Abstract

Compression and Predictive Distributions for Large Alphabets

Xiao Yang

2015

Data generated from large alphabet exist almost everywhere in our life, for example, texts, images and videos. Traditional universal compression algorithms mostly involve small alphabets and assume implicitly an asymptotic condition under which the extra bits induced in the compression process vanishes as an infinite number of data come. In this thesis, we put the focus on compression and prediction for large alphabets with the alphabet size comparable or larger than the sample size.

We first consider sequences of random variables independent and identically generated from a large alphabet. In particular, the size of the sample is allowed to be variable. A product distribution based on Poisson sampling and tilting is proposed as the coding distribution, which highly simplifies the implementation and analysis through independence. Moreover, we characterize the behavior of the coding distribution through a condition on the tail sum of the ordered counts, and apply it to sequences satisfying this condition. Further, we apply this method to envelope classes. This coding distribution provides a convenient method to approximately compute the Shtarkov's normalized maximum likelihood (NML) distribution. And the extra price paid for this convenience is small compared to the total cost. Furthermore, we find this coding distribution can also be used to calculate the NML distribution by a Monte Carlo method with no extra price. And this calculation remains simple due to the independence of the coding distribution.

Further, we consider a more realistic class – the Markov class, and in particular, tree sources. A context tree based algorithm is designed to describe the dependencies among the contexts. It is a greedy algorithm which seeks for the greatest savings in codelength when constructing the tree. Compression and prediction of individual counts associated with the contexts uses the same coding distribution as in the i.i.d case. Combining these two procedures, we demonstrate a compression algorithm based on the tree model.

Results of simulation and real data experiments for both the i.i.d model and Markov model have been included to illustrate the performance of the proposed algorithm.

Compression and Predictive Distributions for Large Alphabets

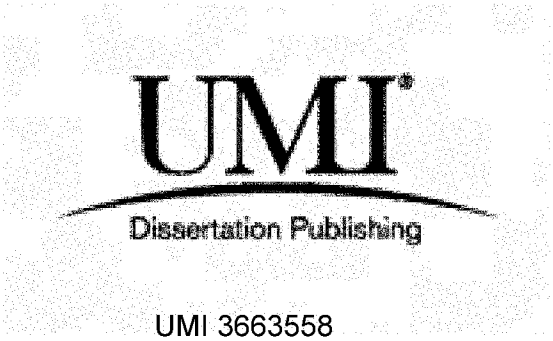
A Dissertation
Presented to the Faculty of the Graduate School
of
Yale University
in Candidacy for the Degree of
Doctor of Philosophy

by

Xiao Yang

Dissertation Director: Andrew R Barron

May 2015



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Contents

List of Figures

Acknowledgements

The past six years is a long journey, but I am very happy I made it. At the end of this trip, I want to sincerely thank my advisor Prof.Andrew Barron. He guided me onto this challenging and interesting path, and gave me endless patience and support. I used to think he is a genius. Since it looks like he knows everything in statistics and can always find the crux, but he doesn't seem to research a lot (mostly because he has other responsibilities like teaching and advising students). Later on, I heard a story about him. It said when he was a graduate student, he read tons of papers and sometimes slept in the office. This is not a unique story, but it told me that the distance between "genius" and me is this determination and commitment. Besides dedication to research, his generous and positive personality also influenced me a lot. He never hesitates to share ideas or to compliment others. I still remember once I had some small progress in my research, he was so happy that he gave me a high five. I am extremely grateful to have him as my advisor.

I also wish to thank my committee Prof. Joseph Chang and Prof. Mokshay Madiman. One thing I learn from Joe is no matter how complex one thing is, he can always explain it in a simple and intuitive manner. Only people who truly capture the essence and turn into their own understandings can do this. To me, it is a magic ability. Mokshay also gave me strong support. He never said no when I requested to meet him even when he was very busy. Because of their presence, I am able to reach the destination.

Moreover, I would like to thank Prof. Peter Jones, Prof. Jun'ichi Takeuchi, Prof. Wojciech Szpankowski, Prof. Narayana Santhanam and Prof. Teemu Roos. They

gave me great encouragement and tremendous help. My minds and my life would be much more restricted without them.

Every now and then, I am thinking I am so lucky to have come to Yale, and have met those friends and spent these happy and fruitful years. This will become my lifetime treasure.

Thank my parents for giving me life and bringing me up. Without their unconditional encouragement and support, I would never have the chance to see what I have seen, to know what I know and to experience what I have experienced.

Last but not least, I would like to thank my fiancé, Lei. Thank him for finding me and loving my defects. Because of him, my everyday memory for the past few years became different.

Chapter 1

Introduction

Today, more than 1 out of 2 phones used by American people are smart phones, and more than 1/4 photos are taken by smartphone cameras. Technology companies detect and create human needs that people didn't realize before the products come. The huge amount of data that each smart device or website generates, and the practical requirement of making things smaller and lighter have posed new challenges to the task of data compression.

Many such problems are of large alphabet in nature, which means the alphabet size is comparable or even larger than the sample size. Examples include Chinese text on the character basis, or DNA sequences. Traditional data compression techniques mostly focus on small alphabets and propose algorithms that work in an asymptotic setup. In recent years, large alphabet problems began to catch people's attention.

This dissertation concerns mainly large alphabet compression and prediction, with focus on compressing, describing and predicting data in a simple and efficient way.

1.1 Universal compression

Data generated by a probability distribution can be compressed almost to its entropy according to Shannon. The probability distribution P generating the data assigns the optimal length of codewords $\log 1/P(x)$ to each symbol x in the alphabet $\mathcal{A}$. So when the true distribution is know, it can readily be used to compress the data. However, this rarely happens in practice. Usually, one assumes the generating distribution comes from a class of distributions $\mathcal{P}$, and universal compression aims to compress the data well no matter which distribution in $\mathcal{P}$ the data are generated from.

Each encoding scheme Q corresponds to a probability distribution over the alphabet. Suppose a sequence of data $X^n = (X_1, X_2, \cdots, X_n)$ is generated from a

distribution P on an alphabet $\mathcal{A}$. An encoding procedure Q is a (sub)probability distribution on $\mathcal{A}^n$ which assigns probability $Q(X^n)$ to each string X^n and produces a binary string of length $\log 1/Q(X^n)$ (we do not worry about the integer constraint). Ideally the true probability distribution $P(X^n)$ could be used, as it produces no extra bits for coding purpose. The *regret* induced by using Q instead of P is

$$R(Q, P, X^n) = \log \frac{1}{Q(X^n)} - \log \frac{1}{P(X^n)},$$

where log is logarithm base 2. Likewise, the *expected regret* is

$$r(Q, P) = \mathbf{E}_P \left(\log \frac{1}{Q(X^n)} - \frac{1}{P(X^n)} \right).$$

In universal coding the expected regret is also called the *redundancy.*

In the pointwise regret story, the set of codelengths $\log(1/P(X^n))$ provides a standard with which our encoding scheme can be compared. Given the family $\mathcal{P}$, consider the best candidate with hindsight $\hat{P}$, which achieves the maximum value, $\hat{P}(X^n) = \max_{P \in \mathcal{P}}(P(X^n))$ (corresponding to $\min_{P \in \mathcal{P}} \log(1/P(X^n))$), and compare it to our strategy $Q(X^n)$.

Then the problem becomes: given the family $\mathcal{P}$, how to choose Q to minimize the maximized regret

$$\min_Q \max_{X^n \in \mathcal{A}^n} R(Q, P, X^n) = \min_Q \max_{X^n \in \mathcal{A}^n} \log \frac{\hat{P}(X^n)}{Q(X^n)},$$

or the redundancy,

$$\min_Q \max_{P \in \mathcal{P}} r(Q, P) = \min_Q \max_{P \in \mathcal{P}} \mathbf{E}_P \log \frac{P(X^n)}{Q(X^n)}.$$

1.2 Normalized maximum likelihood distribution

It is shown by Shtarkov that the normalized maximum likelihood (NML) distribution

$$Q_{NML}(X^n) = \frac{\hat{P}(X^n)}{\sum_{X^n} \hat{P}(X^n)}$$

is the unique pointwise minimax strategy [1]. And the minimax regret for the class $\mathcal{P}$ is $\sum_{X^n} \hat{P}(X^n)$ known as Shtarkov's sum.

The normalized maximum likelihood distribution and Shtarkov's sum play an essential role in universal compression. The NML distribution is the unique minimax strategy. But it is horizon dependent and computationally expensive. Many strategies try to approximate the NML strategy. For example, the posterior update rule with respect to a $Dirichlet(1/2, \ldots, 1/2)$ prior (also called the Krichevsky-Trofimov sequential coding rule) has been studied in an asymptotic setting where the sample size goes to infinity while the alphabet size is held fixed. In recent years, strategies for large alphabet are being considered, for example, for envelope classes [2][3]. The NML distribution still provides a direct method to calculate the minimax regret and a target minimax distribution.

Chapter 2

i.i.d model

Submitted to *IEEE Transactions on Information Theory* as Xiao Yang and Andrew Barron (2013), Large Alphabet Compression and Predictive Distributions through Poissonization and Tilting

2.1 Introduction

Large alphabet compression and prediction problems concern understanding the probabilistic scheme of a huge number of possible outcomes. In many cases the ordered probability of individual outcomes displays a quickly falling shape, with a small number of outcomes happening most often. An example is Chinese characters. A dictionary [4] contains 85568 Chinese characters in total [5], but the number of frequent characters is considerably smaller. Here we consider an i.i.d model for this problem. Despite the possible dependence among the symbols in the alphabet as in language, it serves as a start and can be extended to models taking dependence into account. Some efforts in investigating alphabet of symbols with dependencies are included in [6].

Previous theoretical analysis usually assumes the length of a message is known in advance when it is coded. This is not always true in practice. Serialization writers do not know how many words a novel contains exactly before he finishes the last sentence. Nevertheless, given a limited time or space, one could possibly guess how many words on average can be accommodated.

Suppose a string of random variables $\underline{X} = (X_1, \ldots, X_N)$ is generated independently from a discrete alphabet $\mathcal{A}$ of size m. We allow the string length N to be variable. A special case is when N is given as a fixed number, or it can be random. In either case, $\underline{X}$ is a member of the set $\mathcal{X}^*$ of all finite length strings

$$\begin{aligned} \mathcal{X}^* &= \bigcup_{n=0}^{\infty} \mathcal{X}^n \\ &= \bigcup_{n=0}^{\infty} \{x^n = (x_1, \ldots, x_n) : x_i \in \mathcal{A}, i = 1, \ldots, n\}. \end{aligned}$$

Our goal is to code/predict the string $\underline{X}$. Note that the length N is determined

by the string. There will be an agreed upon distribution of N, perhaps Poisson or deterministic.

Now suppose given N, each random variable X_i is generated independently according to a probability mass function in a parametric family $\mathcal{P}_\Theta = \{P_{\underline{\theta}}(x) : \underline{\theta} \in \Theta \subset R^m\}$ on $\mathcal{A}$. Thus

$$P_{\underline{\theta}}(X_1, \ldots, X_N | N = n) = \prod_{i=1}^{n} P_{\underline{\theta}}(X_i)$$

for $n = 1, 2, \ldots$ Of particular interest is the class of all distributions with $P_{\underline{\theta}}(j) = \theta_j$ parameterized by the simplex $\Theta = \{\underline{\theta} = (\theta_1, \ldots, \theta_m) : \theta_j \geq 0, \sum_{j=1}^{m} \theta_j = 1, j = 1, \ldots, m\}$.

As is familiar in universal coding, the normalized maximum likelihood (NML) distribution defined as $Q^*_{nml}(\underline{X}|N = n) = \max_{\underline{\theta} \in \Theta} P_{\underline{\theta}}(\underline{X}|N = n)/C^*_{m,n}$ provides the unique pointwise minimax strategy when the value $C^*_{m,n} = \sum_{\underline{X}} \max_{\underline{\theta} \in \Theta} P_{\underline{\theta}}(\underline{X}|N = n)$ is finite, and $\log C^*_{m,n}$ is the minimax regret. Coding and prediction of sequences of random variables usually involves computing conditionals of $X_{i+1}|X_1, \ldots, X_i$ as consecutive ratios of its marginals [1][7]. This task is generally hard since the marginalization requires a sum of order m^n, which appears to take exponential time in n. A linear time algorithm (in n) for computing the NML is proposed in [8], but it is not practically usable when the alphabet size m is large. Bayes-like representation of NML has been found which makes possible an easy computation of NML, but only moderate size m is computationally feasible at this moment [9]. Alternatively, one can use the Krichevsky-Trofimov's method [10], which is the mixture with respect to the *Dirichlet*$(1/2, \ldots, 1/2)$ prior, to approximate the NML distribution. But whether it has near minimax regret property is unknown for large m. In this paper, we will overcome this difficulty by applying two tools: one is the factorization of

the coding distribution of the string into a product of the distribution of the counts and the string given the counts. The distribution of the latter is uniform due to the sufficiency of the counts. The other is a tilted Stirling ratio distribution which we introduce here to simplify the encoding of the counts as discussed later.

Let $\underline{N} = (N_1, \ldots, N_m)$ denote the vector of counts for symbol $1, \ldots, m$. The observed sample size N is the sum of the counts $N = \sum_{j=1}^{m} N_j$. Both $P_{\underline{\theta}}(\underline{X})$ and $P_{\underline{\theta}}(\underline{X}|N = n)$ have factorizations based on the distribution of the counts

$$P_{\underline{\theta}}(\underline{X}|N=n) = P(\underline{X}|\underline{N})\, P_{\underline{\theta}}(\underline{N}|N=n),$$

and

$$P_{\underline{\theta}}(\underline{X}) \;\; = \;\; P(\underline{X}|\underline{N})\, P_{\underline{\theta}}(\underline{N}).$$

The first factor of the two equations is the uniform distribution on the set of strings with given counts, which does not depend on $\underline{\theta}$. The vector of counts $\underline{N}$ forms a sufficient statistic for $\underline{\theta}$. Modeling the distribution of the counts is essential for forming codes and predictions. In the particular case of all i.i.d. distributions parameterized by the simplex, the distribution $P_{\underline{\theta}}(\underline{N}|N = n)$ is the $multinomial(n, \underline{\theta})$ distribution.

In the above, there is a need for a distribution of the total count N. Of particular interest is the case that the total count is taken to be $Poisson$, because then the resulting distribution of individual counts makes them independent [11].

Accordingly, we give particular attention to the target family $\mathcal{P}_{\Lambda}^{m} = \{P_{\underline{\lambda}}(\underline{N}) : \lambda_j \geq 0,\ j = 1, \ldots, m\}$, in which $P_{\underline{\lambda}}(\underline{N})$ is the product of $Poisson(\lambda_j)$ distribution for $N_j, j = 1, \ldots, m$. It makes the total count $N \sim Poisson(\lambda_{sum})$ with $\lambda_{sum} = \sum_{j=1}^{m} \lambda_j$ and yields the $multinomial(n, \underline{\theta})$ distribution by conditioning on $N = n$, where

$\theta_j = \lambda_j/\lambda_{sum}$. And the induced distribution on $\underline{X}$ is

$$P_{\underline{\lambda}}(\underline{X}) = P(\underline{X}|\underline{N})P_{\underline{\lambda}}(\underline{N}).$$

The task of coding a string is equivalent to providing a probabilistic scheme. A coder Q for the string is also a (sub)probability distribution on $\mathcal{X}^*$ which assigns a probability $Q(\underline{X})$ to each string $\underline{X}$ and produces a binary string of length $\log 1/Q(\underline{X})$ (we do not worry about the integer constraint). Ideally the true probability distribution $P_{\lambda}(\underline{X})$ could be used if $\underline{\lambda}$ were known, as it produces no extra bits for coding purpose. The *regret* induced by using Q instead of $P_{\underline{\lambda}}$ is

$$R(Q, P_{\underline{\lambda}}, \underline{X}) = \log \frac{1}{Q(\underline{X})} - \log \frac{1}{P_{\underline{\lambda}}(\underline{X})},$$

where log is logarithm base 2. Likewise, the *expected regret* is

$$r(Q, P_{\underline{\lambda}}) = \mathbf{E}_{P_{\underline{\lambda}}} \log \left(\frac{1}{Q(\underline{X})} - \frac{1}{P_{\underline{\lambda}}(\underline{X})} \right).$$

In universal coding the expected regret is also called the *redundancy*. Those quantities also arises as cumulative prediction loss in prediction problems as discussed in Section 2.3.8.

Here we can construct Q by choosing a probability distribution for the counts and then use the uniform distribution for the distribution of strings given the counts, written as P_{unif}. That is

$$Q(\underline{X}) = P_{unif}(\underline{X}|\underline{N})Q(\underline{N}).$$

Then the regret becomes the log ratio of the counts probability

$$\begin{aligned} R(Q, P_{\underline{\lambda}}, \underline{X}) &= \log \frac{P_{\underline{\lambda}}(\underline{N})}{Q(\underline{N})} \\ &= R(Q, P_{\underline{\lambda}}, \underline{N}). \end{aligned}$$

And the redundancy becomes

$$r(Q, P_{\underline{\lambda}}) = \mathbf{E}_{P_{\underline{\lambda}}} \log \frac{P_{\underline{\lambda}}(\underline{N})}{Q(\underline{N})}.$$

In the pointwise regret story, the set of codelengths $\log(1/P_{\underline{\lambda}}(\underline{X}))$ provides a standard with which our coder is to be compared. Given the family $\mathcal{P}_{\Lambda}^{m}$, consider the best candidate with hindsight $P_{\hat{\underline{\lambda}}}(\underline{X})$, which achieves the maximum value, $P_{\hat{\underline{\lambda}}}(\underline{X}) = \max_{\underline{\lambda} \in \Lambda}(P_{\underline{\lambda}}(\underline{X}))$ (corresponding to $\min_{\underline{\lambda} \in \Lambda} \log(1/P_{\underline{\lambda}}(\underline{X}))$), where $\hat{\underline{\lambda}}$ is the maximum likelihood estimator of $\underline{\lambda}$, and compare it to our strategy $Q(\underline{X})$. The maximization is equivalent to maximizing $\underline{\lambda}$ for the count probability, as the uniform distribution does not depend on λ, i.e.

$$\begin{aligned} \max_{\underline{\lambda} \in \Lambda}(P_{\underline{\lambda}}(\underline{X})) &= P_{unif}(\underline{X}|\underline{N}) \max_{\underline{\lambda} \in \Lambda} P_{\underline{\lambda}}(\underline{N}) \\ &= P_{unif}(\underline{X}|\underline{N})\, P_{\hat{\underline{\lambda}}}(\underline{N}). \end{aligned}$$

Moreover, the maximum likelihood estimate is $\hat{\underline{\lambda}} = \underline{N}$. Then the problem becomes: given the family $\mathcal{P}_{\Lambda}^{m}$, how to choose Q to minimize the maximized regret

$$\min_{Q} \max_{\underline{X}} R(Q, P_{\hat{\underline{\lambda}}}, \underline{X}) = \min_{Q} \max_{\underline{N}} \log \frac{P_{\hat{\underline{\lambda}}}(\underline{N})}{Q(\underline{N})},$$

or the redundancy,

$$\min_Q \max_{P_{\underline{\lambda}} \in \mathcal{P}_\Lambda^m} r(Q, P_{\underline{\lambda}}) = \min_Q \max_{P_{\underline{\lambda}} \in \mathcal{P}_\Lambda^m} \mathbf{E}_{P_{\underline{\lambda}}} \log \frac{P_{\underline{\lambda}}(\underline{N})}{Q(\underline{N})}.$$

For the regret, the maximum can be restricted to a set of counts instead of the whole space. A traditional choice being $S_{m,n} = \{(N_1, \ldots, N_m) : \sum_{j=1}^m N_j = n, N_j \geq 0, j = 1, \ldots, m\}$ associated with a given sample size n, in which case the minimax regret is

$$\min_Q \max_{\underline{N} \in S_{m,n}} \log \frac{P_{\hat{\underline{\lambda}}}(\underline{N})}{Q(\underline{N})}.$$

The normalized maximum likelihood distribution

$$Q_{nml}(\underline{N}) = \frac{P_{\hat{\underline{\lambda}}}(\underline{N})}{C(S_{m,n})} \mathbf{1}_{\{\underline{N} \in S_{m,n}\}}$$

provides the unique pointwise minimax strategy for coding and predicting the counts given $C(S_{m,n}) = \sum_{\underline{N} \in S_{m,n}} P_{\hat{\underline{\lambda}}}(\underline{N})$ being finite in accordance with [1]. Again, we have $\log C(S_{m,n})$ as the minimax regret.

