\beta)} \prod_{t\in\mathcal{N}ew} w^t_{x_{t+1}} \prod_{j\in\mathcal{A}} \Gamma(n_j) \tag{4}$$

where Γ is the Gamma function. The symbol count n_j^t increases by 1 for each occurrence of j in the sequence. Therefore $\prod_{t\in\mathcal{O}ld: x_{t+1}=j} n_j^t = 1\cdot...\cdot(n_j-1)=\Gamma(n_j)$, which establishes the second line.

4 Redundancy of S^β for General β

In this section I motivate and define the concepts of redundancy and (log-loss) regret and present an exact expression for the redundancy of S^β for general constant β. Upper and lower bounds are easily derived by bounding the involved Gamma functions. Finally I discuss the β-independent terms in the bound, and how they improve upon the minimax redundancy.

Code length and redundancy/regret. If a data sequence is sampled from some distribution P, then a lower bound on the expected code length is the entropy $H(P)$ of the source P, which can only be achieved by an encoder which encodes sequences $x_{1:n}$ in $-\ln P(x_{1:n})$ nits [Sha48].

Arithmetic encoding [Ris76, WNC87] can (efficiently and online) achieve this lower bound within 2 bits. It is therefore appropriate to call

$$\mathrm{CL}_P(x_{1:n}) := \ln 1/P(x_{1:n})$$

the (optimal) code length of $x_{1:n}$ (in nits w.r.t. P). Arithmetic coding also works for sub-probabilities.

Usually, P is unknown, and one aims at compressors getting close to CL_P for all P that might be "true" and/or for all P for which it is feasible to do so. Let $\mathcal{M}=\{P\}$ be such a class of interest; then $\min_{P\in\mathcal{M}}\mathrm{CL}_P(x_{1:n})$ is an (infeasible) lower bound on the best possible coding if $x_{1:n}$ is sampled from *some* $P\in\mathcal{M}$.

Most modern compressors are themselves based on a (predictive) distribution Q used together with arithmetic coding [Mah12]. This motivates the concept of *redundancy* or *regret* R as a performance measure for Q, which I define as the difference in code length between the data coded with predictor Q and the infeasible optimal code length in hindsight:

$$R_Q(x_{1:n}) \;:=\; \mathrm{CL}_Q(x_{1:n}) - \min_{P\in\mathcal{M}} \mathrm{CL}_P(x_{1:n}) \;=\; \ln \frac{\max_{P\in\mathcal{M}} P(x_{1:n})}{Q(x_{1:n})} \tag{5}$$

For comparing the code lengths of different Q, any quantity from which CL_Q can easily be recovered could be studied: log-loss regret $\mathrm{CL}_Q-\mathrm{CL}_P$ or redundancy $\mathrm{CL}_Q-H(P)$ where P is the true distribution of entropy $H(P)$, or CL_Q-c for any other "constant" c independent of Q, and of course code length CL_Q itself. The redundancy R_Q w.r.t. class $\mathcal{M}$ defined above ($c=\min_{P\in\mathcal{M}}\mathrm{CL}_P(x_{1:n})$) is just often and also here the most convenient choice. Upper and lower bounds on redundancies will be denoted by $\overline{R}$ and $\underline{R}$.

I.i.d. reference class. As reference class $\mathcal{M}$ I choose the class of i.i.d. distributions with symbol $i\in\mathcal{X}$ having probability $\theta_i\in[0;1]$.

$$P^{\boldsymbol{\theta}}_{iid}(x_{1:n}) \;:=\; \theta_{x_1}\cdot\ldots\cdot\theta_{x_n} \;=\; \prod_{i\in\mathcal{X}}\theta_i^{n_i} \;=\; \prod_{j\in\mathcal{A}}\theta_j^{n_j}$$

The maximum is attained at $\theta_i=\hat{\theta}_i:=n_i/n$; therefore

$$P^{\hat{\boldsymbol{\theta}}}_{iid}(x_{1:n}) \;=\; \max_{\boldsymbol{\theta}} P^{\boldsymbol{\theta}}_{iid}(x_{1:n}) \;=\; n^{-n}\prod_{j\in\mathcal{A}} n_j^{n_j} \tag{6}$$

Redundancy of S. Subtracting the logarithm of (4) from the logarithm of (6) and using abbreviation $\mathrm{CL}_w(\mathcal{A}):=\sum_{t\in\mathcal{N}ew}\ln(1/w^t_{x_{t+1}})$ discussed below, one can represent the redundancy of S as follows:

Proposition 1 (Redundancy of S for constant β) *For any constant $\beta>0$, the redundancy of S^β relative to the i.i.d. class $\mathcal{M}=\{P^{\boldsymbol{\theta}}_{iid}\}$ can be represented exactly and bounded as follows:*

$$\underline{R}^\beta_S(x_{1:n}) \;\leq\; R^\beta_S(x_{1:n}) \;\leq\; \overline{R}^\beta_S(x_{1:n}), \quad \textit{where}$$

$$R^\beta_S \;=\; CL_w(\mathcal{A}) - m\ln\beta + \sum_{j\in\mathcal{A}}\ln\frac{n_j^{n_j}}{\Gamma(n_j)} + \ln\frac{\Gamma(n+\beta)}{n^n\,\Gamma(\beta)} \tag{7}$$

$$\overline{R}^\beta_S \;:=\; CL_w(\mathcal{A}) - m\ln\beta + \sum_{j\in\mathcal{A}}\tfrac{1}{2}\ln\frac{n_j}{2\pi} + n\ln(1+\frac{\beta}{n}) + (\beta-\tfrac{1}{2})\ln(\frac{n}{\beta}+1) + 0.082 \tag{8}$$

$$\underline{R}^\beta_S \;:=\; \overline{R}^\beta_S - 0.082(m+2) \tag{9}$$

where $\mathcal{A}\subseteq\mathcal{X}$ are the m (a-priori unknown) symbols appearing in $x_{1:n}\in\mathcal{X}^n$. The lower bound only holds for $\beta\geq 1$. The 0.082 is actually $1-\ln\sqrt{2\pi}$.

The exact expression follows easily by rearranging terms in (4) and (6). The bounds follow from this by inserting the upper and lower bounds (27) on the Gamma function and collecting/cancelling matching terms. As can be seen, the upper and lower bounds only differ by $0.082(m+2)$, hence are quite tight for small m, but loose for large m.

In the following paragraphs I discuss the two β-independent terms. The β-dependent terms will be discussed in the next section. Note that the following interpretation of (7) only refers to code length. The actual way how arithmetic coding works is very different from this "naive" interpretation of the origin of the different terms in (7).

Code length of used alphabet $\mathcal{A}$. The first term in the redundancy (7)

$$\mathrm{CL}_w(\mathcal{A}) \;:=\; \sum_{t\in\mathcal{N}ew} \ln(1/w^t_{x_{t+1}}) \tag{10}$$

can be interpreted as follows: Whenever we see a new symbol $x_{t+1}\notin\mathcal{A}_t$, we need to code the symbol itself. This can be done in $\ln(1/w^t_{x_{t+1}})$ nits, which together leads to code length (10) for the used alphabet $\mathcal{A}$.

A natural choice for the new symbol weights is the uniform distribution $w^t_i=1/D$ with $\mathrm{CL}_w(\mathcal{A})=m\ln D$. Since at time t there are only $D-m_t$ new symbols left, we could use normalized uniform weights $w^t_k=1/(D-m_t)$ with smaller

$$\mathrm{CL}_w(\mathcal{A}) \;=\; \ln(D)+...+\ln(D-m+1) \;=\; \ln[D!/(D-m)!] \tag{11}$$

For large, structured, and/or infinite alphabet, a more natural choice is $w^t_i = \exp(-\mathrm{CL}(i))$ with

$$\mathrm{CL}_w(\mathcal{A}) \;=\; \sum_{t\in\mathcal{N}ew}\mathrm{CL}(x_{t+1}) \;=\; \sum_{j\in\mathcal{A}}\mathrm{CL}(j) \tag{12}$$

were new symbols j are *somehow* coded (prefix-free) in $\mathrm{CL}(j)$ nits. For intstance if $\mathcal{X}$ consists of English words, each word i with ℓ letters could be represented as a byte-string of length ℓ plus a 0 terminating byte, hence $\mathrm{CL}(i)=8\ell+8$. Choice (12) is interesting since it makes the redundancy completely independent of the size of the base alphabet, and hence leads to finite redundancy even for infinite alphabet $\mathcal{X}$.

For all examples of weights above, $\mathrm{CL}_w(\mathcal{A})$ is independent of order and timing of new symbols, which justifies suppressing the dependence on $\mathcal{N}ew$. This holds more generally for all w^t_i of the form $w^t_i=u(i)v(m_t)$

$$\mathrm{CL}_w(\mathcal{A}) \;=\; \sum_{j\in\mathcal{A}}\ln\frac{1}{u(j)}+\sum_{m'=0}^{m-1}\ln\frac{1}{v(m')} \tag{13}$$

For ease of discussion, I will only consider weights of this form, and indeed mostly the normalized uniform (11) and code-length based (12) ones. Then also R_S^β only depends on the counts n_i but not on the symbol order, as intended.

Code length of relative frequencies n_i/n. Oracle $P_{iid}^{\hat{\boldsymbol{\theta}}}$ predicts symbol j with empirical frequency n_j/n, so j can be coded in $\ln(n/n_j)$ nits. I label an estimator ORACLE if it relies on extra information, here, knowing the empirical symbol frequencies in advance. Technically, $P_{iid}^{\hat{\boldsymbol{\theta}}(x_{1:n})}(x_{1:n})$ is an inadmissible super-probability. To get a feasible (but offline) predictor one needs to encode the counts n_i in advance. Arithmetic coding w.r.t. S^β does not work like that but imagine it did. The $\ln(n/n_j)$ terms would cancel in the redundancy leaving a code length for all n_i. $\mathrm{CL}(\mathcal{A})$ tells us which n_i are zero, so only n_j for $j\in\mathcal{A}$ need to be coded, which can be done in $\ln n$ nits per $j\in\mathcal{A}$, and the upper bound (8) suggests possibly even in $\frac{1}{2}\ln(n_j/2\pi)$ nits.

Improvement over minimax redundancy. It is well known that the minimax redundancy of i.i.d. sources is $\frac{1}{2}\ln n+O(1)$ per base alphabet symbol [Ris84, Wal05]. My model improves upon this in two significant ways. Consider the asymptotics $n\to\infty$ in (8). First, all symbols k that do not appear in $x_{1:n}$ induce zero redundancy. Second, each symbol j that appears only finitely often, induces finite bounded redundancy $\mathrm{CL}(j)+\frac{1}{2}\ln\frac{n_j}{2\pi}$ plus β-terms discussed later. Only symbols appearing with non-vanishing frequency $n_i/n\not\to 0$ have asymptotic redundancy $\frac{1}{2}\ln n+O(1)$. This improvement (a) is possible (only) for specific choices of β such that the β-terms are small and (b) was possible by refraining from deriving a uniform minimax redundancy over all sequences, but one which depends on the symbol counts.

$\boldsymbol{\beta}$-independent lower redundancy bound. In Appendix B I derive a β-independent lower bound on the redundancy that cannot be beaten, whatever β is chosen. The following lower bound has the same structure as the upper bounds I derive later, so the terms will be discussed there.

Theorem 2 (β-independent lower redundancy bound) *For any constant $\beta>0$, the redundancy of S^β is lower bounded uniformly in β by:*

$$R_S^\beta(x_{1:n}) \;\geq\; CL_w(\mathcal{A}) - m\ln m + \sum_{j\in\mathcal{A}} \tfrac{1}{2}\ln\frac{n_j}{2\pi} - \tfrac{1}{2}\ln n - 0.45m - 0.43 \qquad (14)$$

5 Redundancy for Approximate Optimal $\boldsymbol{\beta^*}$

I am now in a position to approximately minimize the redundancy of S^β w.r.t. β. Even when only considering asymptotics $n\to\infty$, I need to distinguish six different regimes for β^* depending on how m scales with n. I discuss the more interesting regimes, in particular the unusual situation of negative redundancy.