We will introduce a slightly suboptimal coding distribution that makes the counts independent and show that it is nearly optimal for every $S_{m,n'}$ with n' not too different from a target n. Indeed, we advocate that our simple coding distribution is preferable to use computationally when m is large even if the sample size n were known in advance.

To produce our desired coding distribution we make use of some basic principles. One is that the multinomial family of distributions on counts matches the conditional distribution of $N_1, \ldots, N_m$ given the sum N when unconditionally the counts are independent Poisson. Another is the information theory principle [12][13][14] that the conditional distribution given a sum (or average) of a large number of independent

random variables is approximately a product of distributions, each of which is the one closest in relative entropy to the unconditional distribution subject to an expectation constraint. This minimum relative entropy distribution is an exponential tilting of the unconditional distribution.

In the Poisson family with distribution $\lambda_j^{N_j} e^{-\lambda_j}/N_j!$, exponential tilting (multiplying by the factor e^{-aN_j}) preserves the Poisson family (with the parameter scaled to $\lambda_j e^{-a}$). Those distributions continue to correspond to the multinomial distribution (with parameters $\theta_j = \lambda_j/\lambda_{sum}$) when conditioning on the sum of counts N. A particular choice of $a = \ln(\lambda_{sum}/N)$ provides the product of Poisson distributions closest to the multinomial in regret. Here for universal coding, we find the tilting of individual maximized likelihood that makes the product of such closest to the Shtarkov's NML distribution. This greatly simplifies the task of approximate optimal universal compression and the analysis of its regret.

Indeed, applying the maximum likelihood step to a Poisson count k produces a maximized likelihood value of $M(k) = k^k e^{-k}/k!$. We call this maximized likelihood the Stirling ratio, as it is the quantity that Stirling's approximation shows near $(2\pi k)^{-1/2}$ for k not too small. We find that this $M(k)$ plays a distinguished role in universal large alphabet compression, even for sequences with small counts k. This measure M has a product extension to counts $N_1, N_2, \ldots, N_m$,

$$M^m(\underline{N}) = M(N_1)M(N_2)\cdots M(N_m).$$

Although M has an infinite sum by itself, it is normalizable when tilted for every positive a. The tilted Stirling ratio distribution is

$$P_a(N_j) = \frac{N_j^{N_j} e^{-N_j}}{N_j!} \frac{e^{-aN_j}}{C_a}, \tag{2.1}$$

with the normalizer $C_a = \sum_{k=0}^{\infty} k^k e^{-(1+a)k}/k!$. Figure 2.1 illustrates how C_a decreases with respect to a. While it constitutes the main part of regret for small alphabet case, its value drops quickly as larger a is in use (corresponding to large m size), as demonstrated in Figure 2.1.

The coding distribution we propose and analyze is simply the product of those tilted one-dimensional maximized Poisson likelihood distributions for a value of a we will specify later

$$Q_a(\underline{N}) = P_a^m(\underline{N}) = P_a(N_1) \cdots P_a(N_m).$$

By allowing description of all possible counts $N_j \geq 0$, $j = 1, \ldots, m$, our codelength will be greater for some strings than codelengths designed for the case of a given sum $N = n$. Nevertheless, with N distributed $Poisson(n)$, the probability of the outcome $N = n$ is approximately $P(N = n) \approx 1/\sqrt{2\pi n}$. So the allowance of description of N (not just $N_1, \ldots, N_m$ given N) adds $\log 1/P(N = n)$ which is approximately $\frac{1}{2} \log 2\pi n$ bits to the description length beyond which would have been ideal $\log 1/Q_a(N_1, \ldots, N_m | N = n)$ if $N = n$ were known. This ideal codelength constructed from the tilted maximized Poisson, when conditioning on n, matches the Shtarkov's normalized maximum likelihood based on the multinomial. Thus, $Q_a(\underline{N})$ may also be used in construction of Shtarkov's NML distribution and its conditionals as explained in Section 2.3.7.

For small alphabet with $m << n$, the minimax regret is about $\frac{1}{2} \log n$ bits per free parameter (a total of $\frac{m-1}{2} \log n$ + constant); and for large alphabet when $m \sim n$ and $n = o(m)$, the minimax regret is about $O(n)$ and $n \log \frac{m}{n}$ respectively [1][7][15][16]. The additional $\frac{1}{2} \log n$ bits is a small price to pay for the sake of gaining the coding simplification and additional flexibility.

If it is known that the total count is n, then the regret is a simple function of n

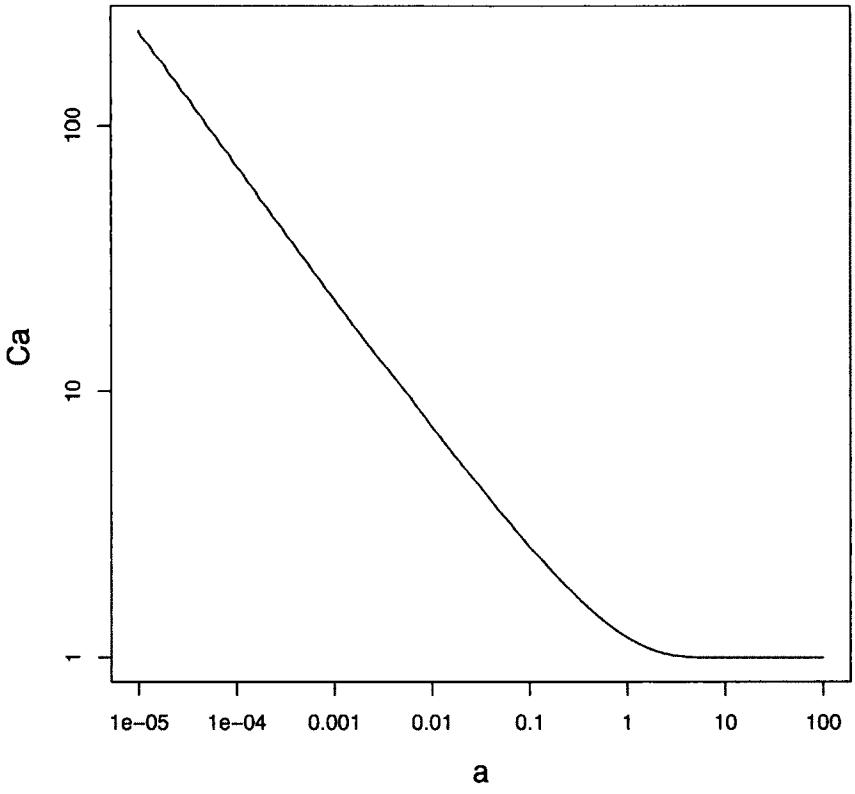


Figure 2.1: Relationship between a and C_a.

and the normalizer C_a. The choice of the tilting parameter a^* given by the moment condition $\mathbf{E}_{Q_a} \sum_{j=1}^{m} N_j = n$ minimizes the regret over all positive a. This arises by differentiation. because $\frac{\partial}{\partial a} \log C_a$ is equal to $-n/m \log e$. Moreover, a^* depends only on the ratio between the size of the alphabet and the total count m/n. Figure 2.2 displays a^* as a function of m/n solved numerically. These values can be stored. Given an alphabet with m symbols and a string generated from it of length n, one can look at the stored values and find the a^* desired according to the m/n given, and then use the a^* to do the encoding.

If, however, the total count N is not given, then the regret depends on N. We use a mixture of a to account for the lack of knowledge in advance, and details are discussed in section 2.3.4.

When a is small, the tilting of the maximized Poisson likelihood distributions does not have much effect except in the tail of the distribution. Over most of the range of count values k it follows the approximate power-law $1/k^{1/2}$ as we have indicated. Power-laws have been studied for count distributions and are shown to be related to Zipf's law for the sorted counts [17]. Our use of a distribution close to a power-law is not because a power-law is assumed to govern the data, but rather because of its near optimum regret properties within suitable sets of counts, demonstrated here for the class of all Poisson count distributions, from which we obtain also its near optimality for the class of all multinomial distributions on counts.

Shtarkov studied the universal data compression problem and identified the exact pointwise minimax strategy [1]. He showed the asymptotic minimax lower bound for the regret is $\frac{m-1}{2} \log n + O(1)$, in which the parameter set Θ is the $m-1$ dimensional simplex of all probability vectors on an alphabet of size m. However, this strategy cannot be easily implemented for prediction or compression [1], because of the computational inconvenience of computing the normalizing constant, and because of

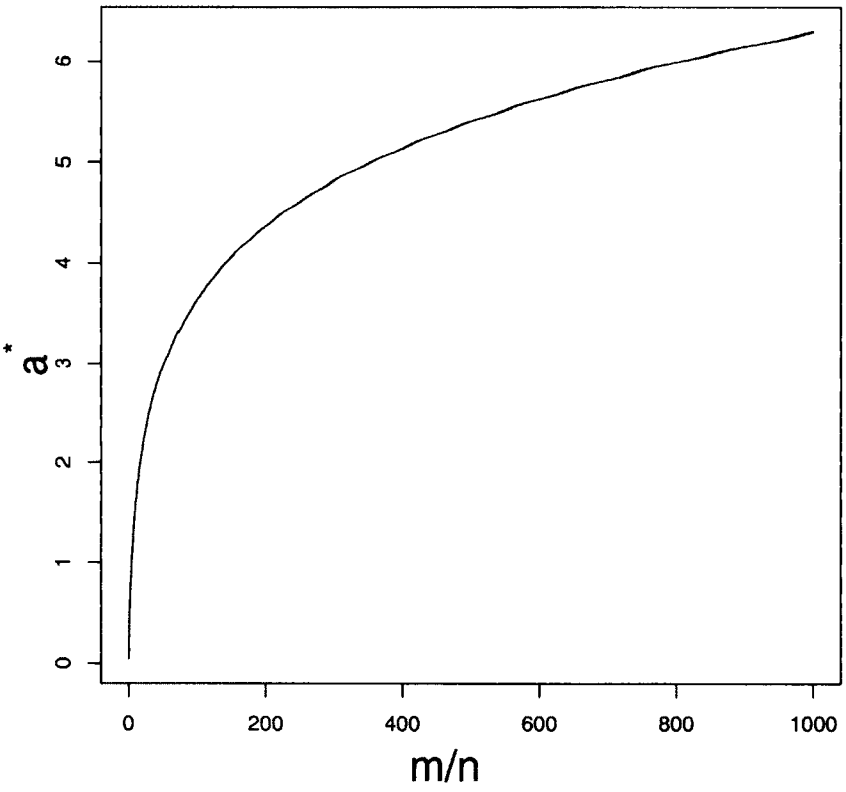


Figure 2.2: Relationship between a^* and $\frac{m}{n}$.

the difficulty in computing the successive conditionals required for implementation (by arithmetic coding). Let m^* be the number of different symbols that appear in a sequence. Shtarkov[18] also pointed out that when m is large, it is typical that m^* is much less than m, and the regret depends mainly on m^* rather than m. Xie and Barron[7][19] gave an asymptotic minimax strategy for coding under both the expected and pointwise regret for fixed size alphabet, which is formulated by a modification of the mixture density using Jeffery's prior. The asymptotic value of both the redundancy and the regret are of the form $\frac{m-1}{2}\log n + C_m + o(1)$, where C_m is a constant depending on m. Orlitsky and Santhanam[20] considered the problem in a large alphabet setting. They found the main terms in the minimax regret for $m = o(n)$, $m \sim n$ and $n = o(m)$ cases take the forms $\frac{m-1}{2}\log\frac{n}{m}$, $O(m)$ and $n\log\frac{m}{n}$ respectively. Szpankowski and Weinberger[16] provided more precise asymptotics in these settings. They also calculated the minimax regret of a source model in which some symbol probabilities are fixed. Boucheron, Garivier and Gassiat[2] focused on countably infinite alphabets with an envelope condition; they used an adapted strategy and gave upper and lower bounds for pointwise minimax regret. Later on Bontemps and Gassiat[3] worked on exponentially decreasing envelope class and provided a minimax strategy and the corresponding regret.

Other related work is in Good[21] who proposed the Good-Turing estimator for estimating the population frequency and the proportion of unseen symbols of ax large alphabet. Orlitsky and Santhanam[15] invented a notation "attenuation" as a way to evaluate and compare estimators, and their result showed that the good-turing estimator is superior to some common estimators, and they also proposed an estimator that is "better" than the good-turing estimator in the sense of "attenuation". Wagnerm, Viswanath, and Kulkarni[22] later pointed out that the good-turing estimator is not consistent in the "rare events regime", in which symbol probabilities

are of the order $O(\frac{1}{n})$, and they also constructed a consistent estimator based on the good-turing estimator. Orlitsky and santhanam[20] explored compression of "shape" and "pattern", which described the symbols' relative magnitude and precedence, respectively, of independent and identically distributed strings, and they showed that the maximum per-symbol shape regret is between 0.027 and 1, and the per-symbol pattern regret diminishes to zero for any alphabet size.

In this paper, we introduce a straightforward and easy to implement method for large alphabet coding. The purpose is three-fold: first, by allowing the sample size to be variable, we are considering a larger class of distributions. This is a less restrictive assumption than presuming a particular length. But the method can also be used for fixed sample size coding and prediction. In addition to simple near optimal compression for the class of all strings of a given length, our method also provides natural extension to the conclusion of [2] and [3].

Second, it unveils an information geometry of three key distributions/measures in the problem: the unnormalized maximum Poisson likelihood measure M^m of the counts, the conditional distribution M_{cond} of M^m given the total count equals n, which matches Shtarkov's normalized maximum multinomial likelihood distribution, and a tilted distribution Q_a, with the tilting parameter a chosen to make the expected total count equal to n. This tilted distribution Q_a minimizes the relative entropy from the original measure M^m within the class $\mathcal{C}$ of distributions with the moment condition $E[N] = n$. Hence, Q_a is the information projection of M^m onto $\mathcal{C}$. Moreover, since M_{cond} is also in $\mathcal{C}$, the Pythagorean-like equality holds [23][12], as verified also in Appendix A.3.

$$D(M_{cond}||M^m) = D(M_{cond}||Q_a) + D(Q_a||M^m). \tag{2.2}$$

The case of a tilted distribution (the information projection) as an approximating conditional distribution is investigated in [14] and [13]. A difference here is that our unconditional measure M^m is not normalizable.

Thirdly, the strategy designed through an independent Poisson model and tilting is much easier to analyze and compute as compared to the strategies based on multinomials. The convenience is gained through independence. To actually apply this two pass code, one could first describe the independent counts $N_1, \ldots, N_m$, for instance by arithmetic coding using $P_a(N_j)$, and then describe $X_1, \ldots, X_n$ given the count vector, by arithmetic coding using the sequence of conditional distributions for X_{i+1} given both $X_1, \ldots, X_i$ and all the counts (which is the sampling without replacement distribution, proportional to the counts of what remains after step i).

As a sufficient statistic, the counts N_j plays an important role in this compression and prediction problem. Coding and predicting the original data is the same as coding and predicting the counts, as the counts contain all the information that data embody about the parameter. Given the counts, the sequences follows a uniform distribution among all sequences with the given counts, and everyone agrees with how to deal with them. Here we model the count of each symbol as an independent random generation from a Poisson distribution, i.e., for each N_j, $j = 1, \ldots, m$, $N_j \sim Poisson(\lambda_j)$. This would induce infinite regret if we did not restrict the total counts, as the Poisson maximum likelihood measure sums to infinity. Luckily the tilting method offers a handy way to take account of the total count. By tilting the maximized likelihood value, the strategy can be designed for coding and predicting data generated from a large alphabet as independent variables. The expected and minimax regret can also be calculated.

This paper is organized in the following way. Section II introduces the model. Section III provides general results and outlines the proof. Section IV gives simulated

and real data examples. And details of proof are left in the appendix.

2.2 The Poisson Model

A Poisson model fits well into this problem. We have for each $j = 1, \ldots, m$,

$$N_j \sim Poisson(\lambda_j),$$

independently, and N also has a Poisson distribution

$$N \sim Poisson(\lambda_{sum}),$$

where $\lambda_{sum} = \sum_{j=1}^{m} \lambda_j$. Write $\underline{\lambda} = (\lambda_1, \ldots, \lambda_m)$, we have

$$P_{\underline{\lambda}}(\underline{X}) = P_{unif}(\underline{X}|\underline{N}) \prod_{j=1}^{m} P_{\lambda_j}(N_j).$$

We know that the MLE for each λ_j is $\hat{\lambda}_j = N_j$, and the first term is a uniform distribution which does not depend on $\underline{\lambda}$. So

$$P_{\hat{\underline{\lambda}}}(\underline{X}) = P_{unif}(\underline{X}|\underline{N}) \prod_{j=1}^{m} M(N_j).$$

where $M(k) = k^k e^{-k}/k!$, $k = 1, 2, \ldots$ (as given in the introduction) is the unnormalized maximized likelihood $M(N_j) = \max_{\lambda_j} P_{\lambda_j}(N_j)$.