Optimal constant $\boldsymbol{\beta}$. I now optimize S^β w.r.t. to β. The redundancy R_S^β is minimized for

$$0 \stackrel{!}{=} \frac{\partial R_S^\beta}{\partial \beta} = -\frac{m}{\beta} + \Psi(n+\beta) - \Psi(\beta) \tag{15}$$

where $\Psi(x) := \mathrm{d}\ln\Gamma(x)/\mathrm{d}x$ is the diGamma function. Neither this equation nor $\partial \overline{R}_S^\beta/\partial\beta = 0$ have closed-form solutions, and even asymptotic approximations are a nuisance. It seems natural to derive expressions for $n\to\infty$ and/or $m\to\infty$, but since β is inside the diGamma functions it turns out that considering β-limits leads to fewer cases. Still one has to separate the regimes $\beta\to\infty$, $\beta\to c\lessgtr^\infty_0$, $\beta\to 0$, $\beta/n\to\infty$, $\beta/n\to c\lessgtr^\infty_0$, and $\beta/n\to 0$. I do this in Appendix C with further discussion and improvements in Appendix F and stitch together the results, leading to a surprisingly neat result:

Theorem 3 (Optimal constant β) *The β which minimizes R_S^β (7) and solves (15) is*

$$\beta^{min} = \frac{m}{c_n(\frac{m}{n})\ln\frac{n}{m}}, \quad \begin{array}{l}\textit{where } c_\infty(\nu) := \lim_{n\to\infty} c_n(\nu) \textit{ is smooth and}\\ \textit{monotone increasing from } c_\infty(0)=1 \textit{ to } c_\infty(1)=2.\end{array}$$

For $n\gg m$ we have $c_n(m/n)\approx 1$, which suggests the approximation

$$\beta^* := \frac{m}{\ln\frac{n}{m}} \tag{16}$$

This has the same asymptotics as β^{min} in all regimes of interest and turns out to lead to excellent experimental results. In practice, $c_n(m/n)$ is closer to 2, so halving β^* leads to slightly better results unless m is extremely small. This is due to a quite peculiar shape of $c_\infty(\nu)$, plotted and discussed in more detail in Appendix F. The performance difference between S^{β^*}, $S^{\beta^*/2}$, and β^{min} are very small though. I hence use β^* (16) for most of the theoretical analysis but recommend $\beta^*/2$ (1) in practice. Since no formal result in this paper explicitly uses that β^* is an approximate solution of (15), we can simply take β^* on faith value and explore its implications.

Discussion of $\boldsymbol{\beta^*}$. The value of β^* can be intuitively understood in this way: if m is much larger than $\ln n$, then we will often be coding new symbols, and therefore we should reserve more probability mass for them by making β large. If however m is much smaller than $\ln n$, coding a new symbol is a rare occurrence, so we use a small β to increase the efficiency of coding already previously seen symbols. More quantitatively, β^* (and β^{min}) scale with $n\to\infty$ for various m as follows (where $0<c<\infty$ and $0\le\alpha<1$)

m	$\to c$	$\propto \ln n$	$\propto n^\alpha$	$\propto n$	$\ge n-c$	$=n$
β^*	$\sim c/\ln n$	$\to c$	$\propto n^\alpha/\ln n$	$\propto n$	$\propto n^2$	∞

(17)

Besides the mentioned $m\lessgtr\ln n$ divide, note that if most symbols appear only once, then $\beta\propto n^2$ grows very rapidly. On the other hand β^* is never very small: $1/\ln n$ is

a lower bound, even if $m=1$. If no symbol appears twice, then $\beta^*=\infty$ is obviously the best choice. Appendix I shows that S^{β^*} works very well in all six regimes.

I also tried "minor" modifications but theory breaks down for some, and experiments for others. The only leeway, apart from replacing $c_n()$ by a constant in [1;2] I could find is adding or subtracting small constants from m and/or n in (16). This will later be used to regularize β^* for $m=n$. Note that β^* depends on the a-priori unknown n and m, so S^{β^*} is not online. This will be rectified in Section 6. In Appendix D I prove the following redundancy bound:

Theorem 4 (Redundancy of S for "optimal" constant β^*) *The redundancy of S^{β^*} with $\beta^*=m/\ln\frac{n}{m}$ is bounded by*

$$R_S^{\beta^*}(x_{1:n}) \;\leq\; CL_w(\mathcal{A}) - (m-\tfrac{1}{2})\ln m + \sum_{j\in\mathcal{A}} \tfrac{1}{2}\ln n_j - \tfrac{1}{2}\ln n + m\ln\ln\frac{en}{m} + 0.56m + 0.082 \tag{18}$$

Discussion of $R_S^{\beta^*}$. The first and third term have already been discussed. The second term is the most important one for large m. It is about $-\ln\Gamma(m)-m+1$ by (27). Therefore for uniform normalized weights (11) we get

$$\mathrm{CL}_w(\mathcal{A}) - (m-\tfrac{1}{2})\ln m \;=\; \ln\binom{D}{m} - m + \ln m + 1 \;\begin{cases} -0.082 \\ +0 \end{cases} \tag{19}$$

There are $\binom{D}{m}$ ways of choosing m symbols out of D, therefore $\ln\binom{D}{m}$ corresponds to the optimal uniform code length for the used unordered alphabet. At first, S^{β^*} seemed to be more wasteful, coding the m'th new symbol in $\ln(D-m'+1)$ nits, hence codes $\mathcal{A}$ including order in $\mathrm{CL}_w(\mathcal{A})$ nits. But through the back door by a suitable choice of β, it actually achieves the theoretically optimal uniform code length $\ln\binom{D}{m}$ for the used alphabet, plus other smaller terms. For large m, this can be significantly smaller than $\mathrm{CL}_w(\mathcal{A})$.

In the extreme case of $m=D$, we have $\ln\binom{D}{D}=0\ll D\ln D$. If also $n=m$, we have $\mathrm{CL}_w(\mathcal{A})=\ln n!$ and $n_i=1\forall i$ and hence

$$R_S^{\beta^*} \;\leq\; \ln n! - n\ln n + 0.56n + 0.082 \;\leq\; \tfrac{1}{2}\ln n - 0.44n + 1.082$$

which is negative for $n>4$. This is not a contradiction. It just says that in this case S codes better than oracle $P_{iid}^{\hat{\boldsymbol{\theta}}}=(\frac{1}{n})^n$. Indeed, if we know that every symbol appears exactly once, we can code their permutation in $\ln n!$ rather than $n\ln n$ nits. The $+0.56n$ slack is an artefact of our bound, not of S^{β^*}, and can be improved to $0.082n$. The argument generalizes to large $m<n$.

In the other extreme of a constant sequence $x_t=j\forall t$, we have $m=1$, $P_{iid}^{\hat{\boldsymbol{\theta}}}=1$, $\beta^*=1/\ln n$ and $\mathrm{CL}_{S^{\beta^*}}=R_S^{\beta^*}\to\mathrm{CL}_w(j)+1$ for $n\to\infty$, i.e. 1 nit above theoretical optimum from (7) and $R_S^{\beta^*}\leq\mathrm{CL}_w(j)+\ln\ln(en)+0.65$ from (18), i.e. asymptotically there is only $\ln\ln n$ nits slack in the bound. This argument generalizes to constant $m>1$.

The $S^{\vec{\beta}^*}$-probability of $x_{t+1}=i\in\mathcal{X}$ given $x_{1:t}\in\mathcal{X}^t$ is defined as

$$S^{\vec{\beta}^*}(x_{t+1}=i|x_{1:t}) \;:=\; \begin{cases} \dfrac{n_i^t}{t+\beta_t^*} & \text{for} \quad n_i^t>0 \\ \dfrac{\beta_t^* w_i^t}{t+\beta_t^*} & \text{for} \quad n_i^t=0 \end{cases} \tag{20}$$

$$\beta_t^* \;:=\; \frac{m_t}{\ln\frac{t+1}{m_t}}, \quad t\geq 1, \qquad 0<\beta_0^*<\infty \text{ (any)}, \qquad \vec{\beta}:=(\beta_0,\beta_1,\beta_2,...) \tag{21}$$

$$\sum_{k\in\mathcal{X}\setminus\mathcal{A}_t} w_k^t \leq 1, \qquad \text{e.g.} \qquad w_i^t \;=\; \frac{1}{D-m_t} \qquad \text{or} \qquad w_i^t \;=\; \mathrm{e}^{-\mathrm{CL}(i)}$$

$$m_t=|\mathcal{A}_t|, \qquad \mathcal{A}_t=\{x_1,...,x_t\}, \qquad n_i^t=|\{\tau\in\{1,...,t\}:x_\tau=i\}|$$

6 Redundancy for Variable $\vec{\beta}^*$

Since the optimal $\beta^*=m/\ln\frac{n}{m}$ depends on m and n, S^{β^*} cannot be used online, which defeats one of its purposes and significantly limits its application as discussed in the introduction. I rectify this problem by allowing a time-dependent β in my model, and by adapting β^* in (nearly) the most obvious way. I derive a redundancy bound for this variable $\vec{\beta}^*$ which for small m is only slightly worse than the previous one for constant β^*.

Choice of $\vec{\beta}^*$. A natural way to arrive at an online algorithm is to replace n by t and m by m_t, both known at time t and converging to n and m respectively. This leads to a time-dependent 'variable' $\beta_t=m_t/\ln\frac{t}{m_t}$. This works fine except if $m_t=t$, in which case $\beta_t=\infty$ assigns zero probability that the next symbol is an old one. This is unacceptable, since $m_t=t$ is typical for small t.

If we are at time t, we use β_t to predict x_{t+1} so should assume that the sequence has (at least) length $t+1$, which suggests $\beta_t=m_{t+1}/\ln\frac{t+1}{m_{t+1}}$. The problem here is that m_{t+1} depends on the unknown x_{t+1}, and technically S becomes an (unusable) super-probability. Since $m_{t+1}=m_t$ if x_{t+1} is old anyway, a natural choice is $\beta_t^*=m_t/\ln\frac{t+1}{m_t}$, which still has the same asymptotics (17) as β^*, except for $m_t=t$ it is finite and grows with t^2. For $t=0$ I define $S(x_1=i)=w_i^t$ or equivalently choose any $0<\beta_0^*<\infty$. For convenience I summarize the adaptive model with parameters and definitions in the box on the next page.

Note that compact representation (4) does not hold anymore: The resulting process $S^{\vec{\beta}^*}(x_{1:n})$ is no longer exchangeable, but close enough in the sense that a comparable upper bound as for β^* holds. The constants are somewhat worse, but mostly due to the crude proof (see Appendix E).

Theorem 5 (Redundancy of S for "optimal" variable $\vec{\beta}^*$) *The redundancy*

of $S^{\vec{\beta}^*}$ *with* $\beta_t^* = m_t/\ln\frac{t+1}{m_t}$ *is bounded by*

$$R_S^{\vec{\beta}^*}(x_{1:n}) \;\leq\; CL_w(\mathcal{A}) - (m-1)\ln m + \sum_{j\in\mathcal{A}} \tfrac{1}{2}\ln n_j - \tfrac{1}{2}\ln n + \tfrac{3}{2} m \ln\ln\tfrac{2n}{m} + 2.33m + 0.86 \tag{22}$$

The bounds (7), (8), (18), and (22), except for the first term, are independent of the base alphabet size D. For $w_i^t = 2^{-\mathrm{CL}(i)}$, the bounds are completely independent of D. They therefore also hold for countably infinite alphabet. Analogous to the Dirichlet-multinomial generalizing to the Chinese restaurant process, S can also be generalized to continuous spaces $\mathcal{X}$. The weights w_i^t become (sub)probability densities ($\int_{\mathcal{X}\setminus A} w_i^t \mathrm{d}i \leq 1$). The bounds remain valid, we only lose the code length interpretation of $\mathrm{CL}_w(\mathcal{A})$.

Proof idea. Unlike in (7) for constant β, $R_S^{\vec{\beta}^*}$ depends on the order of symbols and cannot be expressed in terms of Gamma functions bound by (27). Furthermore, β_t^* is generally not monotone in t, nor does it factor into monotone increasing and/or decreasing functions, which makes the analysis cumbersome but not impossible due to a different special property of β_t^*. I show that by swapping two consecutive symbols, x_t being $\mathcal{O}ld$ and x_{t+1} being $\mathcal{N}ew$, the redundancy always increases. It is therefore sufficient to upper bound $R_S^{\vec{\beta}^*}$ for sequences in which all new symbols come first before they repeat. For such a sequence, by separating $t \leq m$ for which $m_t = t$ and $t \geq m$ for which $m_t = m$, it is then possible to upper bound the handfull of resulting sums.

7 Comparison to Other Methods

In this section I theoretically (and in Section I experimentally) compare our models to various other more or less related ones, namely, the Dirichlet-multinomial with KT and Perks prior, and Bayesian sub-alphabet weighting. An experimental comparison can be found in Appendix I.

Dirichlet-multinomial distribution. The Dirichlet distribution

$$\mathrm{Dir}^{\boldsymbol{\alpha}}(\boldsymbol{\theta}) \;:=\; \frac{\Gamma(\alpha_+)}{\prod_i \Gamma(\alpha_i)} \prod_{i=1}^{D} \theta_i^{\alpha_i - 1}$$

with parameters $\alpha_i > 0$ and $\alpha_+ := \alpha_1 + ... + \alpha_D$ used as a Bayesian prior for $P_{iid}^{\boldsymbol{\theta}}$ leads to joint and predictive Dirichlet-multinomial distribution

$$\begin{aligned} \mathrm{DirM}^{\boldsymbol{\alpha}}(x_{1:n}) &:= \int P_{iid}^{\boldsymbol{\theta}}(x_{1:n}) \mathrm{Dir}^{\boldsymbol{\alpha}}(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} = \frac{\Gamma(\alpha_+)\prod_i \Gamma(n_i+\alpha_i)}{\Gamma(n+\alpha_+)\prod_i \Gamma(\alpha_i)}, \\ \mathrm{DirM}^{\boldsymbol{\alpha}}(x_{t+1}=i|x_{1:t}) &= \frac{n_i^t + \alpha_i}{t+\alpha_+} \qquad\qquad \text{with redundancy} \end{aligned}$$

$$R^{\boldsymbol{\alpha}}_{\mathrm{DirM}}(x_{1:n}) \quad = \quad \sum_{i=1}^{D} \ln \frac{n_i^{n_i}\Gamma(\alpha_i)}{\Gamma(n_i+\alpha_i)} - \ln \frac{n^n\Gamma(\alpha_+)}{\Gamma(n+\alpha_+)} \tag{23}$$

$$\stackrel{n_i\to\infty}{\longrightarrow} \quad \frac{D-1}{2}\ln\frac{n}{2\pi} + \sum_i (\tfrac{1}{2}-\alpha_i)\ln\frac{n_i}{n} + \sum_i \ln\Gamma(\alpha_i) - \ln\Gamma(\alpha_+) \tag{24}$$

If we choose constant weights $w_i^t = \alpha_i/\alpha_+$ and $\beta = \alpha_+$ in S, we see that $\mathrm{DirM}(x_{t+1} = i|x_{1:t})$ is the sum of both cases in (2), hence $\mathrm{DirM}(x_{t+1}=i|x_{1:t}) \geq S(x_{t+1}=i|x_{1:t})$. Therefore, the upper redundancy bound in Proposition 1 also holds for DirM: $R^{\boldsymbol{\alpha}}_{\mathrm{DirM}} \leq R_S^{\alpha_+}[w_i^t := \alpha_i/\alpha_+] \leq$ Eq.(8). The analysis in Section 5 suggests to set the Dirichlet parameters to $\alpha_i^* := w_i^0\beta^*$ for which $R^{\boldsymbol{\alpha}^*}_{\mathrm{DirM}} \leq R_S^{\beta^*}[w_i^t := \alpha_i/\alpha_+] \leq$ Eq.(18). If we allow for time-dependent α_i, Section 6 suggests to set $\alpha_i = \alpha_i^{t*} := w_i^t\beta_t^*$ for which $R^{\vec{\boldsymbol{\alpha}}^*}_{\mathrm{DirM}} \leq R_S^{\vec{\beta}^*} \leq$ Eq.(22), but note that weights w_i^t must normalize over $\mathcal{X}$ rather than $\mathcal{A}_t$ for DirM to form a (sub)probability. This can harm performance but only for large m. Note that for continuous $\mathcal{X}$ and weight *density* $w()$, S and DirM coincide.

The overall suggestion *if* using the (adaptive) Dirichlet-multinomial for prediction or compression or estimation is to choose variable parameters

$$\alpha_i \;=\; \alpha_i^{t*} \;:=\; \frac{m_t}{D\ln\frac{t+1}{m_t}} \quad \text{or} \quad \frac{2^{-\mathrm{CL}(i)}m_t}{\ln\frac{t+1}{m_t}} \tag{25}$$

The KT estimator. As can be seen from (24), for $\alpha_i = \frac{1}{2}$ the DirM redundancy (23) is asymptotically independent of the counts (n_i), and indeed it is well-known that asymptotically this is essentially also the best choice for the worst counts [KT81, Kri98, Wal05]. This so-called KT-estimator has minimax redundancy [BEY06]

$$R^{\mathbf{1}/2}_{\mathrm{DirM}} \;\leq\; \frac{D-1}{2}\ln n + \ln D \tag{26}$$

Asymptotically, this bound is essentially tight. We can compare this to our bound (18). For $m \ll n$, the dominant term in (18) is $\sum_j \frac{1}{2}\ln n_j$. This can be bounded by Jensen's inequality as

$$\sum_{j\in\mathcal{A}} \tfrac{1}{2}\ln n_j - \tfrac{1}{2}\ln n \;\leq\; \frac{m-1}{2}\ln\frac{n}{m} \;\leq\; \frac{m-1}{2}\ln n \;\leq\; \frac{D-1}{2}\ln n$$

so is clearly much smaller than (26) due to symbols that do not appear (gap in the third inequality) and symbols that appear rarely (gap in the first+second inequality). The latter happens often in particular for large m, but then the other terms in (18) gain relevance.

Sparse KT estimators. If we knew the used alphabet $\mathcal{A}$ in advance, we could employ the KT estimator on this sub-alphabet without reference to the base alphabet $\mathcal{X}$ and achieve much smaller redundancy $\leq \frac{m-1}{2}\ln n + \ln m$. In absence of such an oracle, we could code unordered $\mathcal{A}$ in advance in $\ln\binom{D}{m}$ nits, which gives an off-line

estimator with $\leq m\ln\frac{eD}{m}$ extra redundancy above the oracle. We can even get online versions: A light-weight way is at time t to use KT on $\mathcal{A}_t$ but reserve an escape probability of $\frac{1}{t+1}$ for and uniformly distribute it among the unseen symbols $\mathcal{X}\backslash\mathcal{A}_t$, which leads to a similar but larger extra redundancy of $\ln n+m\ln D+m+\ln 2$ [VH12]. A heavy-weight Bayesian solution is to take a weighted average over the $\mathrm{KT}_{\mathcal{A}'}$ estimators for all $\mathcal{A}'\subseteq\mathcal{X}$ [TSW93]. As prior one could take a uniform distribution over the size m' of $\mathcal{A}'$, and then for each m' a uniform distribution over all $\mathcal{A}'$ of size m' with extra redundancy $\leq m\ln\frac{eD}{m}+\ln D$. The resulting exponential mixture can be computed in linear time in D as discussed in Appendix G. This is still a factor of D slower than all other estimators considered in this paper. Otherwise the linear-time update rule has a similar structure to (20), and hence $S^{\vec{\beta}^*}$ may be derivable as an approximation to Bayesian sub-alphabet weighting.

8 Conclusion

I introduced and analyzed a model, closely related to the Dirichlet-multinomial distribution, which predicts an $\mathcal{O}ld$ symbol with its past frequency scaled down by $\frac{t}{t+\beta}$ and a new symbol with its weight, scaled down by $\frac{\beta}{t+\beta}$. Natural weight choices are uniform and $2^{-\mathrm{CodeLength}}$.

I derived exact expressions and for small m rather tight bounds for the code length and redundancy. The bounds were data-dependent rather then expected or worst-case bounds. This led to an (approximately) optimal choice of β different from traditional recommendations. The constant offline β^* (16) depends on the total sequence length n and number of different used symbols m. The variable online $\vec{\beta}^*$ (21) depends on the current sequence length t and number of different symbols observed so far m_t.

The redundancy bounds additionally depend on the individual symbol counts n_i themselves. They show that S^{β^*} has (at most) zero redundancy for unused symbols and finite redundancy for symbols occurring only finitely often, unlike the KT estimator and companions which have redundancy $\frac{1}{2}\ln n+O(1)$ per base symbol, whether it occurs or not. Indeed, my bounds are independent of the base alphabet size D, therefore also hold for denumerable and with suitable reinterpretation for continuous $\mathcal{X}$.

There seems to be not much leeway in choosing a globally good β. Experimentally it seems that even slight changes in β^* can significantly deteriorate performance in some (m,n,D)-regime, but can only marginally and locally improve performance in others. Empirically $S^{\vec{\beta}^*}$ seems superior to the other fast online estimators I compared it to. See Appendix I for some results.

As a simple, online, fast, i.i.d. estimator, $S^{\vec{\beta}^*}$ should be a useful alternative sub-component in more sophisticated (online) estimators/predictors/compressors/modellers such as large-alphabet CTW [TSW93] and others [VNHB12, OHSS12, Mah12]. The derived redundancy bounds are of

theoretical interest, not only for optimizing model parameters.

Acknowledgements. I thank the anonymous reviewers for valuable feedback, and in particular one reviewer for providing the efficient representation of the Bayesian sub-alphabet estimator in Appendix G.

References

[Bay63] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society*, 53:370–418, 1763. [Reprinted in *Biometrika*, 45, 296–315, 1958].

[Ber79] J. M. Bernardo. Reference posterior distributions for Bayesian inference (with discussion). *Journal of the Royal Statistical Society*, B41:113–147, 1979.

[BEY06] R. Begleiter and R. El-Yaniv. Superior guarantees for sequential prediction and lossless compression via alphabet decomposition. *Journal of Machine Learning Research*, 7:379411, 2006.

[BH10] W. Buntine and M. Hutter. A Bayesian view of the Poisson-Dirichlet process. Technical Report arXiv:1007.0296, NICTA and ANU, Australia, 2010.

[Bis06] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[CT06] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Intersience, 2nd edition, 2006.

[CW84] J. G. Cleary and I. H. Witten. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, COM-32(4):396–402, 1984.

[Fin74] B. de Finetti. *Theory of Probability : A Critical Introductory Treatment*. Wiley, 1974. Vol.1&2, transl. by A. Machi and A. Smith.

[Grü07] P. D. Grünwald. *The Minimum Description Length Principle*. The MIT Press, Cambridge, 2007.

[Hal48] J. B. S. Haldane. The precision of observed values of small frequencies. *Biometrika*, 35:297–300, 1948.

[HP05] M. Hutter and J. Poland. Adaptive online prediction by following the perturbed leader. *Journal of Machine Learning Research*, 6:639–660, 2005.

[Hut03] M. Hutter. Optimality of universal Bayesian prediction for general loss and alphabet. *Journal of Machine Learning Research*, 4:971–1000, 2003.

[Jef46] H. Jeffreys. An invariant form for the prior probability in estimation problems. In *Proc. Royal Society London*, volume Series A 186, pages 453–461, 1946.

[Jef61] H. Jeffreys. *Theory of Probability*. Clarendon Press, Oxford, 3rd edition, 1961.

[Kri98] R. E. Krichevskiy. Laplace's law of succession and universal encoding. *IEEE Transactions on Information Theory*, 44(1):296–303, 1998.