If we use a distribution $Q(\underline{N})$ to code the counts, then the regret is

$$\log \frac{P_{\hat{\underline{\lambda}}}(\underline{X})}{P(\underline{X}|\underline{N})Q(\underline{N})} = \log \frac{\prod_{j=1}^{m} M(N_j)}{Q(\underline{N})}.$$

And the redundancy is

$$\mathbf{E}_{P_{\underline{\lambda}}} \log \frac{P(\underline{X}|\underline{\lambda})}{P(\underline{X}|\underline{N})Q(\underline{N})} = \mathbf{E}_{P_{\underline{\lambda}}} \log \frac{P(\underline{N}|\underline{\lambda})}{Q(\underline{N})}.$$

This method can also be applied to fixed total count scenario, which corresponds to the multinomial coding and prediction problem. Suppose $N = n$ is given, the Poisson model, when conditioned on $N = n$, indeed reduces to the i.i.d sampling model

$$P_{\underline{\lambda}}(X_1, \ldots, X_N | N = n) = P_{\underline{\theta}}(X_1, \ldots, X_n).$$

The right hand side is a discrete memoryless source distribution (i.i.d. $P_{\underline{\theta}}$) with probability specified by $P_{\underline{\theta}}(j) = \theta_j$, for $j = 1, \ldots, m$. Note that a sequence $X_1, \ldots, X_N$ with counts $N_1, \ldots, N_m$ of total $N = n$ satisfies

$$\begin{aligned} & P_{\underline{\lambda}}(X_1, \ldots, X_N | N = n) \\ = & \frac{P_{\underline{\lambda}}(X_1, \ldots, X_n)}{P_{\lambda_{sum}}(N = n)} \\ = & \frac{P_{unif}(X_1, \ldots, X_n | N_1, \ldots, N_m) P_{\underline{\lambda}}(N_1, \ldots, N_m)}{P_{\lambda_{sum}}(N = n)}. \end{aligned}$$

The question left is still how to model the counts. The maximized likelihood (the same target as used by Shtarkov) is thus expressible as

$$\begin{aligned} & P_{\hat{\underline{\lambda}}}(X_1, \ldots, X_N | N = n) \\ = & \frac{P_{unif}(X_1, \ldots, X_n | N_1, \ldots, N_m) \prod_{j=1}^{m} M(N_j)}{P_{\hat{\lambda}_{sum}}(N = n)}. \end{aligned}$$

Now again if we use $Q(N_1, \ldots, N_m)$ to code the counts, then the regret is

$$\begin{aligned} & \log \frac{P_{\hat{\underline{\lambda}}}(X_1, \ldots, X_N | N = n)}{P_{unif}(X_1, \ldots, X_n | N_1, \ldots, N_m) Q(N_1, \ldots, N_m)} \\ = & \log \frac{\prod_{j=1}^{m} M(N_j)}{P_{\hat{\lambda}_{sum}}(N = n) Q(N_1, \ldots, N_m)} \\ \approx & \frac{1}{2} \log 2\pi n + \log \frac{\prod_{j=1}^{m} M(N_j)}{Q(N_1, \ldots, N_m)} \end{aligned} \tag{2.3}$$

Here $\hat{\lambda}_{sum} = n$, hence the term $\frac{1}{2} \log 2\pi n$ is Stirling's approximation of $\log 1/P_{\hat{\lambda}_{sum}}(N = n)$ with a difference bounded by $\frac{1}{12n} \log e$ by the Robbin's refinement [24] of the Stirling's approximation. The $\frac{1}{2} \log 2\pi n$ arises because here Q includes description of the total N while the more restrictive target regards it as given.

2.3 Results

2.3.1 Regret

We start by looking at the performance of using independent tilted Stirling ratio distributions as a coding strategy, by examining the regret.

Let S be any set of counts, then the maximized regret of using Q as a coding strategy given a class $\mathcal{P}$ of distributions when the vector of counts is restricted to S is

$$R(Q, \mathcal{P}, S) = \max_{\underline{N} \in S} \log \frac{\max_{P \in \mathcal{P}} P(\underline{N})}{Q(\underline{N})}.$$

Theorem 2.1. *Let P_a be the distribution specified in equation (3.1) (Poisson maximized likelihood, tilted and normalized) and N denote the total count. The regret of using a product of tilted distributions $Q_a = \otimes_{j=1}^{m} P_a$ for a given vector of counts*

$\underline{N} = (N_1, \ldots, N_m)$ *is*

$$R\left(Q_a, \mathcal{P}_\Lambda^m, \underline{N}\right) = aN \log e + m \log C_a.$$

Let $S_{m,n}$ *be the set of count vectors with total count* n *be defined as before, then*

$$R\left(Q_a, \mathcal{P}_\Lambda^m, S_{m,n}\right) = an \log e + m \log C_a. \tag{2.4}$$

Let a^* *be the choice of* a *satisfying the following moment condition*

$$\mathbf{E}_{P_a} \sum_{j=1}^{m} N_j = m\, \mathbf{E}_{P_a} N_1 = n. \tag{2.5}$$

Then a^* *is the minimizer of the regret in expression (3.3). Write* $R_{m,n} = \min_a R(Q_a, \mathcal{P}_\Lambda^m, S_{m,n})$.

When $m = o(n)$, *the* $R_{m,n}$ *is near* $\frac{m}{2} \log \frac{ne}{m}$ *in the following sense.*

$$\begin{aligned} -d_1 \frac{m}{2} \log e &\leq R_{m,n} - \frac{m}{2} \log \frac{ne}{m} \\ &\leq m \log(1 + \sqrt{\frac{m}{n}}), \end{aligned} \tag{2.6}$$

where $d_1 = O\left((\frac{m}{n})^{1/3}\right)$.

When $n = o(m)$, *the* $R_{m,n}$ *is near* $n \log \frac{m}{ne}$ *in the following sense.*

$$\begin{aligned} m \log\left(1 + (1 - d_2)\frac{n}{m}\right) &\leq R_{m,n} - n \log \frac{m}{ne} \\ &\leq m \log\left(1 + \frac{n}{m} + d_3\right) \end{aligned} \tag{2.7}$$

where $d_2 = O(\frac{n}{m})$, *and* $d_3 = \frac{1}{2\sqrt{\pi}} \frac{n^2 e^2}{m(m-ne)}$.

When $n = bm$, *the* $R_{m,n} = cm$, *where the constant* $c = a^* b \log e + \log C_{a^*}$, *and* a^* *is such that* $\mathbf{E}_{P_a} N_1 = b$.

Proof. The expression of the regret is from the definition. The fact that a^* is the minimizer can be seen by taking partial derivative with respect to a of expression (3.3). The upper bounds are derived by applying Lemma A.1 in the appendix. Pick $a = m/2n$ and use the first inequality, we get the upper bound for $m = o(n)$ case; pick $a = \ln(m/ne)$ and use the second inequality, we have the upper bound for $n = o(m)$. Here ln is the logarithm base e. The rest of the proof is left in Appendix B. □

Remark 2.1. *The regret depends only on the number of parameters m, the total counts n and the tilting parameter a. The optimal tilting parameter is given by a simple moment condition in equation (3.4).*

Remark 2.2. *The regret $R_{m,n}$ is close to the minimax level in all three cases listed in Theorem 3.2. The main terms in the $m = o(n)$ and $n = o(m)$ cases are the same as the minimax regret given in [16] except the multiplier for $\log(ne/m)$ here is $m/2$ instead of $(m-1)/2$ for the small m scenario. For the $n = bm$ case, the $R_{m,n}$ is close to the minimax regret in [16] numerically.*

Remark 2.3. *In fact, the regret provides an upper bound for the redundancy. Recall that*

$$\begin{aligned} \mathbf{E}_{P_{\underline{\lambda}}} \log \frac{P_{\underline{\lambda}}}{Q_a} &\leq \mathbf{E}_{P_{\underline{\lambda}}} \max_{\underline{\lambda}} \log \frac{P_{\underline{\lambda}}}{Q_a} \\ &= a\lambda_{sum} \log e + m \log C_a. \end{aligned} \tag{2.8}$$

Theorem A.4 in Appendix A.4 gives more detailed expression of the redundancy for using Q_a. While there is a reduction of $(m/2)\log e$ bits as compared to the pointwise case, the error depends on the λ_j's. Nevertheless, expression (2.8) still provides an uniform upper bound for the redundancy for all possible Poisson means $\underline{\lambda}$ with a given sum.

Remark 2.4. *Simulation shows the value is very close to Szpankowski et al's approximation [16]. For example, with* $m = 70244$ *and* $n = 39161$ *(those are the* m *and* n *used in the simulation in Section 2.4, and* $\alpha = m/n = 1.79$*). The Szpankowski et al's approximation [16] of the minimax regret is* 64519.32*, and the regret we get from optimizing a grid of tilting parameters is* 64529.61*. Please see Figure 2.3 for the comparison of the two regret estimates.*

Corollary 1. *Let* $\mathcal{P}_\Theta^m$ *be a family of multinomial distributions with total count* n*. Then the maximized regret* $R(Q_a, \mathcal{P}_\Theta^m, S_{m,n})$ *has an upper bound within* $\frac{1}{2}\log 2\pi n + \frac{1}{12n}\log e$ *above the upper bound in Theorem 3.2.*

Proof. This can be easily seen by equation (2.3). □

2.3.2 Subset of sequences with partitioned counts

One advantage of using the tilted Stirling ratio distributions is the flexibility of choosing tilting parameters. As mentioned in the introduction, the ratio m/n uniquely determines the optimal tilting parameter. In fact, different tilting parameters can be used for symbols to adjust for their relative importance in the alphabet. Here we consider a situation in which the empirical distribution has most probability captured by a small portion of the symbols. This happens when the sorted probability list is quite skewed.

The following theorem holds for strings with constraints on the sum of tail counts $\sum_{j>L} N_j = nf$. Small remainder occurs in the following regret bound when $nf/(m-L)$ and $L/(n-nf)$ are both small.

Theorem 2.2. *Let* $S_{m,n,f,L}$ *be a subset of count vectors with the tail sum controlled by a value* $0 \le f \le 1$*, that is,* $S_{m,n,f,L} = \{\underline{N} = (N_1, \ldots, N_m) \colon \sum_{j=1}^m N_j = n, \sum_{j>L} N_j =$

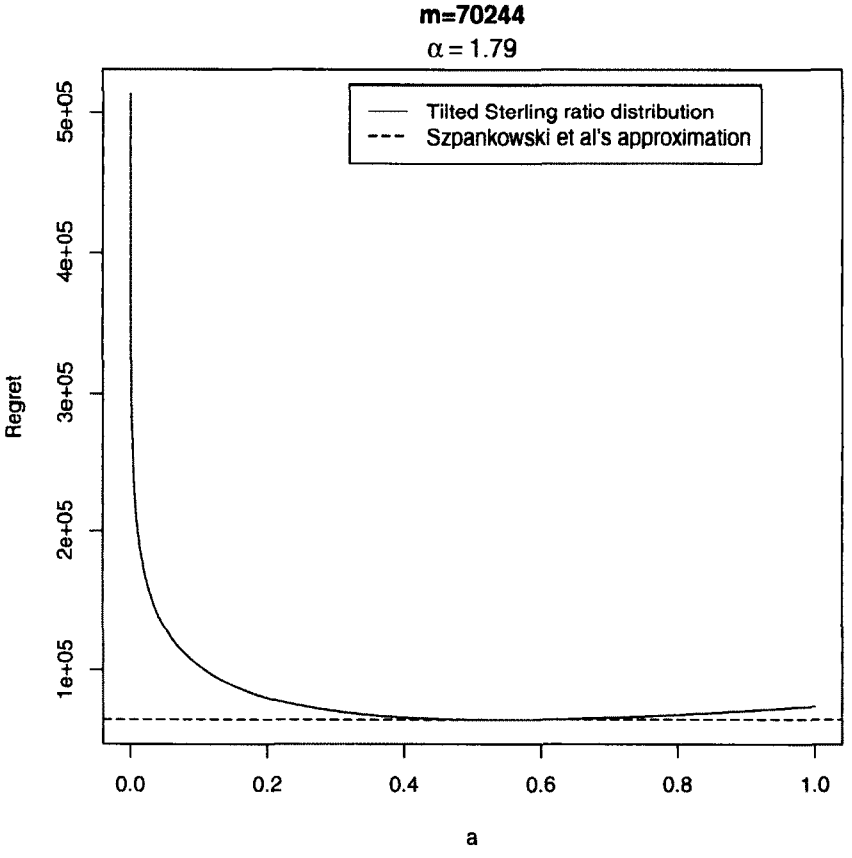


Figure 2.3: Regret for case $m \sim n$.

$nf\}$. *Here* L *is a number between* 0 *and* m. *The regret of using the tilted Stirling ratio distributions for count vectors in* $S_{m,n,f,L}$ *given each* $L \in \{0, \ldots, m\}$ *is mainly*

$$\frac{L}{2} \log \frac{(n-nf)e}{L} + nf \log \frac{(m-L)}{nfe}. \tag{2.9}$$

The remainder is bounded below by r_1 *and above by* r_2, *where*

$$r_1 = -d_1 \frac{L}{2} \log e + (m-L) \log \left(1 + (1-d_2) \frac{nf}{m-L}\right),$$

and

$$\begin{aligned} r_2 &= (m-L) \log \left(1 + \frac{nf}{m-L} + d_3\right) \\ &\quad + L \log \left(1 + \sqrt{\frac{L}{n-nf}}\right). \end{aligned}$$

Here d_1 *is* $O\left(\left(\frac{L}{n-nf}\right)^{1/3}\right)$ *and* d_2 *is* $O\left(\frac{nf}{m-L}\right)$ *and* $d_3 = \frac{1}{2\sqrt{\pi}} \frac{(nfe)^2}{(m-L)((m-L)-nfe)}$.

Proof. Consider the product distribution,

$$\begin{aligned} Q_{a,b}(\underline{N}) &= \prod_{j=1}^{m} P_{a,b}(N_j) \\ &= \prod_{j=1}^{m} \frac{N_j^{N_j} e^{-N_j}}{N_j!} \frac{e^{-aN_j} e^{-bN_j \mathbf{1}_{\{j>L\}}}}{C_{a,b,j}}, \end{aligned}$$

where $C_{a,b,j} = C_a$ if $j \leq L$, and $C_{a,b,j} = C_{a,b}$ is defined as $\sum_{k=0}^{\infty} k^k e^{-(1+a+b)k}/k!$ if $j > L$. It is in fact using an L dimensional product distribution Q_a on the first L symbols, and an $m-L$ dimensional product distribution Q_{a+b} on the rest.

The regret is the same for any $\underline{N} \in S_{m,n,f,L}$ given a and b. That is,

$$\begin{aligned} & R(Q_{a,b}, \mathcal{P}_\Lambda^m, S_{m,n,f,L}) \\ = \; & na\log e + L\log C_a + nfb\log e + (m-L)\log C_{a,b} \\ = \; & R(Q_a, \mathcal{P}_\Lambda^L, S_{L,n-nf}) + R(Q_{a+b}, \mathcal{P}_\Lambda^{m-L}, S_{m-L,nf}). \end{aligned}$$

Here $\mathcal{P}_\Lambda^j$ denotes the class of j independent Poisson distributions and $S_{j,k}$ is the set of j independent Poisson counts with sum equal to k. In the above case, $j = L$ or $m - L$, and $k = n - nf$ or nf.

The choice of a, b providing minimization of $R(Q_{a,b}, \mathcal{P}_\Lambda^m, S_{m,n,f,L})$ is given by the following conditions

$$\mathbf{E}_{P_{a,b}} \sum_{j=1}^{m} N_j = n$$

$$\mathbf{E}_{P_{a,b}} \sum_{j>L} N_j = nf.$$

This result can be derived by applying inequality (2.6) and inquequality (2.7) in Theorem 3.2 to $R(Q_a, \mathcal{P}_\Lambda^L, S_{L,n-nf})$ and
$R(Q_{a+b}, \mathcal{P}_\Lambda^{m-L}, S_{m-L,nf})$ respectively. □

Remark 2.5. *The problem here is treated as two separate coding tasks, one for a small alphabet with L symbols having a total count $n - nf$, and the other for a large alphabet with $m - L$ symbols with total count nf. The two main terms in expression (2.9) represent regret from coding the two subsets of symbols, with one set containing L symbols having relatively large counts, and each symbol induces $\frac{1}{2}\log\frac{n(1-f)e}{L}$ bits of regret, and the other containing the rest $m - L$ symbols with small counts and together cost $nf\log\frac{m}{nfe}$ extra bits.*

Remark 2.6. *We can arrange more flexibility in what the code can achieve by adding small additional pieces to the code. One is to adapt the choice of L between 0 and*

m, including $\log(m+1)$ more bits for the description of L. Next one can either work with the counts in the given order, or use an additional $\log \binom{m}{L}$ bits to describe the subset that has the L largest counts. Then one uses $\log 1/Q_{a,b}(\underline{N})$ bits to describe the counts. Rather than fixing f, one can work with the empirical tail fraction $\hat{f}(L)$, where $n\hat{f}(L)$ is the sum of the counts for the remaining $m - L$ symbols. Finally we can adapt the choices of a and b. A suggested method of doing so is described in Section 2.3.4, in which the $Q_{a,b}$ above is replaced by a mixture over a range of choices of a and b.

Remark 2.7. *The locking in of the tail sum to be a particular value in Theorem 2.2 seems rigid and unrealistic. However, the purpose of the theorem is an analytical tool rather than an application manual. To actually use the code, one could first describe a subset of size L using $\log \binom{m}{L}$ bits for an L between 0 and m, and then pick a and b according to an expected total count and tail behavior respectively, if there is any. In cases there is no such knowledge available beforehand, one could integrate over all tilted distributions and derive a mixture distribution which provides regret not too far away from the best tilted distribution, as will be discussed in Section 2.3.4.*

2.3.3 Envelope class

Besides a subset of strings, we can also consider subclass of distributions. Here we follow the definition of envelope class in [2]. Suppose $\mathcal{P}_{m,f}$ is a class of distributions on $1, \ldots, m$ with the symbol probability bounded above by an envelope function f, i.e.

$$\mathcal{P}_{m,f} = \{P_\theta : \theta_j \leq f(j), j = 1, \ldots, m\}.$$

Given the string length n, we know the count of each symbol follows a Poisson distribution with mean $\lambda_j = n\theta_j$, $j = 1, \ldots, m$. This transfers an envelope condition

from the multinomial distribution to a Poisson distribution, the mean for which is restricted to the following set

$$\Lambda_{m,f} = \{\underline{\lambda} : \lambda_j \leq nf(j), j = 1, \ldots, m\}.$$

Theorem 2.3. *The minimax regret of the Poisson class $\Lambda_{m,f}$ with envelope function f has the following upper bound*

$$\begin{aligned} & R(Q_a, \Lambda_{m,f}, \underline{N}) \\ \leq & \min_{L \in \{1,\ldots m\}} \frac{L}{2} \log \frac{n(1-\bar{F}(L))}{L} + n\bar{F}(L) \log e + r_3, \end{aligned}$$

where $\bar{F}(L) = \sum_{j>L} f(j)$, and

$$r_3 = \frac{L}{2(1-\bar{F}(L))} \log e + L \log \left(1 + \sqrt{\frac{L}{n(1-\bar{F}(L))}}\right).$$

Proof. A tilted distribution with $a = L/2n(1-\bar{F}(L))$ will give the result. Details are left in Appendix A.5. □

Remark 2.8. *Here in order for r_3 to be small, the tail sum of the envelope function $\bar{F}(L)$ needs to be small, although the upper bound holds for general envelope function f and L. This result is of the same order as the upper bound* $\inf_{L:L\leq n} \left((L-1)/2 \log n + n\bar{F}(L) \log e\right) + 2$ *given in [2]. The first main term in the bound given in Theorem 2.3 also matches the minimax regret given in [7] for an alphabet with L symbols and $n(1-\bar{F}(L))$ data points by Stirling's approximation,*

i.e.,

$$\begin{aligned} & \frac{L-1}{2}\log\frac{n(1-\bar{F}(L))}{2\pi}+\log\frac{\Gamma(1/2)^L}{\Gamma(L/2)} \\ \approx\ & \frac{L-1}{2}\log\frac{n(1-\bar{F}(L))e}{L}+\frac{1}{2}\log\frac{e}{2}. \end{aligned}$$

The extra $(1/2)\log(n(1-\bar{F}(L))e/L)$ *is because the tilted distribution allows* m *free parameters instead of* $m-1$.

Remark 2.9. *The best choice of tilting parameters for envelope class only depends on the envelope function and the number of symbols* L *constituting the 'frequent' subset. Unlike the subset of strings case discussed before, neither the order of the counts nor which symbols are those with largest counts matters, all we need is an envelope function decaying fast enough when the symbol probabilities are arranged in decreasing order so that* L *is a small integer and* $\bar{F}(L)$ *is also not big.*

2.3.4 Regret with unknown total count

We know that a^* depends on the value of the ratio $\eta = m/n$. However, when the total count is not known, we can use a mixture of tilted distributions $Q(\underline{N})$.

$$\begin{aligned} Q(\underline{N}) &= \int_0^{m/2} Q_a(\underline{N})\frac{1}{m/2}da \\ &= \int_0^{m/2}\prod_{j=1}^{m}\frac{N_j^{N_j}e^{-N_j}}{N_j!\,C_a}e^{-aN_j}\frac{2}{m}da \\ &\le M(\underline{N})\frac{2}{m}\int_0^{\infty}e^{-Nh(a)}da \end{aligned}$$

where $h(a) = a+\eta\log C_a$, with $\eta = m/N$. Here the upper end of the integrated area is due to Lemma A.2. We have $a^* \le m/(2n) \le m/2$.

For any realized non-negative total count $N = k$, the integrand is maximized at

a^*_η with $\eta = m/k$, defined as solution to the equation $\mathbf{E}_{P_a} N_1 = 1/\eta$. And the integral can be approximated by the Laplace method [25],

$$Q(\underline{N}) = \frac{2}{m}\left(\prod_{j=1}^{m}\frac{N_j^{N_j}e^{-N_j}}{N_j!}\right)e^{-kh(a^*_\eta)}\sqrt{\frac{2\pi}{ck}}\,(1+o(1))\,,$$

where $c = h''(a)|_{a=a^*_\eta}$. Note that the above approximation provides the leading term in an asymptotic expansion of $Q(\underline{N})$. Given η fixed, the leading term approaches the integral as k goes to infinity.

Hence, the regret induced by $Q(\underline{N})$ is

$$\log\frac{M(\underline{N})}{Q(\underline{N})} \approx k(a^*_\eta + \eta\log C_{a^*_\eta}) + \frac{1}{2}\log\frac{ck}{2\pi} + \log\frac{m}{2}.$$

The main part $k(a^*_\eta + \eta\log C_{a^*_\eta})$ is the answer form Theorem 3.2 if we had known the sample size k in advance. By definition,

$$h''(a) = \eta\frac{\partial^2}{\partial a^2}(\log C_a) = \eta Var_{P_a}(N_1),$$

since $\log C_a$ is the cumulant generating function of the tilted Stirling ratio distribution. We plot $V_a = \frac{\partial^2}{\partial a^2}(\log C_a)$ in Figure 2.4.

Here we use Laplace method to approximate the integral. It assumes the integral has a strict minimum over the integration region at an interior point. When $m \sim n$ ($\eta = m/n$ fixed), it approximates the leading order term in the asymptotic expansion of the integral. Evidence shows it's also applicable to larger η case, but more detailed analysis would need to be done to reach a conclusion. Here adopting Laplace's method serves mainly for revealing the characteristics of integrating with the Q_a's.

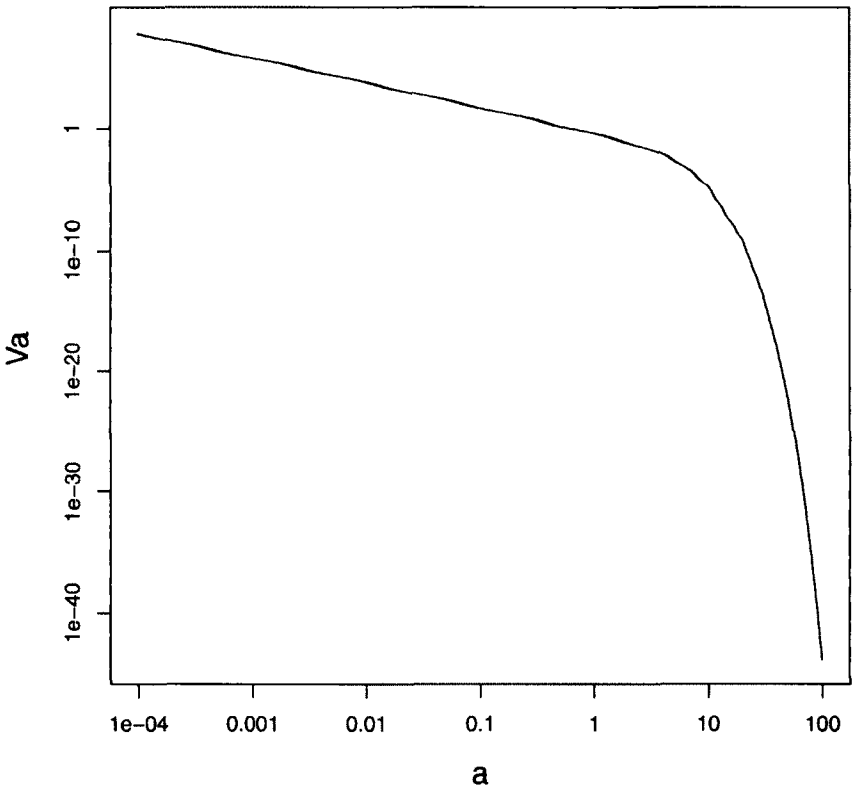


Figure 2.4: Relationship between a and V_a.

2.3.5 Conditional distributions induced by the tilted Stirling ratio distribution.

To account for strings of arbitrary length, our coding strategy Q_a assigns a probability distribution to all finite length strings. However, when considering strings of a known length, we are interested to see what the distribution looks like conditioning on a particular number n.

Let $\underline{N}^n$ denote any count vector in $S_{m,n}$, and N_x^n denote the x's component of $\underline{N}^n$, where $x \in \{1, \ldots, m\}$. Also, let M_{mul} be the *multinomial*$(n, \underline{\theta})$ maximized likelihood. We have

$$Q_a(\underline{N}^n | N = n) = \frac{Q_a(\underline{N}^n)}{Q_a(S_{m,n})} = \frac{M_{mul}(\underline{N}^n)}{M_{mul}(S_{m,n})}. \tag{2.10}$$

The conditioning of Q_a in expression (2.10) reduces the Poisson maximized likelihood (conditioned on the sum $N = n$) to be the same as the multinomial maximized likelihood normalized as indicated, which is indeed the Shtarkov's NML distribution for the multinomial family of distributions of counts.

This conditional distribution of counts, when multiplied by the uniform distribution of strings given the counts, induces a distribution on the strings, i.e.,

$$P_n(\underline{X}^n) = P_{unif}(\underline{X}^n | \underline{N}^n) Q_a(\underline{N}^n | N = n),$$

where $\underline{X}^n$ is the vector $X_1, \ldots, X_n$.

This sequence of distributions P_n are not compatible in the sense that the sum of the probability of $X_1, \ldots, X_n, X_{n+1} = x$ for $x \in \mathcal{A}$ under P_{n+1} does not sum to $P_n(X_1, \ldots, X_n)$, and hence do not have extensions to a stochastic process. To see

this incompatibility one looks at the sum

$$\sum_{x \in \mathcal{A}} P_{n+1}(X_1, \ldots, X_n, X_{n+1} = x)$$

and confirm it is not equal to $P_n(X_1, \ldots, X_n)$. This property is what is called the horizon dependence of NML [26]. For more details, please see Appendix B.1.

2.3.6 Computational simplicity

The coding distribution can be implemented by a two pass code. We first code the distributions of the counts by arithmetic coding using the tilted Stirling ratio distribution. This is an easy implementation since the coding distribution for the counts are independent. Then we could implement arithmetic coding again to code the string given the counts. The distribution of the string given the counts is uniform for all strings with the given counts. To implement arithmetic coding, one uses the conditional probability for x less than or equal to the observed X_{i+1} given its past and the counts, i.e.

$$P\left(X_{i+1} < x_{i+1} | X_1, \ldots, X_i, (N_1, \ldots, N_m)\right),$$

and

$$P\left(X_1, \ldots, X_i, X_{i+1} | (N_1, \ldots, N_m)\right),$$

for each $i = 0, \ldots, n-1$ with $n = \sum_{j=1}^{m} N_j$.

Indeed for $i = 1$, the $P(X_1 = x_1 | (N_1, \ldots, N_m)) = N_{x_1}/n$, and generally let $N_{j,i}$ be the count of the number of occurrence of j in $X_1, \ldots, X_i$, then the remaining counts are $N_{j,i}^{rem} = N_j - N_{j,i}$, and $P(X_{i+1} = x | X_1, \ldots, X_i, (N_1, \ldots, N_m)) = N_{j,i}^{rem}/(n-i)$. This is the consequence of the distribution of $X_1, \ldots, X_n$ given $N_1, \ldots, N_m$ being

uniform on the set of strings with these counts. (It is in accordance with the theory of sampling without replacement that arises with this conditioning.)

This two pass code makes possible a computationally feasible coding in the regime of $m \sim n$ and $n = o(m)$ as well as $m = o(n)$. Alternatively, the one pass Krichevsky–Trofimov [10] sequential coding rule, which is the Laplace posterior update rule with respect to the $Dirichlet(1/2, \ldots, 1/2)$ prior, can also be used for $m = o(n)$, but whether it has near minimax regret is unknown for large m. What we propose here is a simple scheme that achieves nearly minimal regret in all situations. And its implementation is simple due to the independence of the coding distribution of the counts. Computation complexity for the counts is $O(n(\log m + n \log n))$ for large m, and $n \log n$ for small m. Details are included in Appendix B.2. Indeed, we make the counts independent which renders arithmetic coding easy. Shtarkov makes them slightly dependent with conditioning that appears to be hard to compute. We explain more about this here below in Section 2.3.7.

2.3.7 Computating Shtarkov's NML distribution using Q_a

Independence of the tilted Stirling ratio distribution facilitates an approximate computation of Shtarkov's NML distribution. Now we could also use it to do exact calculation. The conditional distribution of $N_1, \ldots, N_j$ can be calculated as follows

$$\begin{aligned} & Q_{nml}^{m,j}(N_j | N_1, \ldots, N_{j-1}) \\ = & \frac{Q_{nml}^{m,j}(N_1, \ldots, N_j)}{Q_{nml}^{m,j-1}(N_1, \ldots, N_{j-1})} \\ = & \frac{\sum_{\{\sum_{i=j+1}^m N_i = n - \sum_{i=1}^j N_i\}} \prod_{i=1}^m M(N_i)/C(S_{m,n})}{\sum_{\{\sum_{i=j}^m N_i = n - \sum_{i=1}^{j-1} N_i\}} \prod_{i=1}^m M(N_i)/C(S_{m,n})}. \end{aligned}$$

Divide and multiply $e^{-a\sum_{i=j}^{m}}/C_a^{m-j}$ to both the numerator and denominator to obtain

$$\frac{M(N_j)e^{-aN_j}}{C_a}\frac{\sum_{\{\sum_{i=j+1}^{m}N_i=n-\sum_{i=1}^{j}N_i\}}\prod_{i=j+1}^{m}P_a(N_i)}{\sum_{\{\sum_{i=j}^{m}N_i=n-\sum_{i=1}^{j-1}N_i\}}\prod_{i=j}^{m}P_a(N_i)}.$$

Therefore the conditional distribution can be expressed as

$$P_a(N_j)\frac{P_a^{m-j}(\sum_{i=1}^{m}N_i=n|N_1,\ldots,N_j)}{P_a^{m-(j-1)}(\sum_{i=1}^{m}N_i=n|N_1,\ldots,N_{j-1})}.$$

To estimate $P_a^{m-j}(\sum_{i=1}^{m}N_i=n|N_1,\ldots,N_j)$, we could draw independent samples according to P_a and evaluate the sample average. Since the above equality holds for any $a>0$, and here we need a such that there is sufficient probability that the sample total matches $n-\sum_{i=1}^{j}N_i$, we could choose a_η at $\eta=\frac{m}{n-\sum_{i=1}^{j}N_i}$. This way the conditionals of Q_{nml} can be computed conveniently.

2.3.8 Prediction

A sequence of conditional distributions for X_{i+1} given the past observations $X_1,\ldots,X_i$ for $i<n$ provides a sequential prediction with cumulative log loss defined by $\sum_{i<n}\log 1/P(X_{i+1}|X_1,\ldots,X_i)$.

There are two natural ways of providing this sequence of conditionals. One is to get the conditionals from the full joint distribution P_n, which is horizon dependent as mentioned above. It produces cumulative log loss prediction regret precisely the same as the regret of using Q_a for data compression. The other is by using the sequence of distributions $P_{i+1}(X_1,\ldots,X_{i+1}), i<n$, called sequential NML [27]. The sequential prediction distribution $P_{i+1}(X_{i+1}=x|X_1,\ldots,X_i)$ is proportional to

$P_{i+1}(X_1, \ldots, X_i, X_i + 1 = x)$ and accordingly simplifies to

$$P(X_{i+1}{=}x|X_1,\ldots,X_i) = \frac{(N_x^i + 1)^{N_x^i+1}/{N_x^i}^{N_x^i}}{\sum_{\tilde{x}=1}^{m}(N_{\tilde{x}}^i + 1)^{N_{\tilde{x}}^i+1}/{N_{\tilde{x}}^i}^{N_{\tilde{x}}^i}}.$$

Note that the prediction rule does not involve a. Previous study by Shtarkov[1] shows that it is approximately proportional for large N_x to the $N_x + 1/2$ rule of the Laplace-Jeffreys $Drichlet(1/2, \ldots, 1/2)$ update rule (also called the Krichevski-Trofimov rule). Yet it differs importantly from the Laplace-Jeffreys rule for small counts N_x.

It can be seen as a modification of Laplace's rule of succession

$$\underbrace{\frac{N_x^i + 1}{i + m}}_{\text{Laplace's rule}} \underbrace{(1 + 1/N_x^i)^{N_x^i} \frac{i + m}{normalizer}}_{\text{modifier}},$$

For all counts large, the $(1 + 1/N_x^i)^{N_x^i}$ term is approximately e, which makes the modifier nearly constant. But for symbols with small count, this modifier shrinks the Laplace's predictor. These conditional distributions, when put together, formulate the sequential normalized maximum likelihood (sNML) in [27].

However, when using two tilting parameters to adjust for relative importance of symbols within an alphabet, for example, $Q_{a,b}$ in Section 2.3.2, the predictive distribution does depend on b, i.e.,

$$\begin{aligned} & P(X_{i+1}{=}x|X_1,\ldots,X_i) \\ = \; & \frac{e^{-\mathbf{1}_{\{x>L\}}b}(N_x^i + 1)^{N_x^i+1}/{N_x^i}^{N_x^i}}{\sum_{\tilde{x}=1}^{m} e^{-\mathbf{1}_{\{\tilde{x}>L\}}b}(N_{\tilde{x}}^i + 1)^{N_{\tilde{x}}^i+1}/{N_{\tilde{x}}^i}^{N_{\tilde{x}}^i}}. \end{aligned}$$

Hence, all symbols beyond L are discounted by an extra fact of e^{-b} when predicted by this rule.

2.4 Application

2.4.1 Simulation

Theorem 2.2 indicates we could optimize L to save coding cost when the ordered counts are skewed. We look at the performance of the tilted Stirling ratio distribution for algebraically decreasing counts with simulated data. The alphabet is partitioned into two subsets – the frequent symbols and the infrequent ones. The tilting parameter is chosen approximately according to the ratio of the number of symbols in a subset and their total count. The regret of assigning different number of symbols as 'frequent' (L) is shown in Fig. 2.5. We can see that more skewness pushes the optimizing L smaller.

Figure 2.6 shows the upper bound of the minimax regret in Theorem 2.3 for an algebraically decreasing envelope class.

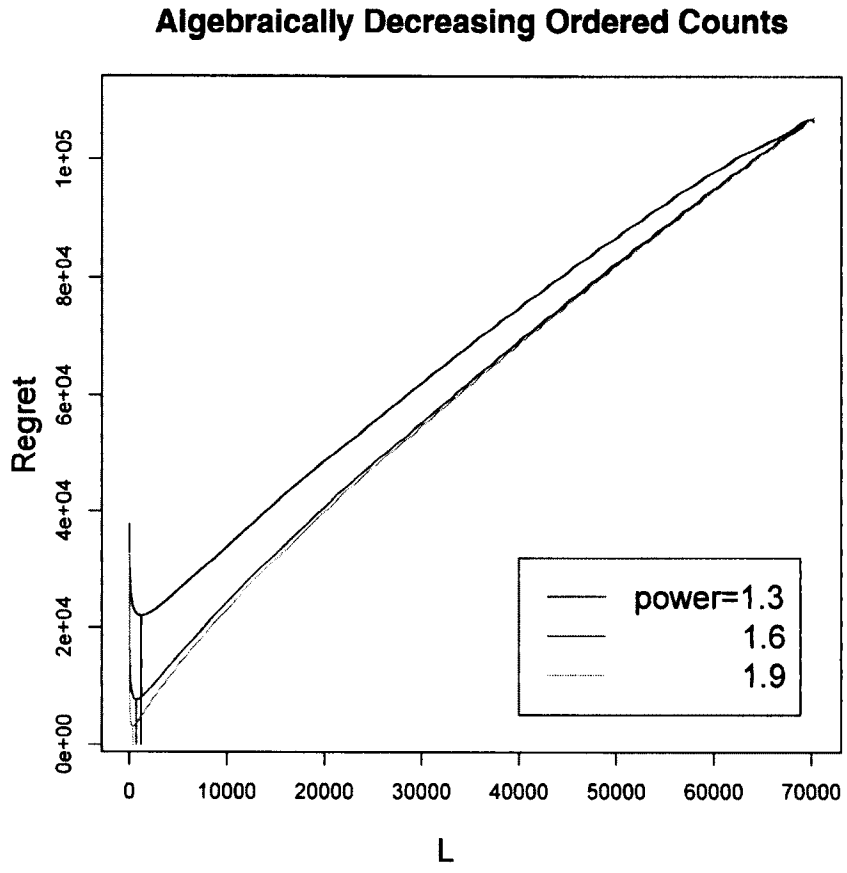


Figure 2.5: Regret of using tilted Stirling ratio distribution for algebraically decreasing counts.

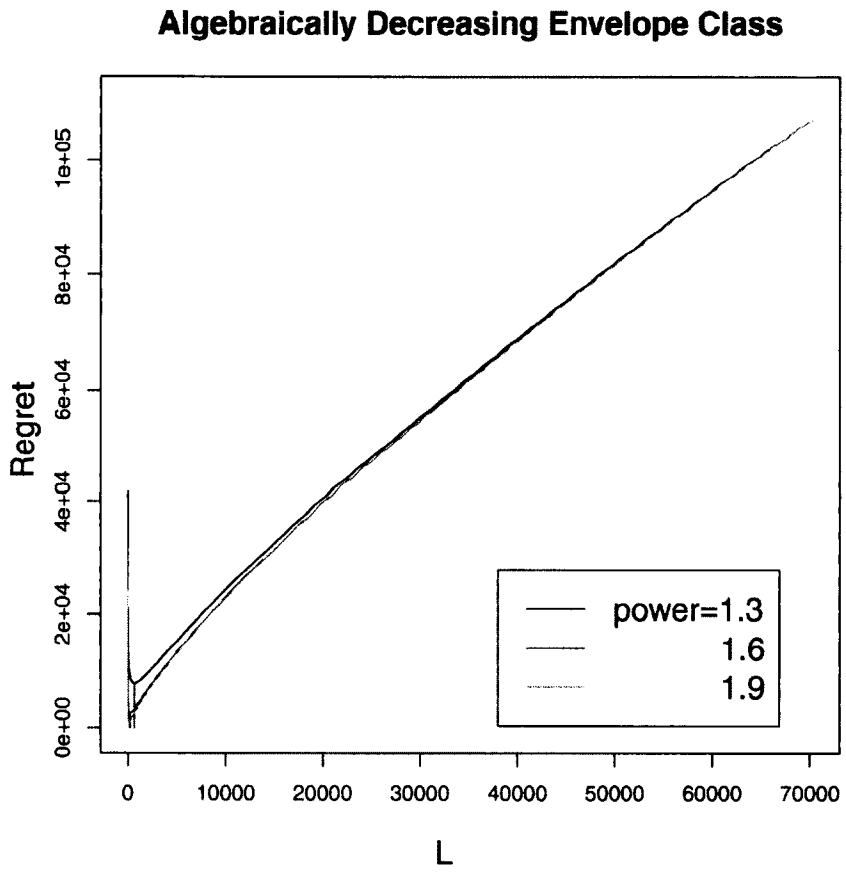


Figure 2.6: Regret of using tilted Stirling ratio distribution for an algebraically decreasing envelope class.