[KT81] R. Krichevsky and V. Trofimov. The performance of universal encoding. *IEEE Transactions on Information Theory*, 27(2):199–207, 1981.

[Lap12] P. Laplace. *Théorie analytique des probabilités*. Courcier, Paris, 1812. [English translation by F. W. Truscott and F. L. Emory: *A Philosophical Essay on Probabilities*. Dover, 1952].

[Mah12] M. Mahoney. *Data Compression Explained*. Dell, Inc, http://mattmahoney.net/dc/dce.html, 2012.

[MS99] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[OHSS12] A. O'Neill, M. Hutter, W. Shao, and P. Sunehag. Adaptive context tree weighting. In *Proc. Data Compression Conference (DCC'12)*, pages 317–326, Snowbird, Utah, USA, 2012. IEEE Computer Society.

[Per47] W. Perks. Some observations on inverse probability including a new indifference rule. *Journal of the Institute of Actuaries*, 73:285–334, 1947.

[Ris76] J. J. Rissanen. Generalized Kraft inequality and arithmetic coding. *IBM Journal of Research and Development*, 20(3):198–203, 1976.

[Ris84] J. J. Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information Theory*, I(4):629–636, 1984.

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[TSW93] T. J. Tjalkens, Y. M. Shtarkov, and F. M. J. Willems. Sequential weighting algorithms for multi-alphabet sources. *Proc. 6th Joint Swedish-Russian Intl. Workshop on Information Theory*, pages 22–27, 1993.

[VH12] J. Veness and M. Hutter. Sparse sequential Dirichlet coding. Technical Report arXiv:1206.3618, UoA and ANU, 2012.

[VNHB12] J. Veness, K. S. Ng, M. Hutter, and M. Bowling. Context tree switching. In *Proc. Data Compression Conference (DCC'12)*, pages 327–336, Snowbird, Utah, USA, 2012. IEEE Computer Society.

[Wal05] C. S. Wallace. *Statistical and Inductive Inference by Minimum Message Length*. Springer, Berlin, 2005.

[WNC87] I. H. Witten, , R. M. Neal, and J. G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987.

A Approximations of the (Di)Gamma Function

$$(x-\tfrac{1}{2})\ln x - x + \ln\sqrt{2\pi} \underset{\substack{\uparrow\\ \forall x>0}}{\leq} \ln\Gamma(x) \underset{\substack{\uparrow\\ \forall x\geq 1}}{\leq} (x-\tfrac{1}{2})\ln x - x + 1 \qquad (27)$$

The lower bound is asymptotically sharp for $x \to \infty$ but a factor of 2 too small for $x \to 0$. The absolute error of upper and lower bound for all $x \geq 1$ is at most $1-\ln\sqrt{2\pi} \doteq 0.081$. Some other used identities, asymptotics, and bounds are:

$$\sum_{t=1}^{n-1} \ln t \;=\; \ln\Gamma(n) \qquad (28)$$

$$1 - 1/x \;\leq\; \ln x \;\leq\; x - 1 \qquad [= \text{ iff } x = 1] \tag{29}$$

$$\Psi(z) \;=\; \frac{\mathrm{d}\ln\Gamma(z)}{\mathrm{d}z} \;\sim\; \ln z - O\Big(\frac{1}{z}\Big) \tag{30}$$

$$\Gamma(z) \;\leq\; \frac{1}{z} \quad \text{for} \quad z \;\leq\; 1 \tag{31}$$

B Proof of Theorem 2

I start with the lower bound (9) rewritten as

$$\underline{R}_S^\beta \;=\; \mathrm{CL}_w(\mathcal{A}) - m\ln\beta + \sum_{j\in\mathcal{A}} \tfrac{1}{2}\ln n_j + n\ln(1+\frac{\beta}{n}) + (\beta - \tfrac{1}{2})\ln(\frac{n}{\beta}+1) - m - [1 - \ln\sqrt{2\pi}] \tag{32}$$

which is valid for $\beta \geq 1$. Let

$$R(\beta) := -m\ln\beta + n\ln(1+\frac{\beta}{n}) + (\beta - \tfrac{1}{2})\ln(\frac{n}{\beta}+1)$$

be the β-dependent terms in (32).

For $\underline{1 \leq \beta \leq n}$,

$$R(\beta) \;\geq\; -m\ln\beta + (\beta - \tfrac{1}{2})\ln 2 \;\geq\; -m\ln m + m\ln\ln 2$$

The last inequality follows from minimizing the first w.r.t. β by differentiation and inserting the minimizer $\beta = m/\ln 2$ and dropping the second term.

For $\underline{\beta \geq n}$ and with abbreviations $z := n/\beta \leq 1$ and $\bar{\nu} = \frac{n}{m} \geq 1$ we get

$$\begin{aligned}
R(\beta) \;&\geq\; -m\ln\beta + n\ln\frac{\beta}{n} + \frac{\beta}{2}\ln\Big(\frac{n}{\beta}+1\Big) \\
&=\; (n-m)\ln\beta - n\ln n + \frac{n}{2}\frac{\ln(1+z)}{z} && \begin{bmatrix}\text{increasing in } \beta \\ \text{decreasing in } z\end{bmatrix} \\
&\geq\; (n-m)\ln n - n\ln n + \frac{n}{2}\ln 2 \\
&=\; -m\ln m + m[\bar{\nu}\tfrac{1}{2}\ln 2 - \ln\bar{\nu}] && [\text{minimized for } \bar{\nu} = 2/\ln 2] \\
&\geq\; -m\ln m + m[1 - \ln 2 + \ln\ln 2] \\
&\geq\; -m\ln m + m\ln\ln 2
\end{aligned}$$

which is the same as for $1 \leq \beta \leq n$. Plugging this into (32) we get for $\beta \geq 1$

$$R_S^\beta(x_{1:n}) \;\geq\; \mathrm{CL}_w(\mathcal{A}) - m\ln m + \sum_{j\in\mathcal{A}} \tfrac{1}{2}\ln n_j - m[1 - \ln\ln 2] - [1 - \ln\sqrt{2\pi}] \tag{33}$$

For $\underline{\beta \leq 1}$ we need to start with the exact expression (7):

$$\begin{aligned}
\sum_{j\in\mathcal{A}} \ln \frac{n_j^{n_j}}{\Gamma(n_j)} &\overset{(27)}{\geq} \sum_{j\in\mathcal{A}} [\tfrac{1}{2}\ln n_j + n_j - 1] \;=\; \sum_{j\in\mathcal{A}} [\tfrac{1}{2}\ln n_j] + n - m \\
-m\ln\beta + \ln\frac{1}{\Gamma(\beta)} &\overset{(31)}{\geq} (m-1)\ln\frac{1}{\beta} \;\geq\; 0 \\
\ln\frac{\Gamma(n+\beta)}{n^n} &\overset{(27)}{\geq} (n+\beta-\tfrac{1}{2})\ln(n+\beta) - (n+\beta) + \ln\sqrt{2\pi} - n\ln n \\
&= n\ln(1+\frac{\beta}{n}) + (\beta-\tfrac{1}{2})\ln(n+\beta) - (n+\beta) + \ln\sqrt{2\pi} \\
&\geq -\tfrac{1}{2}\ln(2n) - n - 1 + \ln\sqrt{2\pi}
\end{aligned}$$

Putting everything together we get for $\beta \leq 1$

$$R_S^\beta(x_{1:n}) \;\geq\; \mathrm{CL}_w(\mathcal{A}) + \sum_{j\in\mathcal{A}} \tfrac{1}{2}\ln n_j - \tfrac{1}{2}\ln n - m - [1 - \ln\sqrt{2\pi} + \tfrac{1}{2}\ln 2] \tag{34}$$

Pairing up terms (sometimes zero) in (33) and (34) and always taking the smaller one, we get after some rewrite (14), valid for all β.

C Derivation of Approximate Optimal $\boldsymbol{\beta^*}$

Exact implicit expression. The redundancy of S is minimized for

$$0 \;\overset{!}{=}\; \frac{\partial R_S^\beta}{\partial\beta} \;=\; -\frac{m}{\beta} + \Psi(n+\beta) - \Psi(\beta) \tag{35}$$

where $\Psi(x) := \mathrm{d}\ln\Gamma(x)/\mathrm{d}x$ is the diGamma function. Our goal is to approximately solve this equation w.r.t. β. Since no formal result in this paper explicitly uses that β^* is an approximate solution of (35), I only motivate the form of β^* by asymptotic considerations without discussing the accuracy of the approximation for finite n. With the following change in variables

$$0 < z := \frac{n}{\beta} < \infty \qquad \text{and} \qquad 0 < \nu := \frac{m}{n} < 1$$

(35) can be written as

$$\nu = \frac{1}{z}\Big[\Psi\big(n(1+\frac{1}{z})\big) - \Psi\big(\frac{n}{z}\big)\Big]$$

We need to solve this w.r.t. z for large n.

$\boldsymbol{\beta\to c>0}$ and $\boldsymbol{\beta\to\infty}$.

$$\underline{\beta\to\infty} \quad\Longrightarrow\quad \frac{n}{z}\to\infty \quad\overset{2\times(30)}{\Longrightarrow}\quad \nu \to \frac{1}{z}\Big[\ln\big(n(1+\frac{1}{z})\big) - \ln\big(\frac{n}{z}\big)\Big] \;=\; \frac{1}{z}\ln(1+z)$$

which is actually good for any n as long as $z=o(n)$. Next consider

$$\underline{\beta\to c>0} \quad\Longrightarrow\quad \frac{n}{z}\to c \quad\overset{(30)}{\Longrightarrow}\quad \nu\to\frac{1}{z}\Big[\underbrace{\ln(1+z)}_{\sim\ln n\to\infty}+\underbrace{\ln\big(\frac{n}{z}\big)-\Psi\big(\frac{n}{z}\big)}_{\to\ln(c)-\Psi(c)=const.}\Big] \;\sim\; \frac{1}{z}\ln(1+z)$$

Therefore we need to solve

$$\nu \;=\; g(z) := \frac{1}{z}\ln(1+z) \quad\text{for}\quad z=O(n), \qquad 0<z<\infty, \qquad 0<\nu<1$$

i.e. invert function g.

Lemma 6 (Inverse of $\ln(1+z)/z$) *The function $g(z):=\frac{1}{z}\ln(1+z)$ with domain $0<z<\infty$ is strictly monotone decreasing and has inverse $g^{-1}(\nu)=\frac{c(\nu)}{\nu}\ln\frac{1}{\nu}$ with domain $0<\nu<1$, where $c(\nu)$ is smooth and strictly monotone increasing from $c(0^+)=1$ to $c(1^-)=2$.*

Proof. Strict monotonicity of g and therefore existence of an inverse follows from

$$g'(z) \;=\; \frac{1}{z^2}\Big[\frac{z}{1+z}-\ln(1+z)\Big] \;\overset{(29)}{<}\; 0$$

I first study the asymptotics of $\nu=g(z)$ for $z\to0$ and $z\to\infty$.

$$\underline{z\to0} \quad\Longrightarrow\quad \nu\to1, \quad\text{more precisely}\quad \nu=1-\tfrac{1}{2}z+O(z^2) \quad\Longrightarrow\quad z\approx2(1-\nu)$$
$$\underline{z\to\infty} \quad\Longrightarrow\quad \nu\to0, \quad\text{and asymptotically}\quad z\approx\frac{1}{\nu}\ln\frac{1}{\nu}$$

I got the last expression by fixed point iteration: Rewrite $\nu=g(z)$ as $z=\frac{1}{\nu}\ln(1+z)$ and now iterate $z_{t+1}=\frac{1}{\nu}\ln(1+z_t)$ starting from any $0<z_0:=c<\infty$. This gives $z_1=\frac{1}{\nu}\ln(1+c)$ and

$$z_2 \;=\; \frac{1}{\nu}\ln\Big[1+\underbrace{\frac{1}{\nu}}_{\to\infty}\ln(1+c)\Big] \;\sim\; \frac{1}{\nu}\ln\Big[\frac{1}{\nu}\ln(1+c)\Big] \;=\; \frac{1}{\nu}\Big[\underbrace{\ln\frac{1}{\nu}}_{\to\infty}+\underbrace{\ln\ln(1+c)}_{finite}\Big] \;\sim\; \frac{1}{\nu}\ln\frac{1}{\nu}$$