2.4.2 Real data

We also provide an example of using the tilted Stirling ratio distribution to code Chinese literature. The target book is an ancient collection of poems named 诗经, translated as the Classic of Poetry. It is the existing earliest collection of Chinese poetry and dates from the 10th to 7th centuries BC [28]. The book is downloaded freely from `http://wenku.baidu.com/`. Since many ancient words are rarely used today, the encoding is done in GB18030 [29], the largest Chinese coded character set. It contains 70244 characters, among which 2889 appear in the book with a total character count 39161. There are 792 characters appear once and 479 appear twice. The smallest regret happens at $L = 2889$ which is the total number of characters appear.

2.5 Discussion

We have introduced the use of independent tilted maximized Poisson likelihood distributions (also here called tilted Stirling ratio distributions) Q_a for coding the counts for independent random variables. The performance of the coding distribution is close to the minimax level. Actually, the difference between the regret and the minimax level is the probability assigned to the set with the observed total count by the tilted distribution with the optimal tilting parameter, i.e.

$$\begin{aligned} R(M_{cond}, \mathcal{P}_\Lambda^m, S_{m,n}) &= R(Q_{a^*}, \mathcal{P}_\Lambda^m, S_{m,n}) \\ &\quad + \log Q_{a^*}(S_{m,n}). \end{aligned}$$

The optimal tilting parameter a^* minimizes the difference among all possible a. Since M_{cond} reproduces the Shtarkov NML distribution for the multinomial family of

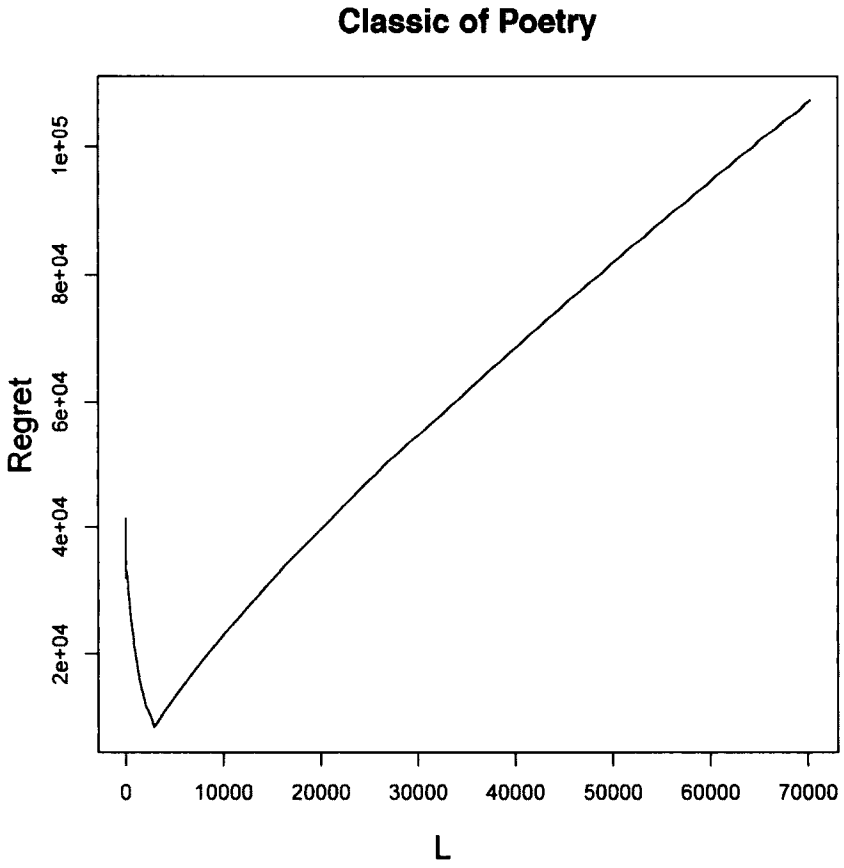


Figure 2.7: Regret of $Q_{a,b}$ for L from 1 to m.

distributions on counts, it is the exact pointwise minimax strategy. As shown in this paper, our findings about the regret produced by the distribution Q_a, taken together with earlier work [1][7][20][16], show that the difference is no larger than about $\log n$ in small alphabet cases, and about $\frac{1}{2}\log n$ for moderate or large alphabets. The probability $Q_a(S_{m,n})$ is the probability distribution for the total count N evaluated at $N = n$ as induced by our distribution Q_a. Further analysis could be done to characterize this distribution of the total count more precisely.

Chapter 3

Markov model

The work of this chapter was published as Compression and Predictive Distributions for Large Alphabet i.i.d and Markov models, Xiao Yang and Andrew Barron, Proceedings of the *2014 IEEE International Symposium on Information Theory(ISIT)*, June 29 2014-July 4, 2504-2508

3.1 Introduction

Non-vanishing per symbol redundancy renders large alphabet compression mission impossible. However, distributions living on large alphabets usually display a decaying trend. For example, in Chinese, a subset of 964 characters covers 90% inputs in Chines [30] though the vocabulary size is more than 100,000 in total.

Coding and prediction of strings of random variables generated from an i.i.d model have been considered for the large alphabet setting with the restriction that the ordered count list rapidly decreasing [31], or satisfies an envelope class property [2][32]. Although this i.i.d model is not the best for compression or prediction when there is dependence between successive characters, it serves as an analytical tool that more complicated models can be based on, and helps understand the behavior of coding and predictive distributions.

Willems, Shtarkov and Tjalkens designed the brilliant context tree weighting (CTW) method for bounded binary tree sources. They derived an upper bound for the regret which is optimal in the sense that it achieves Rissanen (1984) lower bound [33]. Cleary and Witten proposed a convenient method called prediction by partial matching (PPM) which achieves practical efficiency and advantage [34]. Sadakane. Okazaki and Imai implemented CTW for text compression and found difficulties since the original method is for binary sources. Then they proposed a method combining PPM with CTW and showed good practical results by applying it [35].

Suppose a string of random variables $\underline{X} = (X_1, \ldots, X_N)$ is generated independently from a discrete alphabet $\mathcal{A}$ of size m. Here the string length N can be random. Then $\underline{X}$ is a member of the set $\mathcal{X}^*$ of all finite length strings

$$\mathcal{X}^* = \bigcup_{n=0}^{\infty} \{x^n = (x_1, \ldots, x_n) : x_i \in \mathcal{A}, i = 1, \ldots, n\}.$$

Our goal is to code/predict the string $\underline{X}$.

Now suppose given N, each random variable X_i is generated independently according to a probability mass function in a parametric family $\mathcal{P}_\Theta = \{P_{\underline{\theta}}(x) : \underline{\theta} \in \Theta \subset R^m\}$ on $\mathcal{A}$. That is

$$P_{\underline{\theta}}(X_1, \ldots, X_N | N = n) = \prod_{i=1}^{n} P_{\underline{\theta}}(X_i),$$

for $n = 1, 2, \ldots$ We are interested in the class of all distributions with $P_{\underline{\theta}}(j) = \theta_j$ parameterized by the simplex $\Theta = \{\underline{\theta} = (\theta_1, \ldots, \theta_m) : \theta_j \geq 0, \sum_{j=1}^{m} \theta_j = 1, j = 1, \ldots, m\}$.

Let $\underline{N} = (N_1, \ldots, N_m)$ denote the vector of counts for symbol $1, \ldots, m$. The observed sample size N is the sum of the counts $N = \sum_{j=1}^{m} N_j$. Then $P_{\underline{\theta}}(\underline{X})$ have factorizations based on the distribution of the counts

$$P_{\underline{\theta}}(\underline{X}) = P(\underline{X}|\underline{N})\, P_{\underline{\theta}}(\underline{N}).$$

The first factor is the uniform distribution on the set of strings with given counts. The vector of counts $\underline{N}$ forms a sufficient statistic for $\underline{\theta}$. In the particular case of all i.i.d. distributions parameterized by the simplex, the distribution $P_{\underline{\theta}}(\underline{N}|N = n)$ is the *multinomial*$(n, \underline{\theta})$ distribution.

In the above, there is a need for a distribution of the total count N. Of particular interest is the case that the total count is taken to be *Poisson*, because then the resulting distribution of individual counts are independent.

Poisson sampling is a standard technique to simplify analysis [11][36]. Here we consider the target family $\mathcal{P}_\Lambda^m = \{P_{\underline{\lambda}}(\underline{N}) : \lambda_j \geq 0, j = 1, \ldots, m\}$, in which $P_{\underline{\lambda}}(\underline{N})$ is the product of *Poisson*(λ_j) distribution for $N_j, j = 1, \ldots, m$. It makes the total count $N \sim$ *Poisson*(λ_{sum}) with $\lambda_{sum} = \sum_{j=1}^{m} \lambda_j$ and yields the *multinomial*$(n, \underline{\theta})$ distribution by conditioning on $N = n$, where $\theta_j = \lambda_j / \lambda_{sum}$. And the induced

distribution on $\underline{X}$ is

$$P_{\underline{\lambda}}(\underline{X}) = P(\underline{X}|\underline{N})P_{\underline{\lambda}}(\underline{N}).$$

Adopting the conventional definition for *regret*, we have

$$R(Q, P_{\hat{\underline{\lambda}}}, \underline{X}) = \log \frac{1}{Q(\underline{X})} - \log \frac{1}{P_{\hat{\underline{\lambda}}}(\underline{X})},$$

where $P_{\hat{\underline{\lambda}}}(\underline{X}) = \max_{\underline{\lambda} \in \Lambda}(P_{\underline{\lambda}}(\underline{X}))$, and log is logarithm base 2.

Here we can construct Q by choosing a probability distribution for the counts and then use the uniform distribution for the distribution of strings given the counts, written as $P_{unif}(\underline{X}|\underline{N})$. Then the regret becomes

$$R(Q, P_{\hat{\underline{\lambda}}}, \underline{X}) = \log \frac{P_{\hat{\underline{\lambda}}}(\underline{N})}{Q(\underline{N})}.$$

And the problem becomes: given the family $\mathcal{P}_{\Lambda}^{m}$, how to choose Q to minimize the maximized regret

$$\min_{Q} \max_{\underline{X}} R(Q, P_{\hat{\underline{\lambda}}}, \underline{X}) = \min_{Q} \max_{\underline{N}} \log \frac{P_{\hat{\underline{\lambda}}}(\underline{N})}{Q(\underline{N})}.$$

For the regret, the maximum can be restricted to a set of counts instead of the whole space. A traditional choice being $S_{m,n} = \{(N_1, \ldots, N_m) : \sum_{j=1}^{m} N_j = n, N_j \geq 0,\ j = 1, \ldots, m\}$ associated with a given sample size n, in which case the minimax regret is

$$\min_{Q} \max_{\underline{N} \in S_{m,n}} \log \frac{P_{\hat{\underline{\lambda}}}(\underline{N})}{Q(\underline{N})}.$$

As is familiar in universal coding [1][7], the NML distribution

$$Q_{nml}(\underline{N}) = \frac{P_{\hat{\underline{\lambda}}}(\underline{N})}{C(S_{m,n})}$$

is the unique pointwise minimax strategy when $C(S_{m,n}) = \sum_{\underline{N} \in S_{m,n}} P_{\hat{\underline{\lambda}}}(\underline{N})$ is finite, and $\log C(S_{m,n})$ is the minimax value. When m is large, the NML distribution can be unwieldy to compute for compression or prediction. Instead we will introduce a slightly suboptimal coding distribution that makes the counts independent and show that it is nearly optimal for every $S_{m,n'}$ with n' not too different from a target n. Indeed, we advocate that our simple coding distribution is preferable to use computationally when m is large even if the sample size n were known in advance.

To produce our desired coding distribution we make use of two basic principles. One is that the multinomial family of distributions on counts matches the conditional distribution of $N_1, \ldots, N_m$ given the sum N when unconditionally the counts are independent Poisson. Another is the information theory principle [12][13][14] that the conditional distribution given a sum (or average) of a large number of independent random variables is approximately a product distribution, each of which is the one closest in relative entropy to the unconditional distribution subject to an expectation constraint. This minimum relative entropy distribution is an exponential tilting of the unconditional distribution.

In the Poisson family with distribution $\lambda_j^{N_j} e^{-\lambda_j}/N_j!$, exponential tilting (multiplying by the factor e^{-aN_j}) preserves the Poisson family (with the parameter scaled to $\lambda_j e^{-a}$). Those distributions continue to correspond to the multinomial distribution (with parameters $\theta_j = \lambda_j/\lambda_{sum}$) when conditioning on the sum of counts N. A particular choice of $a = \ln(\lambda_{sum}/N)$ provides the product of Poisson distributions closest to the multinomial in regret. Here for universal coding, we find the tilting of individual maximized likelihood that makes the product of such closest to the Shtarkov's NML distribution. This greatly simplifies the task of approximate optimal universal compression and the analysis of its regret.

Indeed, applying the maximum likelihood step to a Poisson count k produces a

maximized likelihood value of $M(k) = k^k e^{-k}/k!$. We call this maximized likelihood the *Stirling ratio*, as it is the quantity that Stirling's approximation shows near $(2\pi k)^{-1/2}$ for k not too small. We find that this $M(k)$ plays a distinguished role in universal large alphabet compression, even for sequences with small counts k. Although M has an infinite sum by itself, it is normalizable when tilted for every positive a. The *tilted Stirling ratio distribution* is

$$P_a(N_j) = \frac{N_j^{N_j} e^{-N_j}}{N_j!} \frac{e^{-aN_j}}{C_a}, \tag{3.1}$$

with the normalizer $C_a = \sum_{k=0}^{\infty} k^k e^{-(1+a)k}/k!$.

The coding distribution we propose and analyze is simply the product of those tilted one-dimensional maximized Poisson likelihood distributions for a properly chosen a

$$Q_a(\underline{N}) = P_a^m(\underline{N}) = P_a(N_1) \cdots P_a(N_m).$$

If it is known that the total count is n, then the regret is a simple function of n and the normalizer C_a. The choice of the tilting parameter a^* given by the moment condition $\mathbf{E}_{Q_a} \sum_{j=1}^m N_j = n$ minimizes the regret over all positive a. Moreover, value of a^* depends only on the ratio between the size of the alphabet and the total count m/n. Details about finding a^* can be found in [31].

As compared to i.i.d class, Markov sources are richer and more realistic. Suppose given N, each random variable X_i is generated according to a probability mass function depending on its *context* (string of symbols preceding it). Following Willems et al' notations in [33], a tree source can be determined by a context set $\mathcal{S}$. Elements of $\mathcal{S}$ are strings of symbols from $\mathcal{A}$ or concatenation of "others" and suffixes of the existing contexts. The case "others" represents complements of the contexts in $\mathcal{S}$ with a common *parent*. The "others" can be different for each set of branches

from a node (as it is the complement of the set of symbols identified on the other branches). Note that CTW can be applied to large alphabet, but it does not have the flexibility of collapsing the symbols on each branch. The collection of distributions is $\mathcal{P}_{\Theta_S} = \{P_{\underline{\theta}_s}(x) : \underline{\theta}_s \in \Theta_S, s \in \mathcal{S}\}$, where Θ_S is the parameter set defined later. For simplicity, we require the order of the model no larger than $T \in \{0, 1, 2, \ldots\}$, so $\mathcal{S} \in \mathcal{C}_T$, where $\mathcal{C}_T$ is the class of tree sources with order T or less.

For each context $s \in \mathcal{S}$ with a given $\mathcal{S}$, let θ_{sx} denote the probability of symbol $x \in \mathcal{A}$ showing up after s, for all $x \in \mathcal{A}$. Then $\underline{\theta}_{s\cdot} = (\theta_{s1}, \ldots, \theta_{sm})$ lies in the set

$$\Theta_S = \{\underline{\theta}_{s\cdot} = (\theta_{s1}, \ldots, \theta_{sm}) : x \in \mathcal{A}, \theta_{sx} \geq 0, \sum_{x \in \mathcal{A}} \theta_{sx} = 1\}.$$

Again, we could take advantage of factorizations based on the distribution of the counts $\mathbf{N}_S = (\underline{N}_{s\cdot})_{s \in \mathcal{S}}$, where $\underline{N}_{s\cdot} = (N_{s1}, \ldots, N_{sm})$ is the count for all symbols given context $s \in \mathcal{S}$, and Pick the distribution for the total count to be *Poisson*. It leads to the target family $\mathcal{P}_\Lambda^{|\mathcal{S}|m} = \{P_{\underline{\lambda}}(\mathbf{N}_S) : \lambda_{sj} \geq 0, j = 1, \ldots, m, s \in \mathcal{S}\}$, in which $P_{\underline{\lambda}}(\mathbf{N}_S)$ is the product of $Poisson(\lambda_{sj})$ distribution for $N_{sj}, j = 1, \ldots, m$ and $s \in \mathcal{S}$.

There are two sources of costs involved in using a tree model. One is the *coding cost* for the string given the tree. The other is the *description cost* $D(\mathcal{S})$ for describing the tree. Overall, we want to find Q which uses shorter codelength for sequences generated from an unknown tree source $\mathcal{S} \in \mathcal{C}_T$. That is, to minimize

$$\min_{\mathcal{S} \in \mathcal{C}_T} \left(\log 1/Q(\underline{X}|\mathcal{S}) + D(\mathcal{S})\right).$$

We use the same coding distribution as given in equation (3.1) for count variables conditional on each given context s. The coding distribution for the counts given s

is simply the product

$$Q_{a_s}(\underline{N}_{s\cdot}) = P_{a_s}^m(\underline{N}_{s\cdot}) = P_{a_s}(N_{s1}) \cdots P_{a_s}(N_{sm}), \tag{3.2}$$

with a properly chosen a_s for each context $s \in \mathcal{S}$. Using the product of tilted distribution P_{a_s} as a coding distribution, the regret is simply a sum of the individual regrets.

To construct the tree, we adopt a method similar to Rissanen's approach in [37]. It is different from [33] in that we adapt the method to work with large alphabet and the inclusion of the symbol class "others" on each branch. Using the total codelength to evaluate the performance of different models and coding distributions, we adopt a greedy algorithm to build the context tree with details discussed in Section 3.3.3. An illustrative example tree is given in Figure 3.1.

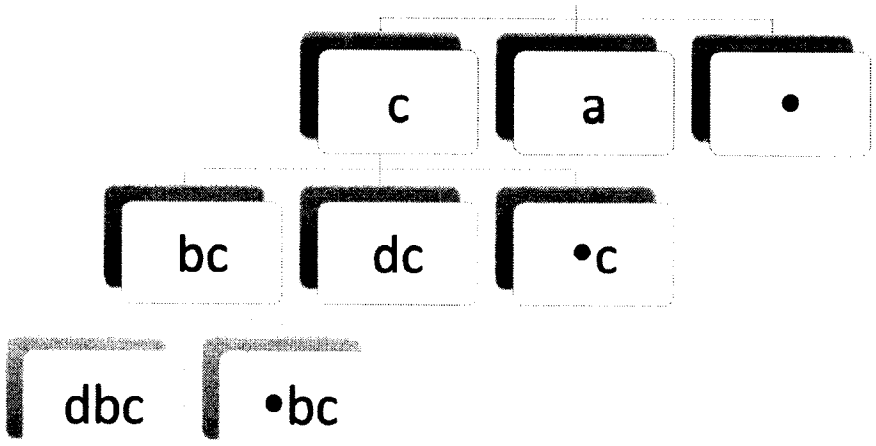


Figure 3.1: An example context tree with $\mathcal{A} = \{a, b, c, d\}$ where • represents "others".

3.2 i.i.d class

The following result is given in Chapter 2.

Theorem. *The regret of using a product of tilted Sterling Ratio distributions Q_a for a given vector of counts $\underline{N} = (N_1, \ldots, N_m)$ is*

$$R\left(Q_a, \mathcal{P}_\Lambda^m, \underline{N}\right) = aN \log e + m \log C_a.$$

Let $S_{m,n}$ be the set of count vectors with total count n be defined as before, then

$$\max_{\underline{N} \in S_{m,n}} R\left(Q_a, \mathcal{P}_\Lambda^m, \underline{N}\right) = an \log e + m \log C_a. \tag{3.3}$$

Let a^* be the choice of a satisfying the following moment condition

$$\mathbf{E}_{P_a} \sum_{j=1}^{m} N_j = m\, \mathbf{E}_{P_a} N_1 = n. \tag{3.4}$$

Then a^* is the minimizer of the regret in expression (3.3). Write $R_{m,n} = \min_a R(Q_a, \mathcal{P}_\Lambda^m, S_{m,n})$.

When $m = o(n)$, the $R_{m,n}$ is near $\frac{m}{2} \log \frac{ne}{m}$ with

$$\begin{aligned} -d_1 \frac{m}{2} \log e \;&\leq\; R_{m,n} - \frac{m}{2} \log \frac{ne}{m} \\ &\leq\; m \log(1 + \sqrt{\frac{m}{n}}), \end{aligned}$$

where $d_1 = O\left((\frac{m}{n})^{1/3}\right)$.

When $n = o(m)$, the $R_{m,n}$ is near $n \log \frac{m}{ne}$ as follows.

$$\begin{aligned} m \log\left(1 + (1 - d_2)\frac{n}{m}\right) \;&\leq\; R_{m,n} - n \log \frac{m}{ne} \\ &\leq\; m \log\left(1 + \frac{n}{m} + d_3\right) \end{aligned}$$

where $d_2 = O(\frac{n}{m})$, *and* $d_3 = \frac{1}{2\sqrt{\pi}} \frac{n^2 e^2}{m(m-ne)}$.

When $n = bm$, *the* $R_{m,n} = cm$, *where the constant* $c = a^* b \log e + \log C_{a^*}$, *and* a^* *is such that* $\mathbf{E}_{P_a} N_1 = b$.

Proof. Details of proof can be found in [31]. □

Remark 3.1. *: The regret depends only on the number of parameters m, the total counts n and the tilting parameter a. The optimal tilting parameter is given by a simple moment condition in equation (3.4).*

Remark 3.2. *: The regret* $R_{m,n}$ *is close to the minimax level in all three cases listed in Theorem 3.2. The main terms in the* $m = o(n)$ *and* $n = o(m)$ *cases are the same as the minimax regret given in [16] except the multiplier for* $\log(ne/m)$ *here is* $m/2$ *instead of* $(m-1)/2$ *for the small m scenario. For the* $n = bm$ *case, the* $R_{m,n}$ *is close to the minimax regret in [16] numerically.*

3.3 Tree source

3.3.1 Coding cost

The coding distribution for a given tree is the product of all the $Q_{a_s}(\underline{N}_{s\cdot})$, i.e.

$$Q_a^{\mathcal{S}}(\mathbf{N}_{\mathcal{S}}) = \prod_{s \in \mathcal{S}} Q_{a_s}(N_{s\cdot}).$$

Let $S_{m,n,\mathcal{S}} = \{\mathbf{N}_{\mathcal{S}} : \sum_{s\in\mathcal{S}} \sum_{j=1}^{m} N_{sj} = n, N_{sj} \geq 0,\ j = 1, \ldots, m, s \in \mathcal{S}\}$.

Corollary 2. *Using independent tilted Stirling ratio distribution* $Q_a^{\mathcal{S}}$ *to code the counts in* $S_{m,n,\mathcal{S}}$, *the regret equals*

$$\max_{\mathbf{N}_{\mathcal{S}} \in S_{m,n,\mathcal{S}}} R(\mathcal{P}_\Lambda^{|\mathcal{S}|m}, Q_a^{\mathcal{S}}, \mathbf{N}_{\mathcal{S}}) = \sum_{s\in\mathcal{S}} (a_s N_s \log e + m \log C_{a_s}).$$

This can be easily seen by applying the definition.

3.3.2 Description cost

To describe a given context set $\mathcal{S}$, we use the following rule

$$D(S) = 1 + N_{branches}\,(1 + \log|\mathcal{A}|)\,,$$

where $N_{branches}$ is the number of "labeled" branches in the tree. Here "labeled" means having a specified symbol in the alphabet. For instance, $N_{branches} = 5$ in the example tree.

The first bit is used to describe if the model is nondegenerate (i.i.d or Markov). For each branch other than "others", we first use 1 bit to say if it is nondegenerate, and then $\log m$ bits to convey which symbol it is. Our example tree uses $1 + 5(1 + \log 4) = 16$ bits.

3.3.3 Using codelength to construct the tree

Here we use the example in Figure 3.1 to illustrate how we construct the tree. Starting from a null tree (the i.i.d model), we first choose the single symbol (c) that produces the most savings (if any) in codelength. Next, we consider two possible leaves: one is another symbol in the first level (a) that achieves the most savings; the other is to extend to the second level based on the symbols just found. After calculating possible savings produced by these two candidates, we pick again the one with larger savings. Continue in this fashion until no more savings is available or the maximum number of symbols to condition on (T) is reached, the context tree is built. "others" represents contexts with the same parent that are not picked up. It includes b and d in the first level in the example tree.

3.4 A real example

We apply the proposed method to a contemporary Chinese novel translated as Fortress Besieged. The book contains 216,601 characters encoded in GB18030, the largest official Chinese character set which contains 70,244 characters.

The i.i.d model uses 1,954,777 bits. For the tree model, the first single character to condition on saves 12,631. We restrict the order of the Markov model to be no larger than 5, but it turns out no context exceeding two characters shows up. There are 342 branches in the tree, among which 95 are in the first level, and 5 of them extends to the second level. In fact, second level branches are picked up only after most first level ones are chosen. A small part of the tree is displayed in Figure 3.2. It corresponds to the earlier steps that produce the most savings in the tree construction. The dots on the right stand for the rest of the model that cannot be shown. And the blank cell in the middle of the first level is the space symbol. The total savings amount to 401,922 bits (about 20.56%) as compared to the i.i.d model. Please note that existing models for tree sources are mostly designed for small alphabet compression, hence direct comparison with which would not be quite fair.

3.5 Conclusion

In this chapter, we consider a compression and prediction problem under bounded tree models for large alphabets, and design a greedy algorithm to construct the context tree. Combining this method with the tilted Stirling ratio distribution, we have a convenient and efficient way for compression and prediction for variables generated from Markov models.

3.6 Discussion

Further investigation can be done to find out the performance of the tree construction algorithm and whether it finds the optimal structure. Moreover, developing a "large alphabet context tree weighting algorithm" like the context tree weighting algorithm for binary alphabet [33] will have great values.

Figure 3.2: Context tree for Fortress Beseiged.

Chapter 4

Summary and future work

This study focuses on compression and prediction of sequences of random variables generated from large alphabets. An alphabet is a set containing all the possible outcomes of a discrete random variable. Many conventional statistical problems assume the sample size is larger than the alphabet size. Yet this assumption is not always true, and can result in the failure of traditional techniques when it doesn't hold. An alphabet is "large" when its size is comparable or even larger than the size of the sample. Large alphabet is of particular interest when the underlying distribution lives on a huge set such as language processing on the word basis.

In the previous study, we started from the i.i.d model and proposed a simple coding distribution formulated by a product of tilted Poisson distributions. This coding distribution achieves close to optimal performance for compressing the i.i.d class. It also simplifies the analysis and computation through the independent structure. By using this distribution, we were able to characterize the regret (the extra bits induced by using the coding distribution instead of the unknown true distribution) through a tail sum of the ordered counts. This is particularly useful when the underlying distributions live in a subclass in which most probability is captured by a small number of symbols. Also, this coding distribution can be applied to the envelope class. The simplicity of the independent structure of the coding distribution can also be used to do exact computation of the Shtarkov's NML distribution. We include a brief discussion of the idea in Section 2.3.6. And further analytical and numerical studies and be done to make the discussion complete.

Further, we considered a more realistic case–Markov models, and in particular, tree sources. A context tree based algorithm was designed to illustrate the dependency of the contexts in the context set. It is a greedy algorithm which seeks for the greatest savings in codelength when constructing the tree. Compression and prediction of individual counts associated with the contexts also uses a product of

tilted Stirling ratio distributions. This algorithm serves as our first trial to provide a solution to this problem. It can be improved since currently it does not guarantee the optimal result. One natural idea is to develop a weighting method mimicking the context tree weighting (CTW) algorithm which can deal with large alphabets (instead of only the binary alphabet). This is particularly usefully given the need to handle large alphabet distributions with dependent structure and the optimal theoretical behavior of CTW.

Appendices

Appendix A

Proof of Theorems

A.1 Some facts

Fact 1. *For any $a > 0$,*

$$\frac{1}{\sqrt{2\pi}} \int_0^1 t^{-\frac{1}{2}} e^{-at} dt < \sqrt{\frac{2}{\pi}}.$$

Proof.

$$\begin{aligned} \frac{1}{\sqrt{2\pi}} \int_0^1 t^{-\frac{1}{2}} e^{-at} dt &\overset{u=at}{=} \frac{1}{\sqrt{2\pi}} \int_0^a (\frac{u}{a})^{-\frac{1}{2}} e^{-u} \frac{1}{a} du \\ &= \frac{1}{\sqrt{2\pi a}} \int_0^a u^{-\frac{1}{2}} e^{-u} du \end{aligned}$$

The integrand is smaller than $u^{-\frac{1}{2}}$ on $[0, a]$, so the integral is upper bounded by

$$\frac{1}{\sqrt{2\pi a}} \int_0^a u^{-\frac{1}{2}} du = \sqrt{\frac{2}{\pi}}.$$

□

Fact 2. *For any $a > 0$,*

$$\sum_{k=1}^{\infty} \frac{k^{-\frac{1}{2}}}{\sqrt{2\pi} e^{r_k}} e^{-ak} \geq \frac{1}{\sqrt{2\pi}} \int_1^{\infty} t^{-\frac{1}{2}} e^{-at} dt$$

when $\frac{1}{12k+1} \leq r_k \leq \frac{1}{12k}$.

Proof. It suffice to show

$$\sum_{k=1}^{\infty} \frac{k^{-\frac{1}{2}}}{e^{\frac{1}{12k}}} e^{-ak} \geq \int_1^{\infty} t^{-\frac{1}{2}} e^{-at} dt \tag{A.1}$$

Note that $t^{-\frac{1}{2}} e^{-at}$ is convex in t, so we have $\int_k^{k+1} f(t) dt$ upper bounded by $(f(k) + f(k+1))/2$. Then we only need to show the latter is upper bounded by $f(k) e^{-1/12k}$.

This can be done by proving the following inequality.

$$\left(1+\left(\frac{k}{k+1}\right)^{\frac{1}{2}}e^{-a}\right)e^{\frac{1}{12k}} \leq 2$$

for each $k \geq 1$ and $a > 0$. Check that the left hand side is increasing in k by taking derivative, its value goes up to $1+e^{-a}$ which is not larger than the right hand side for every $a \geq 0$. Therefore, Inequality (A.1) follows. □

Lemma A.1 (Bounds for C_a). *For any $a > 0$, the following bounds hold for C_a*

$$\max(1, 1-\sqrt{\frac{2}{\pi}}+\frac{1}{\sqrt{2a}}) < C_a < 1+\frac{1}{\sqrt{2a}}, \tag{A.2}$$

and

$$1+e^{-(a+1)} < C_a < 1+e^{-(a+1)}+\frac{1}{2\sqrt{\pi}}\frac{e^{-2a}}{1-e^{-a}}. \tag{A.3}$$

Proof. The argument to prove the upper bounds is analogous to Fact 2. Indeed,

$$C_a = \sum_{k=0}^{\infty}\frac{k^k e^{-k}}{k!}e^{-ak} \overset{(a)}{=} 1+\sum_{k=1}^{\infty}\frac{k^{-\frac{1}{2}}}{\sqrt{2\pi}e^{r_k}}e^{-ak} \tag{A.4}$$

Here (a) is by Robbins' refinement of Stirling's approximation where $\frac{1}{12k+1} < r_k < \frac{1}{12k}$.

We recognize the similarity of this Stirling approximation $\frac{k^{-\frac{1}{2}}}{\sqrt{2\pi}}e^{-ak}$ to the $Gamma(1/2, a)$ density as plotted in Figure A.1. Indeed, the sum C_a can be bounded by a gamma integral as demonstrated in Figure A.1, so

$$\begin{aligned} C_a &\leq 1+\frac{1}{\sqrt{2\pi}}\int_0^{\infty} t^{-\frac{1}{2}}e^{-ta}dt \\ &= 1+\frac{1}{\sqrt{2\pi}}\frac{\Gamma(\frac{1}{2})}{a^{\frac{1}{2}}} \\ &= 1+\frac{1}{\sqrt{2a}}. \end{aligned}$$

Also, following expression (A.4), C_a has the following lower bound.

$$\begin{aligned} C_a &= 1+\sum_{k=1}^{\infty}\frac{k^{-\frac{1}{2}}}{\sqrt{2\pi}e^{r_k}}e^{-ak} \\ &\overset{(b)}{\geq} 1-\sqrt{\frac{2}{\pi}}+\sqrt{\frac{2}{\pi}}+\frac{1}{\sqrt{2\pi}}\int_1^{\infty}t^{-\frac{1}{2}}e^{-at}dt \\ &\overset{(c)}{>} 1-\sqrt{\frac{2}{\pi}}+\frac{1}{\sqrt{2\pi}}\int_0^{1}t^{-\frac{1}{2}}e^{-at}dt \\ &\quad +\frac{1}{\sqrt{2\pi}}\int_1^{\infty}t^{-\frac{1}{2}}e^{-at}dt \\ &= 1-\sqrt{\frac{2}{\pi}}+\frac{1}{\sqrt{2\pi}}\int_0^{\infty}t^{-\frac{1}{2}}e^{-at}dt \\ &= 1-\sqrt{\frac{2}{\pi}}+\frac{1}{\sqrt{2a}}. \end{aligned}$$

Here again $\frac{1}{12k+1} < r_k < \frac{1}{12k}$, and inequality (b) is due to Fact 2 and inequality (c) is by Fact 1.

Note that inequality (A.2) is good for small a. For a moderately large a ($a > 0.2$), the following upper bound is better.

$$\begin{aligned} C_a &\leq 1+e^{-(a+1)}+\sum_{k=2}^{\infty}\frac{1}{\sqrt{2\pi k}}e^{-ka} \\ &< 1+e^{-(a+1)}+\frac{1}{2\sqrt{\pi}}\frac{e^{-2a}}{1-e^{-a}}. \end{aligned}$$

□

Lemma A.2. *For any $a > 0$,*

$$e^{-(a+1)} \leq \mathbf{E}_{P_a}N_1 \leq \frac{1}{2a}.$$

Proof. Let $k^* = \min_{k\in\mathbf{N}_+}\left|k-\frac{1}{2a}\right|$. We prove the upper bound by consider a within

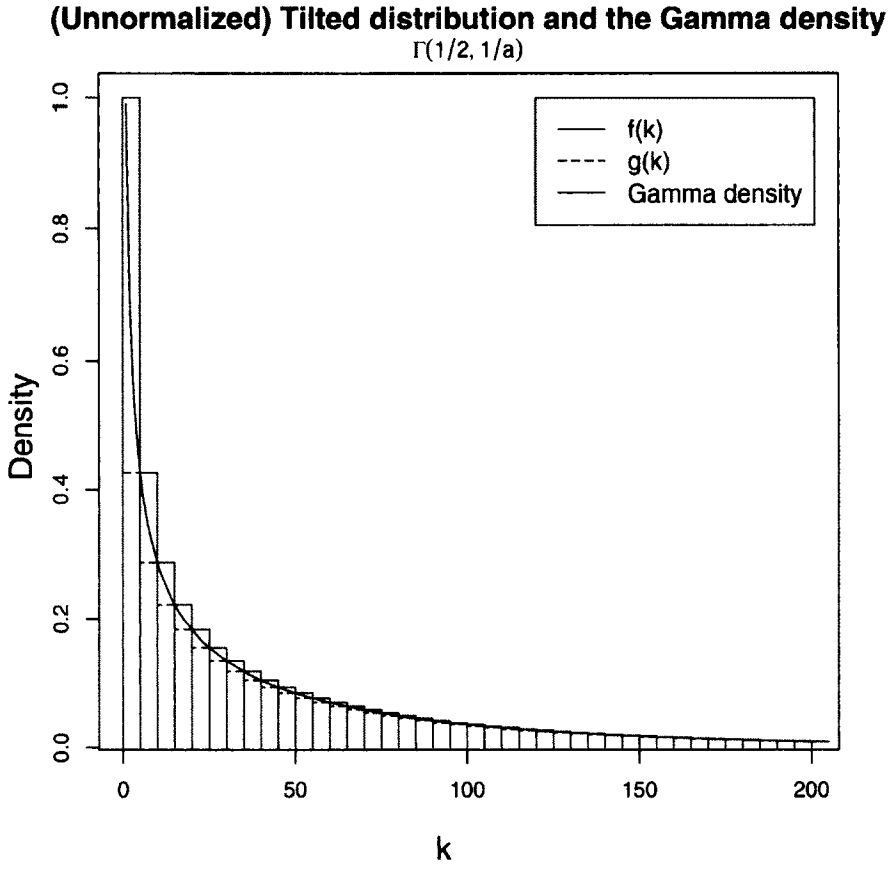


Figure A.1: tilted distribution and the $\Gamma(\frac{1}{2}, \frac{1}{a})$ *density with* $a = 0.01$.

two different intervals. First, if $a \leq e(\sqrt{\pi} - \sqrt{2})^2$, we know

$$\begin{aligned}
& \sum_{k=1}^{\infty} \frac{k^{k+1}e^{-k}}{k!} e^{-ak} \\
= & \sum_{k=1}^{k^*-1} \frac{k^{k+1}e^{-k}}{k!} e^{-ak} + \sum_{k=k^*+1}^{\infty} \frac{k^{k+1}e^{-k}}{k!} e^{-ak} \\
& + \frac{k^{*k^*+1}e^{-k^*}}{k^*!} e^{-ak^*} \\
\overset{(a)}{\leq} & \sum_{k=1}^{k^*-1} \frac{k^{1/2}e^{-ak}}{\sqrt{2\pi}} + \sum_{k=k^*+1}^{\infty} \frac{k^{1/2}e^{-ak}}{\sqrt{2\pi}} \\
& + \frac{k^{*1/2}e^{-ak^*}}{\sqrt{2\pi}}
\end{aligned} \tag{A.5}$$

where (a) is an upper bound by Stirling's approximation.