No more iterations are needed! If we tentatively apply the $\nu\to0$ expression for $\nu\to1$ we get

$$z \;\sim\; \frac{1}{\nu}\ln\frac{1}{\nu} \;=\; (1-\nu)+O((1-\nu)^2) \;\to\; 0 \quad\text{for}\quad \nu\to1$$

The limit value is right, but the slope is $1/2$ of what it should be. $\frac{2}{\nu}\ln\frac{1}{\nu}$ would have the right slope at $\nu=1$. Therefore

$$z \;=\; \frac{c(\nu)}{\nu}\ln\frac{1}{\nu} \quad\text{for some function } c(\nu) \text{ with } c(0^+)=1 \text{ and } c(1^-)=2$$

which suggests that $c(\nu)$ might always lie in interval [1;2]. I prove this by showing that $c(\nu)$ is a monotone increasing function of ν.

$$\text{From}\quad g^{-1}(\nu)=\frac{c(\nu)}{\nu}\ln\frac{1}{\nu} \quad\text{we get}\quad c(\nu)=\frac{\nu g^{-1}(\nu)}{\ln(1/\nu)}$$

Since $g(z)$ is smooth, also $g^{-1}(\nu)$ and $c(\nu)$ are smooth. Since $g()$ is monotone decreasing, rather than proving $c()$ to be increasing, it is equivalently to show that

$$f(z) \;:=\; c(g(z)) \;=\; ... \;=\; \frac{\ln(1+z)}{\ln z - \ln\ln(1+z)}$$

is monotone decreasing in z. For this, it is sufficient to show

$$0 \;>\; f'(z) \;=\; ... \;=\; \frac{\ln z - \ln\ln(1+z) - \frac{1+z}{z}\ln(1+z) + 1}{(1+z)[\ln z - \ln\ln(1+z)]^2} \;=:\; \frac{h(z)}{\text{denominator}}$$

Since $h(0^+)=0$, it is sufficient to show $h'(z)<0$:

$$h'(z) \;=\; ... \;=\; \frac{[\ln(1+z)]^2 - \frac{z^2}{1+z}}{z^2\ln(1+z)} \;<\; 0 \qquad\Longleftrightarrow\qquad r(z) \;:=\; \ln(1+z) - \frac{z}{\sqrt{1+z}} \;<\; 0$$

Since $r(0^+)=0$, it is sufficient to show $r'(z)<0$:

$$r'(z) \;=\; ... \;=\; \frac{\sqrt{1+z} - (1+\frac{1}{2}z)}{(1+z)^{3/2}} \;<\; 0, \quad \text{which is true, since} \quad 1+z \;<\; (1+\tfrac{1}{2}z)^2$$

■

Approximation of $c(\nu)$. In Appendix F I discuss approximations for $c(\nu)$. In the main text I simply replace $c(\nu)$ by 1, i.e. $z=\frac{1}{\nu}\ln\frac{1}{\nu}$ which has the right asymptotics for the $\nu\to 0$ ($m\ll n$) regime I am primarily interested in and still the right limit for $\nu\to 1$. I also found that this choice is consistent with the other regimes in (17), in particular with $\beta\to 0$. Back in (n,m,β) notation we get

$$\beta \;=\; \frac{n}{z} \;=\; \frac{n}{\frac{1}{\nu}\ln\frac{1}{\nu}} \;=\; \frac{m}{\ln\frac{n}{m}} \;=:\; \beta^*$$

$\beta\to 0$. I finally consider the $\beta\to 0$ regime. Using the general recurrence $\Psi(\beta)=\Psi(\beta+1)-\frac{1}{\beta}$ in (35) we get

$$0 \;=\; -\frac{m-1}{\beta} + \Psi(n+\beta) - \Psi(\beta+1) \;\to\; -\frac{m-1}{\beta} + \Psi(n) - \Psi(1) \;\overset{(30)}{\sim}\; -\frac{m-1}{\beta} + \ln n$$

Solving this w.r.t. β we get $\beta=\frac{m-1}{\ln n}$. This has not yet the right form but since $0\leq\frac{\ln m}{\ln n}\leq\frac{m-1}{\ln n}=\beta\to 0$, we can write this as

$$\beta \;=\; \frac{m-1}{\ln n} \;\sim\; \frac{m-1}{(1-\frac{\ln m}{\ln n})\ln n} \;=\; \frac{m-1}{\ln\frac{n}{m}}$$

which apart from the -1 is consistent with the β-expressions in the other regimes.

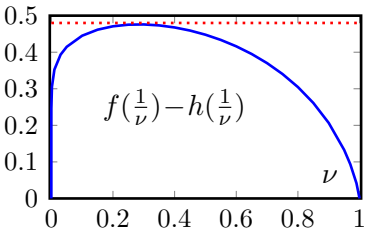
D Proof of Theorem 4

I first prove Theorem 4 for $m<n$. Inserting (16) into (8) and abbreviating $\bar\nu:=\frac{n}{m}>1$ we get after rearranging terms

$$R_S^{\beta^*} \;\le\; \overline{R}_S^{\beta^*} \;=\; \mathrm{CL}_w(\mathcal{A}) - m\ln m + \sum_{j\in\mathcal{A}} \tfrac{1}{2}\ln\frac{n_j}{2\pi} + 0.082 + m\cdot f(\bar\nu) - \tfrac{1}{2}\ln(\bar\nu\ln\bar\nu+1)$$

$$\text{where}\quad f(\bar\nu) \;:=\; \ln\ln\bar\nu + \bar\nu\ln(1+\frac{1}{\bar\nu\ln\bar\nu}) + \frac{\ln(\bar\nu\ln\bar\nu+1)}{\ln\bar\nu}$$

It is easy to see that $f(\bar\nu)\sim\ln\ln\bar\nu$ for $\bar\nu\to\infty$ and $f(1^+)=1$. This motivates the approximation $h(\bar\nu):=1+\ln(1+\ln\bar\nu)$, which has the correct $\bar\nu\to1$ limit and correct $\bar\nu\to\infty$ asymptotics. Next I upper bound $f(\bar\nu)-h(\bar\nu)$. Since $f-h$ is continuous and tends to zero at 0 and at 1, it is upper bounded by some finite constant. It is easy to see graphically and numerically but quite cumbersome to show analytically that $f(\frac{1}{\nu})-h(\frac{1}{\nu})$ is concave for $0<\nu<1$ with maximum 0.476... at $\nu=0.284...$, hence $f(\bar\nu)\le1.48+\ln(1+\ln\bar\nu)$. Now using $\frac{1}{2}\ln\frac{n_j}{2\pi}\doteq\frac{1}{2}\ln n_j-0.92$ and $-\frac{1}{2}\ln(\bar\nu\ln\bar\nu+1)\le-\frac{1}{2}\ln\bar\nu$ (use $\ln x\ge1-1/x$ on the inner $\ln\bar\nu$) leads to the desired bound (18) for $m<n$.



For $m=n$, we have $n_i=1\forall i$, hence $P_{iid}^{\hat{\boldsymbol\theta}}=(\frac{1}{n})^n$ from (6), and $\beta^*=\infty$, hence $S(x_{t+1}=i|x_{1:t})=w_i^t$ from (2), so $S(x_{1:n})=\mathrm{CL}(\mathcal{A})$. Inserting this into (5) gives $R_S^\infty(x_{1:n})=\mathrm{CL}(\mathcal{A})-n\ln n$. On the other hand, (18) for $m=n$ is $\mathrm{CL}(\mathcal{A})-n\ln n+0.56n+0.082$, which is clearly larger. ■

1.48 is a quite crude upper bound on $f(1^+)=1$. By introducing ugly other terms, one can improve 1.48 to 1 and hence $0.56m$ to $0.082m$ in bound (18).

E Proof of Theorem 5

***S* and *R* for variable $\vec\beta$.** For variable $\vec\beta$ the joint S distribution and its redundancy are

$$S^{\vec\beta}(x_{1:n}) \;=\; \prod_{t=0}^{n-1} S^{\beta_t}(x_{t+1}|x_{1:t}) \;=\; \prod_{t=0}^{n-1}\frac{1}{t+\beta_t}\prod_{t\in\mathcal{O}ld} n^t_{x_{t+1}}\prod_{t\in\mathcal{N}ew}\beta_t w^t_{x_{t+1}}$$

$$R_S^{\vec\beta} \;=\; \underbrace{\mathrm{CL}_w(\mathcal{A})}_{(I)} + \underbrace{\sum_{t=1}^{n-1}\ln(t+\beta_t)}_{(II)} \underbrace{-\sum_{t\in\mathcal{N}ew\setminus\{0\}}\ln\beta_t}_{(III)} + \underbrace{\sum_{j\in\mathcal{A}}\ln\frac{n_j^{n_j}}{\Gamma(n_j)}}_{(IV)} \underbrace{-n\ln n}_{(V)} \qquad (36)$$

In the redundancy I removed the $\ln(0+\beta_0)-\ln(\beta_0)$ contribution. Note that $S^{\vec\beta}$ and $R_S^{\vec\beta}$ are now not only dependent on the counts but also on exactly when new symbols

appear, i.e. on the $\mathcal{N}ew$ set. (for $\vec{\beta}^*=m_t/\ln\frac{t+1}{m_t}$ the dependence is in a sense mild though). The sums cannot be represented as Gamma functions anymore.

(I) and (IV) and (V) are independent of $\mathcal{N}ew$, for (I) by assumption. (III) obviously depends on $\mathcal{N}ew$ but also (II) via m_t in β_t.

Redundancy change when swapping two consecutive symbols. I first show that the earlier new symbols appear, the larger is $R_S^{\vec{\beta}^*}$. This fact heavily relies on the specific form of $\vec{\beta}^*$, which makes the proof cumbersome. Assume at time t there is an old symbol but at time $t+1$ there is a new symbol for some $t\in\{1...n-1\}$. That is, $t\in\mathcal{O}ld$ and $m_{t-1}=m_t$ but $t+1\in\mathcal{N}ew$ and $m_{t+1}=m_t+1$. Note that $m_t<t$, and $x_{t+1}\neq x_t$, since x_t is old and x_{t+1} is new. I now swap x_t with x_{t+1}. I mark all quantities that change by a prime $'$. That is, $x'_t=x_{t+1}$ and $x'_{t+1}=x_t$. Now x_t is new ($t-1\in\mathcal{N}ew'$) and x_{t+1} is old ($t\in\mathcal{O}ld'$). Further $m'_t=m_t+1$, and $\beta'^*_t=m'_t/\ln\frac{t+1}{m'_t}$. Quantities for all other t remain unchanged. Only one term in (II) and one term in (III) are affected. The change in redundancy is therefore

$$\begin{aligned}\Delta R(t,m_t) &:= R_S^{\vec{\beta}'^*}-R_S^{\vec{\beta}^*} = \ln(t+\beta'^*_t)-\ln(t+\beta^*_t)-\ln\beta'^*_{t-1}+\ln\beta^*_t\\ &= \ln(t+\frac{m_t+1}{\ln\frac{t+1}{m_t+1}})-\ln(t+\frac{m_t}{\ln\frac{t+1}{m_t}})-\ln\frac{m_t}{\ln\frac{t}{m_t}}+\ln\frac{m_t}{\ln\frac{t+1}{m_t}}\end{aligned}$$

where I have used $m'_{t-1}=m_{t-1}=m_t$. Collecting terms we get

$$\Delta R(t,m) = \ln\frac{\ln\frac{t}{m}+\frac{m+1}{t}\frac{\ln\frac{t}{m}}{\ln\frac{t+1}{m}}}{\ln\frac{t+1}{m}+\frac{m}{t}} \stackrel{?}{>} 0 \quad\text{for}\quad 0<m<t$$

This is positive, if the numerator is larger than the denominator. Rearranging terms we can write this as

$$f_{t,m}(0)\stackrel{?}{>}f_{t,m}(1),\quad\text{with}\quad f_{t,m}(a) := \ln\frac{t+a}{m}-\frac{m+a}{t}\frac{\ln\frac{t}{m}}{\ln\frac{t+a}{m+a}}$$

Another change in variables gives us

$$f_{\bar\nu}(x):=f_{t,m}(a)=\ln[\bar\nu(1+x)]-(\frac{1}{\bar\nu}+x)\frac{\ln\bar\nu}{\ln\frac{1+x}{1/\bar\nu+x}},\quad\text{where}\quad a=x\cdot t\text{ and }\bar\nu:=\frac{t}{m}>1$$

By differentiation one can show that $f_{\bar\nu}(x)$ is a decreasing function in x for all $x>0$ and $\bar\nu>1$, which implies $f_{t,m}(0)>f_{t,m}(1)$ and hence $\Delta R(t,m)>0$.

Bounding the redundancy for all new symbols first. We can repeat swapping symbols and thereby increasing $R_S^{\vec\beta}$ until all symbols appear first before they repeat, that is, $m_t=\min\{t,m\}$ and $\mathcal{N}ew=\{0,...,m-1\}$. For this oder we have

$$\beta^*_t=\frac{t}{\ln\frac{t+1}{t}}\geq t^2\quad\text{for}\quad t\leq m,\qquad\text{and}\qquad b^*_t=\frac{m}{\ln\frac{t+1}{m}}\quad\text{for}\quad t\geq m$$

I now bound each of the 5 terms (I)-(V) in $R_S^{\vec{\beta}}$, where I split the sum in (II) and merge in (III).

$$\text{(I)} = \underline{\mathrm{CL}_w(\mathcal{A})} \quad \text{and} \quad \text{(V)} = \underline{-n\ln n} \qquad \text{[nothing to do here]}$$

$$\text{(IIa)+(III)} \;=\; \sum_{t=1}^{m-1}\ln(t+\beta_t^*) - \sum_{t=1}^{m-1}\ln\beta_t^* \;=\; \sum_{t=1}^{m-1}\ln(1+\frac{t}{\beta_t^*}) \;\leq\; \sum_{t=1}^{m-1}\frac{t}{\beta_t^*} \;\leq\; \sum_{t=1}^{m-1}\frac{1}{t} \;\leq\; \underline{1+\ln m}$$

$$\text{(IIb)} \;=\; \sum_{t=m}^{n-1}\ln(t+\beta_t^*) \;=\; \sum_{t=m}^{n-1}\ln t + \sum_{t=m}^{n-1}\ln(1+\frac{m/t}{\ln\frac{t+1}{m}})$$

Using (27) and (28), the first terms can be bound by

$$\begin{aligned}\text{(IIb1)} \;&=\; \sum_{t=m}^{n-1}\ln t \;=\; \ln\Gamma(n) - \ln\Gamma(m)\\ &\leq\; \underline{(n-\tfrac{1}{2})\ln n - n + 1 - (m-\tfrac{1}{2})\ln m + m - \ln\sqrt{2\pi}}\end{aligned}$$

I split the second term in (IIb) further into $t<2m$ and $t\geq 2m$:

$$\begin{aligned}\text{(IIb2)} \;&=\; \sum_{t=m}^{\min\{2m-1,n-1\}}\ln(1+\frac{m/t}{\ln\frac{t+1}{m}}) \overset{\ln\frac{t+1}{m}\geq 1-\frac{m}{t+1}}{\underset{\downarrow}{\leq}} \sum_{t=m}^{2m-1}\ln(1+\frac{\overbrace{(t+1)m}^{>1}}{t(t+1-m)}) \;\leq\; \sum_{t=m}^{2m-1}\ln\frac{2m(t+1)}{t(t+1-m)}\\ &=\; m\ln(2m) + \ln\tfrac{2m}{m} - \ln m! \;\leq\; \underline{m\ln(2m) + \ln 2 - (m+\tfrac{1}{2})\ln m + m - \ln\sqrt{2\pi}}\end{aligned}$$

If $2m<n$

$$\text{(IIb3)} \;=\; \sum_{t=2m}^{n-1}\ln(1+\frac{m/t}{\ln\frac{t+1}{m}}) \;\leq\; \sum_{t=2m}^{n-1}\frac{m/t}{\ln\frac{t+1}{m}} \;\leq\; \frac{3}{2}\sum_{t=2m}^{n-1}\frac{1}{\frac{t+1}{m}\ln\frac{t+1}{m}}$$

where I have used $\frac{t+1}{t} = 1+1/t \leq 1+1/2m \leq 3/2$. If we upper bound the sum by an integral and set $x=\frac{t+1}{m}$, we get

$$\leq \frac{3}{2}\int_{2m-1}^{n-1}\frac{\mathrm{d}t}{\frac{t+1}{m}\ln\frac{t+1}{m}} = \frac{3}{2}\int_2^{n/m}\frac{m\mathrm{d}x}{x\ln x} = \frac{3}{2}m[\ln\ln\frac{n}{m} - \ln\ln 2] \leq \underline{\frac{3}{2}m[\ln\ln\frac{2n}{m} - \ln\ln 2]}$$

If $2m\geq n$, (IIb3)=0. We can stich both cases together by either using a max-operation, or as I have done by increasing $n\rightsquigarrow 2n$, which ensures that the last expression is never negative.

$$\text{(IV)} \;=\; \sum_{j\in\mathcal{A}}\ln\frac{n_j^{n_j}}{\Gamma(n_j)} \;\leq\; \sum_{j\in\mathcal{A}}[\tfrac{1}{2}\ln n_j + n_j - \ln\sqrt{2\pi}] \;=\; \underline{n - \frac{m}{2}\ln(2\pi) + \sum_{j\in\mathcal{A}}\tfrac{1}{2}\ln n_j}$$

Putting everything together. We can now collect all underlined terms together and get

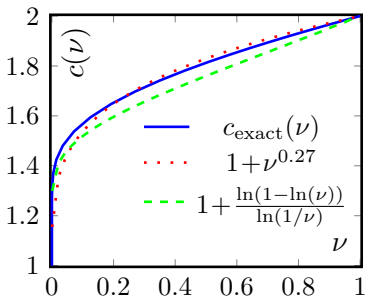
$$\begin{aligned} R_S^{\vec{\beta}^*}(x_{1:n}) &\leq \mathrm{CL}_w(\mathcal{A}) - (m-1)\ln m + \textstyle\sum_{j\in\mathcal{A}} \frac{1}{2}\ln n_j - \frac{1}{2}\ln n \\ &+ \tfrac{3}{2} m \ln\ln \tfrac{2n}{m} + m[2+\ln 2 - \tfrac{3}{2}\ln\ln 2 - \ln\sqrt{2\pi}] + [2-\ln\pi] \end{aligned}$$

Since the all-new-symbols-first order has maximal redundancy, the bound holds in general.

F Improvement on β^*

Here I generalize β^* to $\beta^c := \beta^*/c$. From Appendix C we know that for $n\to\infty$, the exact optimal β^c has $1\leq c(\nu)\leq 2$.

Discussion of $c(\nu)$. The figure on the right plots the exact function $c(\nu)$ implicitly given by $g(\frac{c}{\nu}\ln\frac{1}{\nu})=\nu$. While it is true that $c\to 1$ for $\nu\to 0$, β^1 only starts to have lower redundancy than β^2 for very small values of ν, namely $\nu \lesssim 10^{-2}$. So in practice, $c=2$ should perform better except for $m \lesssim \frac{n}{100}$. We could try to find approximate $c(\nu)$ in various ways, e.g. $c(\nu)=1+\nu^{0.27}$ makes $|g(\frac{c(\nu)}{\nu}\ln\frac{1}{\nu})-\nu| < 0.002$. $c(\nu)=1+\frac{\ln(1-\ln(\nu))}{\ln(1/\nu)}$ is theoretically motivated by an extra iteration of g.



Constant β^c. The proof of Theorem 4 in Appendix D still goes through for $\beta^*\rightsquigarrow\beta^c$ with now

$$f_c(\bar\nu) \;=\; \ln(c\ln\bar\nu) + \bar\nu\ln(1+\frac{1}{c\bar\nu\ln\bar\nu}) + \frac{\ln(c\bar\nu\ln\bar\nu+1)}{c\ln\bar\nu}$$

$f_c - 1 - \ln(1+\ln\bar\nu)$ is still upper bounded by 1.48 for all $1\leq c\leq 2$, so the upper bound in (18) is still valid for $\beta^*\rightsquigarrow\beta^c$.

Variable $\vec\beta^{\vec c}$. The proof of Theorem 5 in Appendix E breaks down for $c>1$. R still increases when moving new symbols earlier if many symbols have already appeared but actually decreases when only a few symbols have appeared so far. That is, $\Delta R(t,m)>0$ for large m as before, but $\Delta R(t,m)<0$ for small m. R is therefore maximized if all new symbols appear somewhere in the middle of the sequence. This may lead to a proof and bound analogous to the $c=1$ case.

Here is a simpler proof with a possibly cruder bound. I reduce $R_S^{\vec\beta^{\vec c}}$ to $R_S^{\vec\beta^*}$ and also allow for time-dependent $c=c_t$. From expression (36) it is easy to see that

$$R_S^{\vec\beta^{\vec c}} \;=\; R_S^{\vec\beta^*} + \sum_{t=1}^{n-1}\ln\frac{1+\beta_t^{c_t}}{1+\beta_t^*} + \sum_{t\in\mathcal{N}ew\setminus\{0\}}\ln c_t \;\leq\; (m-1)\ln 2$$

where I have exploited $c_t\leq 2$ and $\beta_t^{c_t}\leq\beta_t^*$ for $c_t\geq 1$. That is, if we add another $(m-1)\ln 2$ to bound (22) it becomes valid for $\vec\beta^{\vec c}$ for any choice of $1\leq c_t\leq 2$.

G Bayesian sub-alphabet weighting

The Bayesian sub-alphabet weighting estimator [TSW93] averages over the $\mathrm{KT}_{\mathcal{A}'}$ estimators for all possible $\mathcal{A}' \subseteq \mathcal{X}$ with a prior uniform in $|\mathcal{A}'|$ and uniform in $\mathcal{A}'$ given $|\mathcal{A}'|$:

$$P_{\mathrm{Bayes}}(x_{1:n}) \;=\; \sum_{\mathcal{A}':\mathcal{A}\subseteq\mathcal{A}'\subseteq\mathcal{X}} \mathrm{Prior}(\mathcal{A}') P_{\mathrm{KT}_{\mathcal{A}'}}(x_{1:n}) \quad \text{with} \quad \mathrm{Prior}(\mathcal{A}') \;=\; \frac{1}{D\binom{D}{|\mathcal{A}'|}} \tag{37}$$

This mixture of exponential size 2^{D-m} can be computed in time and space linear in D [TSW93]:

$$P_{\mathrm{Bayes}}(x_{1:n}) \;=\; \sum_{i=1}^{D} \frac{1}{D} G_i(x_{1:n}) \tag{38}$$

with the following sequential representation of G_i:

$$G_i(x_{t+1}|x_{1:t}) \;:=\; \begin{cases} 0 & \text{if} \quad m_{t+1} > i \\ \dfrac{n^t_{x_{t+1}} + \frac{1}{2}}{t + i/2} & \text{if} \quad m_{t+1} \leq i \quad \& \quad x_{t+1} \in \mathcal{A}_t \\ \dfrac{i - m_t}{D - m_t} \cdot \dfrac{1/2}{t + i/2} & \text{if} \quad m_{t+1} \leq i \quad \& \quad x_{t+1} \notin \mathcal{A}_t \end{cases}$$

This is still a factor D slower than all other estimators considered in this paper.