Both sums in the last expression can be upper bounded by a gamma integral as shown in Figure A.2 and Figure A.3, and $k^{*1/2}e^{-ak^*}$ is no larger than the maximum of the unnormalized $Gamma(3/2, 1/a)$ density, which is achieved at $1/(2a)$. Note that for approximation of the mean, the power of k is $1/2$ rather than $-1/2$. Accordingly, we use the $Gamma(3/2, 1/a)$ density in approximating the terms of the sum. Hence, we have the following upper bound for expression (A.5).

$$\begin{aligned}
& \int_0^{k^*} \frac{t^{1/2}e^{-at}}{\sqrt{2\pi}} dt + \int_{k^*}^{\infty} \frac{t^{1/2}e^{-at}}{\sqrt{2\pi}} dt + \frac{(1/2a)^{1/2}e^{-1/2}}{\sqrt{2\pi}} \\
= & \frac{\Gamma(3/2)}{a^{3/2}\sqrt{2\pi}} + \frac{(1/2a)^{1/2}}{\sqrt{2\pi e}} \\
= & \frac{1}{(2a)^{3/2}} + \frac{1}{\sqrt{2\pi e}} \frac{1}{(2a)^{1/2}}
\end{aligned}$$

Using this upper bound for C_a, we could prove an upper bound for the expected

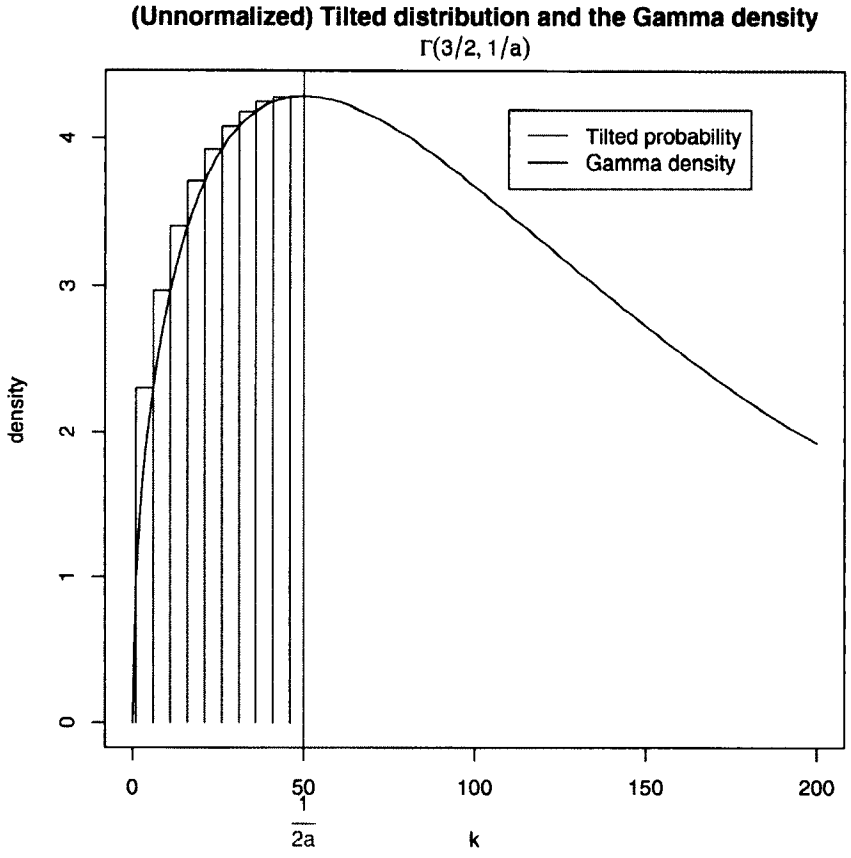


Figure A.2: Tilted distribution and the Gamma density. The relevant sum is only to the left of $\frac{1}{2a}$ *with* $a = 0.01$.

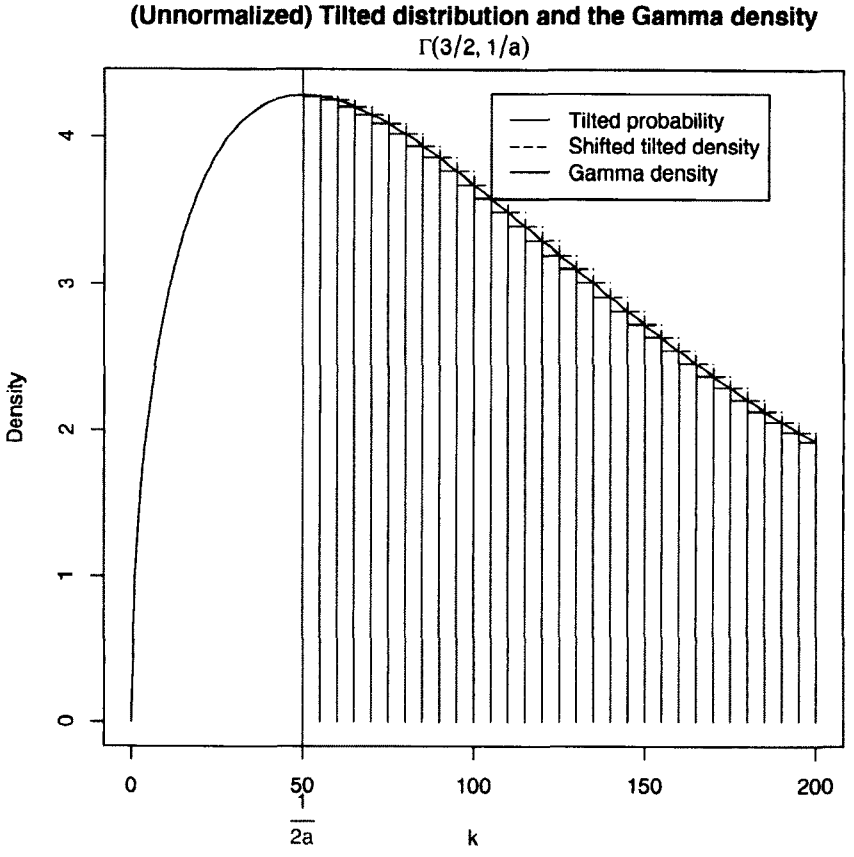


Figure A.3: Tilted distribution and the Gamma density. The relevant sum is only to the right of $\frac{1}{2a}$ *with* $a = 0.01$.

value.

$$\begin{aligned}
\mathbf{E}_{P_a} N_1 &= \sum_{k=1}^{\infty} \frac{k^{k+1} e^{-k}}{k!\, C_a} e^{-ak} \\
&\overset{(b)}{\leq} \frac{\frac{1}{(2a)^{3/2}} + \frac{1}{\sqrt{2\pi e}} \frac{1}{(2a)^{1/2}}}{\frac{1}{(2a)^{1/2}} + 1 - \sqrt{\frac{2}{\pi}}} \\
&= \frac{1}{2a} \underbrace{\left(\frac{\frac{1}{(2a)^{1/2}} + \frac{1}{\sqrt{2\pi e}} (2a)^{1/2}}{\frac{1}{(2a)^{1/2}} + 1 - \sqrt{\frac{2}{\pi}}} \right)}_{(A)}
\end{aligned}$$

The lower bound for the denominator in (b) is attributed to Lemma A.1. A little algebra can show that term (A) is not larger than 1 when a is restricted to $(0, e(\sqrt{\pi} - \sqrt{2})^2]$.

If $a > e(\sqrt{\pi} - \sqrt{2})^2$, we have $\arg\max_{k \geq 1} k^{1/2} e^{-ak} = 1$. Using Stirling's approximation and split the sum into $k = 1$ and $k > 1$, we have

$$\begin{aligned}
& \sum_{k=1}^{\infty} \frac{k^{k+1} e^{-k}}{k!} e^{-ak} \\
\leq\ & \frac{e^{-a}}{\sqrt{2\pi}} + \sum_{k=2}^{\infty} \frac{k^{1/2} e^{-ak}}{\sqrt{2\pi}} \\
\overset{(c)}{\leq}\ & \frac{1}{\sqrt{2\pi}} \left(\frac{1}{2} e^{-a} + \int_0^{\infty} t^{1/2} e^{-at} dt \right) \\
=\ & \frac{1}{\sqrt{2\pi}} \left(\frac{1}{2} e^{-a} + \frac{\Gamma(3/2)}{a^{3/2}} \right) \\
=\ & \frac{1}{2\sqrt{2\pi}} e^{-a} + \frac{1}{(2a)^{3/2}}
\end{aligned}$$

where (c) is because the sum $\sum_{k=2}^{\infty} k^{1/2} e^{-ak}$ is bounded above by the integral $\int_1^{\infty} t^{1/2} e^{-at} dt$, and the difference between $\int_0^1 t^{1/2} e^{-at} dt$ and e^{-a} (value of $k^{1/2} e^{-ak}$ at $k = 1$) is less than $\frac{1}{2} e^{-a}$ due to the concavity of $t^{1/2} e^{-at}$ to the left of $1/2a$.

By this upper bound for the numerator and Lemma A.1 again,

$$\begin{aligned}\mathbf{E}_{P_a} N_1 &\leq \frac{\frac{1}{(2a)^{3/2}} + \frac{1}{2\sqrt{2\pi}} e^{-a}}{\frac{1}{(2a)^{1/2}} + 1 - \sqrt{\frac{2}{\pi}}} \\ &= \frac{1}{2a} \underbrace{\left(\frac{\frac{1}{(2a)^{1/2}} + \frac{1}{\sqrt{2\pi}} a e^{-a}}{\frac{1}{(2a)^{1/2}} + 1 - \sqrt{\frac{2}{\pi}}} \right)}_{(B)}.\end{aligned}$$

Term (B) is not larger than 1 because $\frac{1}{\sqrt{2\pi}} a e^{-a} \leq 1 - \sqrt{\frac{2}{\pi}}$ for all a.

For the lower bound,

$$\begin{aligned}\mathbf{E}_{P_a} N_1 &= \sum_{k=1}^{\infty} \frac{k^{k+1} e^{-k}}{k!\, C_a} e^{-ak} \\ &= \frac{e^{-(a+1)} \left(\sum_{k=1}^{\infty} \frac{k^k e^{-(k-1)}}{(k-1)!} e^{-a(k-1)} \right)}{C_a} \\ &\overset{l=k-1}{=} \frac{e^{-(a+1)} \left(\sum_{l=0}^{\infty} \frac{(l+1)^{l+1} e^{-l}}{l!} e^{-al} \right)}{C_a} \\ &= e^{-(a+1)} \underbrace{\left(\frac{\sum_{l=0}^{\infty} \frac{(l+1)^{l+1} e^{-l}}{l!} e^{-al}}{\sum_{k=0}^{\infty} \frac{k^k e^{-k}}{k!} e^{-ak}} \right)}_{(C)} \\ &\overset{(d)}{\geq} e^{-(a+1)}\end{aligned} \tag{A.6}$$

Here inequality (d) is because term (C) is above 1. Hence, the upper bound is deduced.

□

A.2 Proof of Theorem 3.2

Proof. It remains to show the two lower bounds in expression (2.6) and (2.7). In both cases we need a lower bound for $na^* \log e + m \log C_{a^*}$, and we do it by lower bounding a^* and C_{a^*}, respectively. Let $\tilde{a} = \frac{m}{2n}$.

- Bounds for a^*

We know a^* is the solution for the following equation.

$$\mathbf{E}_{P_{a^*}} N_1 = \frac{n}{m}$$

By Lemma A.2, we have

$$\frac{1}{2a^*} \geq \frac{n}{m}$$

That gives

$$a^* \leq \frac{m}{2n} = \tilde{a} \tag{A.7}$$

Since C_a is decreasing in a, we have

$$C_{a^*} \geq C_{\tilde{a}} > \frac{1}{\sqrt{2\tilde{a}}} = \sqrt{\frac{n}{m}}.$$

For any $j \in \{1, \ldots, m\}$, and $a > 0$, we have

$$\begin{aligned} \mathbf{E}_{P_a} N_1 &= \sum_{k=1}^{\infty} \frac{k^{k+1} e^{-k}}{k!\, C_a} e^{-ak} \\ &\overset{(a)}{\geq} \frac{\sum_{k=1}^{\infty} \frac{k^{k+1} e^{-k}}{k!} e^{-ak}}{1 + \frac{1}{\sqrt{2a}}} \\ &\overset{(b)}{=} \frac{\sum_{k=1}^{\infty} \frac{k^{\frac{1}{2}}}{\sqrt{2\pi} e^{r_k}} e^{-ak}}{1 + \frac{1}{\sqrt{2a}}} \end{aligned} \tag{A.8}$$

Here (a) is attributed to inequality (A.2), step (b) is by Stirling's approximation, and $\frac{1}{12k+1} < r_k < \frac{1}{12k}$. Pick $k_1 = a^{-1/3}$, then the numerator of expression (A.8) can be lower bounded by

$$\begin{aligned} & \sum_{k=\lfloor k_1 \rfloor}^{\infty} \frac{k^{1/2}}{\sqrt{2\pi} e^{r_k}} e^{-ak} \\ \geq & \sum_{k=\lfloor k_1 \rfloor}^{\infty} \frac{k^{1/2}}{\sqrt{2\pi} e^{\frac{1}{12\lfloor k_1 \rfloor}}} e^{-ak} \\ \geq & \frac{1}{\sqrt{2\pi} e^{\frac{1}{12(k_1 - 1)}}} \int_{\lfloor k_1 \rfloor}^{\infty} t^{1/2} e^{-at} dt \end{aligned}$$

Taking the integral from 0 to ∞ and subtracting the part from 0 to k_1 yields the lower bound

$$\begin{aligned} & \frac{1}{\sqrt{2\pi} e^{\frac{1}{12(k_1 - 1)}}} \left(\frac{\Gamma(3/2)}{a^{3/2}} - \int_0^{k_1} t^{1/2} e^{-at} dt \right) \\ \geq & \frac{1}{\sqrt{2\pi} e^{\frac{1}{12(k_1 - 1)}}} \left(\frac{\Gamma(3/2)}{a^{3/2}} - \int_0^{k_1} t^{1/2} dt \right) \\ = & \frac{1}{\sqrt{2\pi} e^{\frac{1}{12(k_1 - 1)}}} \left(\frac{\Gamma(3/2)}{a^{3/2}} - \frac{2}{3a^{1/2}} \right). \end{aligned}$$

Write $r_a = \frac{1}{12(k_1-1)} = \frac{a^{1/3}}{12(1-a^{1/3})}$. By the above calculation, we have a lower bound for the expectation under the tilting distribution. For a^*,

$$\frac{\frac{1}{\sqrt{2\pi}e^{r_{a^*}}}\left(\frac{\Gamma(3/2)}{a^{*3/2}} - \frac{2}{3a^{*1/2}}\right)}{1+\frac{1}{\sqrt{2a^*}}} \leq \mathbf{E}_{a^*} N_1 = \frac{n}{m}.$$

Arranging the terms, we have

$$\begin{aligned}\frac{1}{2a^*} &\leq \frac{n}{m}\left(1+\sqrt{2a^*}\right)e^{r_{a^*}} + \frac{2}{3\sqrt{\pi}} \\ &\overset{(c)}{\leq} \frac{n}{m}\left(1+\sqrt{2\tilde{a}}\right)e^{r_{\tilde{a}}} + \frac{2}{3\sqrt{\pi}}\end{aligned}$$

Here (c) is because $a^* \leq \tilde{a}$ by inequality (A.7). So,

$$a^* \geq \frac{\tilde{a}}{\left(1+\sqrt{2\tilde{a}}\right)e^{r_{\tilde{a}}} + \frac{4}{3\sqrt{\pi}}\tilde{a}}$$

By Taylor expansion, this is no smaller than

$$\begin{aligned}&\frac{\tilde{a}}{\left(1+\sqrt{2\tilde{a}}\right)\left(1+r_{\tilde{a}}+O(r_{\tilde{a}}^2)\right) + \frac{4}{3\sqrt{\pi}}\tilde{a}} \\ =\ & \tilde{a}\left(1 - \frac{r_{\tilde{a}} + \sqrt{2\tilde{a}} + \sqrt{2\tilde{a}}r_{\tilde{a}} + \frac{4}{3\sqrt{\pi}}\tilde{a} + O(r_{\tilde{a}}^2)}{\left(1+\sqrt{2\tilde{a}}\right)\left(1+r_{\tilde{a}}+O(r_{\tilde{a}}^2)\right) + \frac{4}{3\sqrt{\pi}}\tilde{a}}\right) \\ \geq\ & \tilde{a}\left(1 - r_{\tilde{a}} - \sqrt{2\tilde{a}} - \sqrt{2\tilde{a}}r_{\tilde{a}} - \frac{4}{3\sqrt{\pi}}\tilde{a} - O(r_{\tilde{a}}^2)\right)\end{aligned}$$

When $m = o(n)$, $r_{\tilde{a}}$ is the leading term, so

$$a^* \geq \tilde{a}\left(1 - O\left(r_{\tilde{a}}\right)\right) = \frac{m}{2n}\left(1 - O\left(\left(\frac{m}{n}\right)^{\frac{1}{3}}\right)\right)$$

As a result,

$$na^* \log e \geq \left(1 - O\left(\left(\frac{m}{n}\right)^{\frac{1}{3}}\right)\right) \frac{m}{2} \log e$$

Hence we get inequality (2.6).

The above lower bound works when a^* is small (i.e., when m is small compared to n), yet when it is large, the following bound is better. Let $a_0 = \ln \frac{m}{ne}$.