A relation to S can be enforced as follows: First, generalize $P_{\mathrm{KT}_{\mathcal{A}'}} \equiv P^{\mathbf{1/2}}_{\mathrm{DirM}_{\mathcal{A}'}}$ to $P^{\boldsymbol{\alpha}}_{\mathrm{DirM}_{\mathcal{A}'}}$, then

$$G^{\alpha}_{\beta/\alpha}(x_{t+1}|x_{1:t}) \;\stackrel{\alpha\to 0}{\longrightarrow}\; S^{\beta}(x_{t+1}|x_{1:t}) \quad \text{for} \quad w^t_i = \frac{1}{D - m_t}$$

While (37) mixes G_i's, S^{β^*} maximizes S^{β}. So S^{β^*} with uniform renormalized weights might be an integer-relaxed, maximum-likelihood approximation of Bayesian sub-alphabet weighting with Haldane prior. There are several caveats though.

An anonymous reviewer suggested the following alternative representation:

$$\begin{aligned} P_{\mathrm{Bayes}}(x_{t+1}|x_{1:t}) \;\propto\; P_{\mathrm{Bayes}}(x_{1:t+1}) \;&=\; \sum_{\mathcal{A}':\mathcal{A}_{t+1}\subseteq\mathcal{A}'\subseteq\mathcal{X}} \mathrm{Prior}(\mathcal{A}') P_{\mathrm{KT}_{\mathcal{A}'}}(x_{1:t+1}) \\ &=\; \sum_{\mathcal{A}':\mathcal{A}_{t+1}\subseteq\mathcal{A}'\subseteq\mathcal{X}} \mathrm{Prior}(\mathcal{A}') \frac{\Gamma(\frac{1}{2}|\mathcal{A}'|)}{\Gamma(t+1+\frac{1}{2}|\mathcal{A}'|)} \prod_{i\in\mathcal{X}} \frac{\Gamma(n^{t+1}_i+\frac{1}{2})}{\Gamma(\frac{1}{2})} \\ &=\; (n^t_{x_{t+1}}+\tfrac{1}{2}) \Bigg(\prod_{i\in\mathcal{X}} \frac{\Gamma(n^t_i+\frac{1}{2})}{\Gamma(\frac{1}{2})} \Bigg) \frac{1}{D} \sum_{\mathcal{A}':\mathcal{A}_{t+1}\subseteq\mathcal{A}'\subseteq\mathcal{X}} \binom{D}{|\mathcal{A}'|}^{-1} \frac{\Gamma(\frac{1}{2}|\mathcal{A}'|)}{\Gamma(t+1+\frac{1}{2}|\mathcal{A}'|)} \\ &\propto\; (n^t_{x_{t+1}}+\tfrac{1}{2}) \sum_{m'=m_{t+1}}^{D} \binom{D-m_{t+1}}{m'-m_{t+1}} \binom{D}{m'}^{-1} \frac{\Gamma(\frac{1}{2}m')}{\Gamma(t+1+\frac{1}{2}m')} \end{aligned}$$

The latter sum can have two values, depending on whether x_{t+1} is new ($m_{t+1}{=}m_t{+}1$) or old ($m_{t+1}{=}m_t$). We can hence write this as

$$P_{\text{Bayes}}(x_{t+1}=i|x_{1:t}) \;\propto\; \begin{cases} (n_i^t+\frac{1}{2})\gamma^t_{m_t} & \text{if} \quad n_i^t>0, \\ \frac{1}{2}\gamma^t_{m_t+1} & \text{if} \quad n_i^t=0, \end{cases}$$
$$\text{where} \qquad \gamma^t_m \;:=\; \sum_{m'=m}^{D} \binom{D-m}{m'-m}\binom{D}{m'}^{-1} \frac{\Gamma(\frac{1}{2}m')}{\Gamma(t{+}1{+}\frac{1}{2}m')}$$

By summation, the normalizer can be worked out to be $(t+\frac{1}{2}m_t)\gamma^t_{m_t}+\frac{1}{2}(D-m_t)\gamma^t_{m_t+1}$, which allows us to rewrite the result as

$$P_{\text{Bayes}}(x_{t+1}=i|x_{1:t}) \;=\; \begin{cases} \frac{n_i^t+1/2}{t+m_t/2+\beta_t} & \text{if} \quad n_i^t>0, \\ \frac{\beta_t/(D-m_t)}{t+m_t/2+\beta_t} & \text{if} \quad n_i^t=0, \end{cases} \quad \text{with} \quad \beta_t := \frac{D-m_t}{2}\frac{\gamma^t_{m_t+1}}{\gamma^t_{m_t}} \tag{39}$$

This has the same structure as (20) apart from the $+1/2$ and $+m_t/2$, which is due to using the KT prior rather than a Haldane prior, and apart from a significantly more complex expression for β_t, which I expect to be approximately β_t^*. An advantage of (39) over (38) is that not only can it be used to compute $P_{\text{Bayes}}(x_{t+1}|x_{1:t})$ in time $O(D)$ but also the cumulative distribution $P_{\text{Bayes}}(X_{t+1}{<}x_{t+1}|x_{1:t})$, required for arithmetic coding.

H Algorithms & Applications & Computation Time

All estimators discussed in this paper, except for Bayesian sub-alphabet weighting (SAW-Bayes) require just $O(1)$ time and $O(D)$ space for computing $P(x_{t+1}|x_{1:t})$ and for updating the relevant parameters like counts n_i, the number m_t of symbols seen so far, parameter β_t^*, etc. Space can be reduced to $O(m)$ by hashing. Only SAW-Bayes requires $O(D)$ time per t and $O(D)$ space.

Knowledge of $P(x_{t+1}|x_{1:t})$ for all t allows to determine code length, likelihood, and redundancy of $x_{1:n}$, relevant and sufficient e.g. for model selection such as MDL. Many other tasks like data compression via arithmetic encoding and Bayesian decision making require $P(X_{t+1}{=}i|x_{1:t})$ for all (or at least multiple) $i{\in}\mathcal{X}$, which naively requires $O(D)$ time per t.

For arithmetic encoding, we actually only need the conditional distribution function $P(X_{t+1}{<}x_{t+1}|x_{1:t})$ at x_{t+1} for $\mathcal{X}{\cong}\{1,...,D\}$. For DirM and S this can be computed in time $O(\log D)$ as follows: Maintain a binary tree of depth $\lceil\log_2 D\rceil$ with counts $n_1,n_2,...,n_D$ at the leafs in this order. Inner nodes store the sum of their two children. In this tree, computing $\sum_{i<x_{t+1}} n_i$ and updating $n_{x_{t+1}} \rightsquigarrow n_{x_{t+1}}{+}1$ can be performed in time $O(\log D)$ by accessing/updating the single path from root to leaf x_{t+1}. It is clear how this allows to compute $\text{DirM}(X_{t+1}{<}x_{t+1}|x_{1:t})$ in time $O(\log D)$

and space $O(D)$. Time can be reduced to $O(\log m)$ and space to $O(m)$ by maintaining a self-balancing binary tree of only the non-zero counts, which is rebalanced when inserting new non-zero counts.

To compute $S^{\vec{\beta}^*}(X_{t+1}<x_{t+1}|x_{1:t})$ in time $O(\log D)$, we have to additionally and in the same way store and maintain $\tilde{w}_1,\tilde{w}_2,...,\tilde{w}_D$ at the leafs (and their sum at inner nodes), where $\tilde{w}_i=w_i^0$ if $i\notin\mathcal{A}_t$ and $\tilde{w}_i=0$ else.

Expectations $\sum_i f(i)P(X_{t+1}=i|x_{1:t})$ can easily be updated in $O(1)$ time with $O(m)$ space, hence Bayes-optimal decisions $\text{argmin}_{y\in\mathcal{Y}}\sum_i \text{Loss}(y,i)P(X_{t+1}=i|x_{1:t})$ can be updated in $O(|\mathcal{Y}|)$ time.

A similar tree construction can speed up SAW-Bayes (38) from $O(D^2)$ to $O(D\log D)$, or one uses (39), but time $O(D)$ seems not further improvable. This renders SAW-Bayes impractical for large-alphabet data compression.

Finally, if computation time is at a premium and the logarithm in β_t^* too slow, one can with virtually no loss in compression quality update $1/\ln\frac{t+1}{m_t}$ only whenever m_t or t have changed by more than 10% since the last update.

I Experiments

I determined the code length of various estimators for various sequence lengths n, used alphabet sizes m, and base alphabet sizes D on artificially generated data sequences and the Calgary corpus. I consider the new estimator S and the Dirichlet-multinomial with approximately optimal constant $\beta^*\!/2$ and variable $\vec{\beta}^*\!/2$ and with Perks prior, the KT estimator for the base and for the used alphabet, and Bayesian sub-alphabet weighting, introduced in Section 7. I also compare against the true distribution and the empirical entropy.

Data generation. I sampled $\theta_1,...,\theta_m$ uniformly from the $m-1$-dimensional probability simplex and set $\theta_{m+1}=...=\theta_D=0$. I then sampled $x_{1:n}$ from $P_{iid}^{\boldsymbol{\theta}}$. Unless $n\gg m$ or $D\gg n$, this usually results in sequences that actually contain less than m symbols, and e.g. $|\mathcal{A}|=n$ is virtually impossible to achieve in this way. I therefore generate sequences by first setting $x_t=t$ for $t=1...\min\{m,n\}$, then sample the remaining x_t from $P_{iid}^{\boldsymbol{\theta}}$, and then scramble the result. The resulting code lengths were virtually indistinguishable from the "normal" i.i.d. sampling, when the latter was also feasible.

I also generated sequences with a version of D'Hondt's method for allocating seats in party-list proportional representation, which ensures $|n_i-\theta_i\cdot n|<1$ and adapted it to also ensure $n_i>0$ if $\theta_i>0$ and $i\leq n$ by dividing by zero (rather than 1) first. As expected, the results were a bit less noisy, but otherwise very similar.

In another experiment I chose $\boldsymbol{\theta}$ to be Zipf-distributed, i.e. $\theta_i\propto i^{-\gamma}$ with varying Zipf exponent $\gamma>0$, which for $\gamma\approx 1$ mimics quite well the empirical distribution of words in English texts. The larger γ, the smaller the used alphabet $\mathcal{A}$.

My S-estimators. I determined the code length of my models ($S^{\beta^*\!/2}$and $S^{\vec{\beta}^*\!/2}$) with constant and variable optimal β^*. I chose uniform normalized weights $w_i^t=$

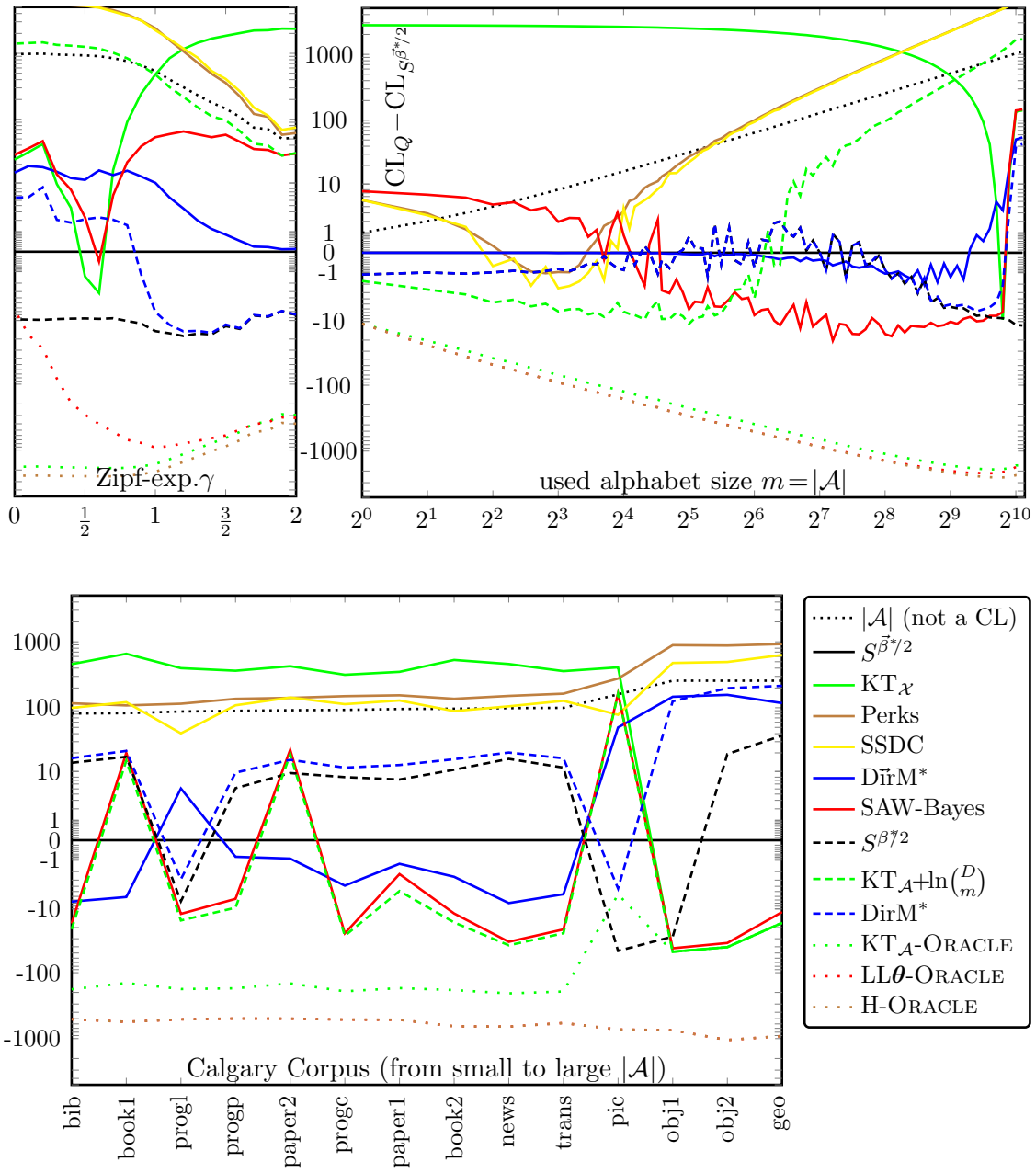


Figure 1: Plotted are code length differences to $S^{\vec{\beta}*/2}$ of various estimators. The two top graphs are for fixed sequence length $n=1024$ and total alphabet size $D=10\,000$ for varying Zipf exponents γ and used alphabet sizes $m=|\mathcal{A}|$. The bottom graph is for the 14 files from the Calgary corpus with $21504\leq n\leq 768\,771$ and byte alphabet ($D=256$). The online/offline/oracle estimators have solid/dashed/dotted lines. A curve above/below zero means worse/better than $S^{\vec{\beta}*/2}$. The black dotted curve is not a code length but shows the used alphabet size $m\equiv|\mathcal{A}|$.

$1/(D-m_t)$. I also played around with other β and $\vec{\beta}$, but performance either severely deteriorated, or only marginally and locally improved. The code length is very sensitive to some changes, e.g. $\beta=m/\ln n$ and $\beta=m/\ln\frac{2n}{m}$ perform badly for large m, since these β have the wrong scaling for $m\to n$, but less sensitive to other changes, e.g. $\beta=(m+c)/\ln\frac{n+c'}{m+c}$ for small c,c' are generally ok. For the experiments I used $\beta^{c=2}=\beta^*\!/2$ and $\beta_t^{c=2}=\beta_t^*/2$.

Other estimators. I also determined the code length of the other estimators discussed in Section 7. I considered:
(i) the Dirichlet-multinomial with $\boldsymbol{\alpha}=\mathbf{1}/D$ (Perks) and optimized constant $\boldsymbol{\alpha}^*$ (DirM^*) and optimal variable $\vec{\boldsymbol{\alpha}}^*$ ($\text{D}\vec{\text{i}}\text{r}\text{M}^*$) with uniform weights (25);
(ii) the KT-estimator with base alphabet $\mathcal{X}$ ($\text{KT}_{\mathcal{X}}$),
(iii) the KT-estimator for used alphabet $\mathcal{A}$ ($\text{KT}_{\mathcal{A}}$-ORACLE), a feasible off-line version by pre-coding $\mathcal{A}$ ($\text{KT}_{\mathcal{A}}+\ln\binom{D}{m}$), and the online version using escape probability $1/_{t+1}$ (SSDC) discussed in Section 7;
(iv) the Bayesian sub-alphabet weighting (SAW-Bayes) discussed in Appendix G;
(v) the empirical entropy $nH(\frac{\boldsymbol{n}}{n})=\sum_i n_i\ln\frac{n}{n_i}$ (H-ORACLE);
(vi) the log-likelihood of the sampling distribution $\ln 1/P_{iid}^{\boldsymbol{\theta}}$ (LL$\boldsymbol{\theta}$-ORACLE) for artificial data.

Results. Figure 1 plots the results for the various estimators. The vertical axis is the code length (or redundancy) difference of the estimator under consideration and our prime model $S^{\vec{\beta}^*/2}$. So negative/positive values indicate better/worse performance than $S^{\vec{\beta}^*/2}$. The two top graphs are for artificially generated data with fixed sequence length $n=1024$ and total alphabet size $D=10\,000$. In the right graph I varied $m=1,2,4,...2^{10}$ and in the left graph I varied the Zipf exponent $\gamma\in[0;2]$. The bottom graph shows results for the 14 files from the Calgary corpus with byte alphabet ($D=256$). All results are plotted and discussed relative to $S^{\vec{\beta}^*/2}$. Rather than averaging over multiple runs and plotting error bars for the artificial data, I generated (necessarily) one new sequence for each γ and m for sufficiently many γ and m. The noise level of the curves captures the sample variation very well.

Discussion. The results generally confirm the theory with few/small surprises.

The online estimators are plotted with solid lines. $\text{D}\vec{\text{i}}\text{r}\text{M}^*$ mostly coincides within ±10 nits with $S^{\vec{\beta}^*/2}$ for most m. Only when m approached n is $S^{\vec{\beta}^*/2}$ superior to $\text{D}\vec{\text{i}}\text{r}\text{M}^*$ due to renormalized weights leading to shorter $\text{CL}_w(\mathcal{A})$. Among the proper estimators, SAW-Bayes works best by a small margin, except for very small ($m\lesssim\ln n$) and very large ($m\approx n$) used alphabet and Zipf distributed data, but note that it is D (here 10 000 or 256) times slower than all the other algorithms. SSDC is virtually indistinguishable from Perks on the artificial data and only slightly better on the real data. Both perform poorly except for very small $m\lesssim\ln n$. Note that Perks performs as well as DirM^* (only) around $m\approx 2\ln\frac{n}{m}$, i.e. when their priors coincide. $\text{KT}_{\mathcal{X}}$ as well as $\text{Dir}\text{M}^{\boldsymbol{\alpha}}$ with any other fixed choice of $\boldsymbol{\alpha}$ perform very badly, especially for small m. $\text{KT}_{\mathcal{X}}$ performs well only for $m\approx D$ and for $m\approx 0.9n$ when $\beta^*\!/2$ is accidentally close to $\alpha_+=D/2$.

The offline estimators (densely dashed lines), DirM*, $S^{\beta\prime/2}$ with constant optimal parameters $\boldsymbol{\alpha}^*$ and β^* mostly coincide within ± 10 nits with their variable $\vec{\boldsymbol{\alpha}}^*$ and $\vec{\beta}^*$ online versions, except for very large m they are slightly better. This shows that making them online is essentially for free, which is consistent with the close bounds for small m in both cases. This has been observed for other offline-online algorithm pairs as well [HP05]. There is very little gain in knowing $\boldsymbol{\alpha}^*$ or β^* in advance. As expected off-line $\mathrm{KT}_{\mathcal{A}}$+$\ln\binom{D}{m}$ significantly improves upon $\mathrm{KT}_{\mathcal{X}}$ for small m and even beats $S^{\vec{\beta}^*/2}$ by a couple of bits for sufficiently small m, but breaks down for medium and large m, and anyway is off-line.

These observations are rather consistent across uniform, Zipf, and real data. Only for Zipf data, SAW-Bayes and $\mathrm{KT}_{\mathcal{A}}$+$\ln\binom{D}{m}$ seem to be worse, and the relative performance of many estimators on b&w fax *pic* is reversed.

The oracle estimators (dotted lines) possess significant extra knowledge: $\mathrm{KT}_{\mathcal{A}}$-Oracle the used alphabet $\mathcal{A}$, and LL$\boldsymbol{\theta}$-Oracle and H-Oracle even the counts $\boldsymbol{n}$. The plots show the magnitude of this extra knowledge.

Summary. Results are similar for other (n,D,m) and (n,D,γ) combinations but code length differences can be more or less pronounced but are seldom reversed. In short, $\mathrm{KT}_{\mathcal{X}}$ performs very poorly unless $m \approx D$, and Perks and SSDC perform poorly unless $m \lesssim \ln n$; $\mathrm{KT}_{\mathcal{A}}$+$\ln\binom{D}{m}$, DirM*, $S^{\beta\prime/2}$ are not online; the oracles LL$\boldsymbol{\theta}$-Oracle, H-Oracle, $\mathrm{KT}_{\mathcal{A}}$-Oracle are not realizable; and SAW-Bayes is extremely slow; which leaves $\vec{\text{Dir}}$M* and $S^{\vec{\beta}^*/2}$ as winners. They perform very similar unless m gets very close to $\min\{n,D\}$ in which case $S^{\vec{\beta}^*/2}$ wins.

J List of Notation

Symbol	Explanation
$\mathcal{X}$	total (large) base alphabet of size D
$D=\lvert\mathcal{X}\rvert$	size of (large) base alphabet $\mathcal{X}$
n	sequence length
$x_{1:n}$	total sequence
n_i	number of times i appears in $x_{1:n}$
$\mathcal{A}\subseteq\mathcal{X}$	symbols actually appearing in sequence $x_{1:n}$
$m=\lvert\mathcal{A}\rvert$	size of alphabet used in $x_{1:n}$
i,j,k	indices ranging over symbols in $\mathcal{X}$, $\mathcal{A}$, $\mathcal{X}\setminus\mathcal{A}$ respectively
$\bar{\nu}:=\frac{n}{m},\nu:=\frac{m}{n}$	average multiplicity of symbols and its inverse
t	current time ranging from 0 to $n-1$
$x_{1:t}$	sequence seen so far
$\mathcal{A}_t$	$=\{x_1,...,x_t\}$ = symbols seen so far
$m_t=\lvert\mathcal{A}_t\rvert$	number of different symbols observed so far (in $x_{1:t}$)
x_{t+1}	next symbol to be predicted

n_i^t	number of times i appears in $x_{1:t}$
$\mathcal{N}ew$	set of t for which x_{t+1} is new, i.e. $x_{t+1} \notin \mathcal{A}_t$
$\mathcal{O}ld$	set of t for which x_{t+1} is old, i.e. $x_{t+1} \in \mathcal{A}_t$
P,Q	probability over sequences
$P_{\text{name}}^{\text{param}}$	parameterized and named probability
$R_{\text{name}}^{\text{param}}$	$= -\ln P_{\text{name}}^{\text{param}} - n \cdot H(\boldsymbol{n}/n)$ = redundancy of $P_{\text{name}}^{\text{param}}$
$\overline{R}$,$\underline{R}$	upper/lower bound on redundancy
CL	code length in nits
θ_i	probability that $x_t = i$
α_i,α_+	Dirichlet parameters and their sum
$\beta = \beta_n$,β_t	general (constant,variable) parameter β
$\beta^* \neq \beta_n^*$,β_t^*	optimal (constant,variable) parameter β
w_i^t	weight of new symbol i at time t
ln	Natural logaritm. Results are in 'nits'
$\boldsymbol{v}$	vector over alphabet $\mathcal{X}$
$\vec{v}$	vector over time $t = 0...n-1$
Γ,Ψ	Gamma and diGamma function
c	constant > 0 and $< \infty$