From Lemma A.2,

$$e^{-(a^*+1)} \leq \frac{n}{m}.$$

Then

$$\begin{aligned} e^{a^*} &\geq \frac{m}{ne} = e^{a_0} \\ a^* &\geq a_0 \end{aligned} \tag{A.9}$$

Thus,

$$na^* \log e \geq na_0 \log e = n \log \frac{m}{ne}$$

- Bounds for C_{a^*}

Now we want to lower bound C_{a^*}. Recall inequality (A.6), let term (C) be defined as

$$s_a = \frac{\sum_{l=0}^{\infty} (l+1)^{l+1} e^{-l} e^{-al}/l!}{\sum_{k=0}^{\infty} k^k e^{-k} e^{-ak}/k!}.$$

We have

$$s_{a^*} e^{-(a^*+1)} = \mathbf{E}_{P_{a^*}} N_j = \frac{n}{m} = e^{-(a_0+1)}.$$

It gives

$$e^{-(a^*+1)} = \frac{e^{-(a_0+1)}}{s_{a^*}}.$$

By definition,

$$C_{a^*} \geq 1 + e^{-(a^*+1)} = 1 + \frac{e^{-(a_0+1)}}{s_{a^*}}. \tag{A.10}$$

By Stirling's approxmation, the numerator of s_a is bounded above.

$$\begin{aligned} \sum_{l=0}^{\infty} \frac{(l+1)^{l+1} e^{-l} e^{-al}}{l!} &\leq 1 + \frac{1}{\sqrt{2\pi}} \sum_{l=1}^{\infty} (1+\frac{1}{l})^l \frac{l+1}{\sqrt{l}} e^{-al} \\ &\overset{(d)}{\leq} 1 + \frac{e}{\sqrt{2\pi}} \sum_{l=1}^{\infty} \frac{l+1}{\sqrt{l}} e^{-al} \\ &\leq 1 + \frac{e}{\sqrt{2\pi}} \left(\sum_{l=1}^{\infty} l e^{-al} + \sum_{l=1}^{\infty} e^{-al} \right) \end{aligned} \tag{A.11}$$

where (d) is because $(1+\frac{1}{l})^l$ is bounded above by e for each $l > 0$. We know $\sum_{l=1}^{\infty} l e^{-al}(1-e^{-a})$ is equal to the expectation of a geometric random variable with success probability $1-e^{-a}$, which equals to $1/(1-e^{-a})-1$. And $\sum_{l=1}^{\infty} e^{-al}(1-e^{-a}) = e^{-a}$. Hence, equation (A.11) has the following upper bound

$$1 + \frac{e}{\sqrt{2\pi}} \frac{e^{-a}(2-e^{-a})}{(1-e^{-a})^2}.$$

Using the above inequality and $C_{a^*} \geq 1 + e^{-(a^*+1)}$, we have

$$\begin{aligned} \frac{1}{s_{a^*}} &\geq \frac{1 + e^{-(a^*+1)}}{1 + \frac{e}{\sqrt{2\pi}} \frac{e^{-a^*}(2-e^{-a^*})}{(1-e^{-a^*})^2}} \\ &= 1 - \frac{\frac{e}{\sqrt{2\pi}} \frac{e^{-a^*}(2-e^{-a^*})}{(1-e^{-a^*})^2} - e^{-(a^*+1)}}{1 + \frac{e}{\sqrt{2\pi}} \frac{e^{-a^*}(2-e^{-a^*})}{(1-e^{-a^*})^2}} \\ &= 1 - \frac{\frac{e^2}{\sqrt{2\pi}} \frac{2-e^{-a^*}}{(1-e^{-a^*})^2} - 1}{1 + \frac{e}{\sqrt{2\pi}} \frac{e^{-a^*}(2-e^{-a^*})}{(1-e^{-a^*})^2}} e^{-(a^*+1)} \end{aligned}$$

Multiply $(1-e^{-a^*})^2$ on both the numerator and denominator of the second term, we have the above expression equal to

$$\begin{aligned} & 1 - \frac{\frac{2e^2}{\sqrt{2\pi}} - 1 - (\frac{e^2}{\sqrt{2\pi}} - 2)e^{-a^*} - e^{-2a^*}}{(1-e^{-a^*})^2 + \frac{e}{\sqrt{2\pi}}e^{-a^*}(2-e^{-a^*})} e^{-(a^*+1)} \\ = & \ 1 - \frac{\frac{2e^2}{\sqrt{2\pi}} - 1 - (\frac{e^2}{\sqrt{2\pi}} - 2)e^{-a^*} - e^{-2a^*}}{\frac{e}{\sqrt{2\pi}} + (1-\frac{e}{\sqrt{2\pi}})(1-e^{-a^*})^2} e^{-(a^*+1)}. \end{aligned}$$

The denominator of the second term is lower bounded by 1 since $0 < e^{-a^*} < 1$. Therefore,

$$\begin{aligned} \frac{1}{s_{a^*}} & \geq 1 - \left(\frac{2e^2}{\sqrt{2\pi}} - 1 - (\frac{e^2}{\sqrt{2\pi}} - 2)e^{-a^*} - e^{-2a^*}\right) e^{-(a^*+1)} \\ & \geq 1 - \left(\frac{2e^2}{\sqrt{2\pi}} - 1\right) e^{-(a^*+1)} \\ & \geq 1 - \left(\frac{2e^2}{\sqrt{2\pi}} - 1\right) e^{-(a_0+1)}. \end{aligned}$$

The last inequality is due to inequality (A.9). Now, using inequality (A.10), we have

$$C_{a^*} \geq 1 + \left(1 - c_1 e^{-(a_0+1)}\right) e^{-(a_0+1)}$$

where $c_1 = 2e^2/\sqrt{2\pi} - 1$. From this lower bound on C_a^* and using $a_0 = \log \frac{m}{ne}$, we derive that

$$m \log C_{a^*} \geq m \log \left(1 + \left(1 - O\left(\frac{n}{m}\right)\right) \frac{n}{m}\right).$$

Therefore, inequality (2.7) follows. □

A.3 Proof of Pythagorean Equality

Theorem A.0. *Let $M(k) = k^k e^{-k}/k!$ denote the Stirling ratio measure for $k = 0, 1, \ldots$ as defined before. Let $M^m = \otimes_{j=1}^m M$ assign a product measure to $\underline{N} =$*

$(N_1, \ldots, N_m)$. Let M_{cond} be the probability distribution on $\underline{N}$ obtained from conditioning on $\frac{1}{m}\sum_{j=1}^{m} N_j = \alpha$ (suppose α is a value that the average of the N_j's is possible to obtain). Define $P_a(k) = M(k)\frac{e^{-ak}}{C_a}$ for an a chosen by the condition $\mathbf{E}_{P_a} N_1 = \alpha$ (suppose such an a can be obtained). Let $\mathcal{C}_\alpha$ be a class of distributions with the expected value of the average of N_j equal to α

$$\mathcal{C}_\alpha = \{P : \mathbf{E}_P \frac{1}{m}\sum_{j=1}^{m} N_j = \alpha\}.$$

Then, $Q_a = \otimes_{j=1}^{m} P_a$ is the information projection of M on $\mathcal{C}_\alpha$ in the sense of uniquely minimizing $D(Q||M)$ among all Q in C_α. In fact,

$$D(Q||M^m) = D(Q||Q_a) + D(Q_a||M^m)$$

for all $Q \in C_\alpha$. In particular, we have

$$D(M_{cond}||M^m) = D(M_{cond}||Q_a) + D(Q_a||M^m).$$

Therefore, equality (2.2) *stands.*

This is similar to what has been shown in [12], [13], and [14]. Theorem A.0 says the tilted distribution is closest to the original distribution in relative entropy among all distributions with the expected value of a function equal to α. Hence it is the redundancy minimizing distribution over the class of distributions with a given moment condition. Note that $D(Q||M^m)$ and $D(Q_a||M^m)$ could be negative since M^m is not a probability measure, but $D(Q||Q_a) \geq 0$ for all $Q \in C_\alpha$.

Proof. For any $Q \in C_\alpha$ and $m \geq 1$,

$$\begin{aligned}
& D(Q||M^m) \\
= & \sum_{N_1,\ldots,N_m} Q(N_1,\ldots,N_m) \log \frac{Q(N_1,\ldots,N_m)}{Q_a(N_1,\ldots,N_m)} \\
& + \sum_{N_1,\ldots,N_m} Q(N_1,\ldots,N_m) \log \frac{Q_a(N_1,\ldots,N_m)}{M^m(N_1,\ldots,N_m)} \\
= & D(Q||Q_a) + \mathbf{E}_Q\left(\log e^{-a\sum_{j=1}^m N_j}\right) \\
\overset{(a)}{=} & D(Q||Q_a) + \mathbf{E}_{Q_a}\left(\log e^{-a\sum_{j=1}^m N_j}\right) \\
\overset{(b)}{=} & D(Q||Q_a) + D(Q_a||M^m) \\
\geq & D(Q_a||M^m).
\end{aligned}$$

Here (a) is because Q_a and Q are both in the convex set $\mathcal{C}_\alpha$, and (b) holds since $Q_a(N_j) = M(N_1,\ldots,N_m)\frac{e^{-a\sum_{j=1}^m N_j}}{C_a^m}$. □

A.4 Redundancy

Theorem A.4. *Consider the family of distributions that makes $N_1,\ldots,N_m$ independent Poisson $\lambda_1,\ldots,\lambda_m$. Let $\lambda_{sum} = \sum_{j=1}^m \lambda_j$, and let $\mathcal{P}^m_{\lambda_{sum}}$ denote the family. The redundancy of using a tilted Stirling ratio distribution Q_a on the counts generated by any $P^m_{\underline{\lambda}} \in \mathcal{P}^m_{\lambda_{sum}}$ is mainly*

$$r(Q_a, P_{\underline{\lambda}}) = \underbrace{\left((-\frac{m}{2} + a\lambda_{sum})\log e + m\log C_a\right)}_{(A)},$$

with the error bounded by

$$\sum_{j=1}^m (\frac{1}{3\lambda_j^2} + \frac{5}{6\lambda_j})\log e.$$

Moreover, the minimizer of the redundancy is a^*, with a^* chosen by making $\mathbf{E}_{P_a} N_1 = \lambda_{sum}/m$.

When $m = o(\lambda_{sum})$, term (A) satisfies the following inequality

$$0 \leq \left| (A) - \frac{m}{2} \log \frac{\lambda_{sum}}{m} \right| \leq m \log(1 + \sqrt{\frac{m}{\lambda_{sum}}}). \tag{A.12}$$

When $\lambda_{sum} = o(m)$, term (A) satisfies the following inequality

$$\begin{aligned} & m \log \left(1 + \frac{\lambda_{sum}}{m} \right) - \lambda_{sum} \log e \\ \leq & \left| (A) - \left(\lambda_{sum} \log \frac{m}{\lambda_{sum}} - \frac{m}{2} \log e \right) \right| \\ \leq & \frac{1}{2\sqrt{\pi}} \frac{\lambda_{sum}^2 e^2}{m - \lambda_{sum} e} \log e. \end{aligned} \tag{A.13}$$

Remark 8: The expression (A) for the redundancy agrees with the regret $a^* \lambda_{sum} \log e + m \log C_{a^*}$ except for the $-\frac{m}{2} \log e$. This difference is due to the difference in the numerator in which the expected $\log P_{\underline{\lambda}}(\cdot)$ is used in the redundancy, and $\log P_{\underline{\lambda}}(\cdot)$ is used in regret. Here the expected difference $\mathbf{E} \log \frac{P_{\underline{\lambda}}(\cdot)}{P_{\underline{\lambda}}(\cdot)}$ is shown to be near $-\frac{m}{2} \log e$. A similar phenomenon occurs in [38].

Proof. The first part of the proof follows Lemma 3 in [7], and the second part resembles the proof of Theorem 3.2.

$$\begin{aligned} & \mathbf{E}_{\underline{\lambda}} \ln \frac{\prod_{j=1}^m P_{\lambda_j}(N_j)}{Q_a(\underline{N})} \\ = & \sum_{j=1}^m (\lambda_j \ln \lambda_j) - \sum_{j=1}^m \mathbf{E}_{\lambda_j} (N_j \ln N_j) + a\lambda_{sum} \\ & + m \ln C_a \end{aligned} \tag{A.14}$$

Following Lemma 3 in [7], by Taylor's expansion, for each j,

$$\begin{aligned}
& \mathbf{E}_{\lambda_j}(N_j \ln N_j) \\
\geq\ & \lambda_j \ln \lambda_j + \mathbf{E}_{\lambda_j}(N_j - \lambda_j)(1 + \ln \lambda_j) \\
& + \mathbf{E}_{\lambda_j} \frac{1}{2}(N_j - \lambda_j)^2 \frac{1}{\lambda_j} + \frac{1}{6}\mathbf{E}_{\lambda_j}(N_j - \lambda_j)^3(-\frac{1}{\lambda_j^2}) \\
=\ & \lambda_j \ln \lambda_j + \frac{1}{2} - \frac{1}{6\lambda_j}.
\end{aligned}$$

We also know by Jensen's Inequality that

$$\mathbf{E}_{\lambda_j}(N_j \ln N_j) \geq \lambda_j \ln \lambda_j.$$

Hence,

$$\mathbf{E}_{\lambda_j}(N_j \ln N_j) \geq \lambda_j \ln \lambda_j + \frac{1}{2} + \max(-\frac{1}{6\lambda_j}, -\frac{1}{2}).$$

And by inequality (30) in [7],

$$\begin{aligned}
& \mathbf{E}_{\lambda_j}(N_j \ln N_j) \\
\leq\ & \lambda_j \ln \lambda_j + (\mathbf{E}_{\lambda_j} N_j - \lambda_j)(1 + \ln \lambda_j) \\
& + \frac{\mathbf{E}_{\lambda_j}(N_j - \lambda_j)^2}{2\lambda_j} - \frac{\mathbf{E}_{\lambda_j}(N_j - \lambda_j)^3}{6\lambda_j^2} \\
& + \frac{\mathbf{E}_{\lambda_j}(N_j - \lambda_j)^4}{3\lambda_j^3} \\
=\ & \lambda_j \ln \lambda_j + \frac{1}{2} + \frac{1}{3\lambda_j^2} + \frac{5}{6\lambda_j}.
\end{aligned}$$

Therefore,

$$\begin{aligned} & -\left(\sum_{j=1}^{m}\frac{1}{3\lambda_j^2}+\frac{5}{6\lambda_j}\right) \\ \leq\ & \mathbf{E}_{\underline{\lambda}}\ln\frac{\prod_{j=1}^{m}P_{\lambda_j}(N_j)}{Q_a(\underline{N})} \\ & -\left(-\frac{m}{2}+a\lambda_{sum}+m\ln C_a\right) \\ \leq\ & \min\left(\sum_{j=1}^{m}\frac{1}{6\lambda_j},\frac{m}{2}\right). \end{aligned}$$

The fact that a^* is the minimizer can be easily seen by taking partial derivative with respect to a for the redundancy expression (A.14). The two inequalities are attributed to Lemma A.1, by picking $a = m/(2\lambda_{sum})$ and $a = \ln(m/\lambda_{sum}e)$ respectively. □

A.5 Proof of Theorem 2.3

Proof. The MLE for an envelope class is the following

$$\hat{\lambda}_j = \arg\sup_{\lambda_j \leq nf(j)} P_{\lambda_j}(N_j) = N_j \wedge nf(j),$$

where $\wedge$ denotes the minimum.

We formulate a tilted distribution by multiplying the exponential tilting factor e^{-aN_j} for each $j \in \{1,\ldots,m\}$ and normalize it.

$$P_a(N_j) = \begin{cases} \frac{N_j^{N_j}e^{-N_j}}{N_j!}\frac{e^{-aN_j}}{C_{a,j}} & \text{if } N_j \leq nf(j) \\ \frac{(nf(j))^{N_j}e^{-nf(j)}}{N_j!}\frac{e^{-aN_j}}{C_{a,j}} & \text{if } N_j > nf(j) \end{cases}$$

where $C_{a,j} = \sum_{N_j \leq nf(j)}\frac{N_j^{N_j}e^{-N_j}}{N_j!}e^{-aN_j} + \sum_{N_j > nf(j)}\frac{(nf(j))^{N_j}e^{-nf(j)}}{N_j!}e^{-aN_j}$.

The regret of using independent P_a for each N_j in $\underline{N} \in S_{m,n}$ is

$$\log \prod_{j=1}^{m} \frac{P_{\hat{\lambda}_j}(N_j)}{P_a(N_j)} = na \log e + \sum_{j=1}^{m} \log C_{a,j}. \tag{A.15}$$

Again, a^* minimizes expression (A.15).

For each j and any positive a,

$$\begin{aligned} C_{a,j} &= \sum_{N_j \le \lfloor nf(j) \rfloor} \frac{N_j^{N_j} e^{-N_j}}{N_j!} e^{-aN_j} \\ &\quad + \sum_{N_j > nf(j)} \frac{(nf(j))^{N_j} e^{-nf(j)}}{N_j!} e^{-aN_j}. \end{aligned}$$

The sum only depends on the envelope function $f(j)$ for given a and j.

Since $(nf(j))^x e^{-nf(j)} \le x^x e^{-x}$ for all $x > 0$, for any symbol j with $N_j > nf(j)$, we have

$$\frac{(nf(j))^{N_j} e^{-nf(j)}}{N_j!} e^{-aN_j} \le \frac{N_j^{N_j} e^{-N_j}}{N_j!} e^{-aN_j}.$$

Hence we have,

$$C_{a,j} \le \sum_{N_j=0}^{\infty} \frac{N_j^{N_j} e^{-N_j}}{N_j!} e^{-aN_j} \le 1 + \sqrt{\frac{1}{2a}}.$$

The second inequality is due to Lemma A.1.

However, if $nf(j)$ is small, the following upper bound is better. For $N_j \le \lfloor nf(j) \rfloor$,

$$\begin{aligned} \sum_{N_j \le \lfloor nf(j) \rfloor} \frac{N_j^{N_j} e^{-N_j}}{N_j!} e^{-aN_j} &\le \sum_{N_j \le \lfloor nf(j) \rfloor} \frac{N_j^{N_j}}{N_j!} \\ &\le \sum_{N_j \le \lfloor nf(j) \rfloor} \frac{(nf(j))^{N_j}}{N_j!}. \end{aligned}$$

For the second partial sum, we also have

$$\begin{aligned} & \sum_{N_j > nf(j)} \frac{(nf(j))^{N_j} e^{-nf(j)}}{N_j!} e^{-aN_j} \\ \leq & \sum_{N_j > nf(j)} \frac{(nf(j))^{N_j}}{N_j!}. \end{aligned}$$

Deduce,

$$C_{a,j} \leq \sum_{N_j=0}^{\infty} \frac{(nf(j))^{N_j}}{N_j!} = e^{nf(j)}.$$

Hence for any given a, j and $L \in \{1, 2, \ldots, m\}$, the following upper bound holds.

$$\begin{aligned} & na \log e + \sum_{j=1}^{m} \log C_{a,j} \\ \leq & \; na \log e \\ & + \log \left(\prod_{j=1}^{L} \left(1 + \sqrt{\frac{1}{2a}} \right) \prod_{j=L+1}^{m} \left(e^{nf(j)} \right) \right) \\ = & \; na \log e + L \log \left(1 + \sqrt{\frac{1}{2a}} \right) \\ & + \left(\sum_{j=L+1}^{m} nf(j) \right) \log e. \end{aligned}$$

Let $a = \frac{L}{2\left(n - \sum_{j>L} nf(j)\right)}$, the result follows. □

Appendix B

Supplementary materials

B.1 Incompatibility of P_n

$$\begin{aligned}
&\sum_{x\in\mathcal{A}} P_{n+1}(X_1,\ldots,X_n,X_{n+1}=x)\\
&= \sum_{x\in\mathcal{A}} \frac{1}{\binom{n+1}{N_1^n..N_x^n+1..N_m^n}} \frac{Q_a(N_1^n,\ldots,N_x^n+1,\ldots,N_m^n)}{Q_a(S_{m,n+1})}\\
&= \underbrace{\frac{1}{\binom{n}{N_1^n..N_x^n..N_m^n}} \frac{M^m(\underline{N}^n)}{M^m(S_{m,n})}}_{(A)} \underbrace{\frac{M^m(S_{m,n})}{M^m(S_{m,n+1})}}_{(B)}\\
&\quad \underbrace{\sum_{x\in\mathcal{A}} \left(\frac{N_x^n+1}{n+1}\frac{M(N_x^n+1)}{M(N_x^n)}\right)}_{(C)}.
\end{aligned}$$

Term (A) equals to the distribution of the count vector $\underline{N}^n$ conditioning on its total equal to n through expression (2.10). Hence, it suffices to check whether the rest equals to 1. This is obviously not true, since term (C) equals

$$\frac{e^{-1}}{n+1}\sum_{x\in\mathcal{A}} \frac{(N_x^n+1)^{N_x^n+1}}{N_x^{n\,N_x^n}}$$

which depends on the specific value of the count vector $\underline{N}^n$, while the ratio $M^m(S_{m,n})/M^m(S_{m,n+1})$ is a constant given m and n. Hence the P_n's are not compatible.

B.2 Computation complexity

We use a two pass code to implement the encoding distribution. As a preliminary step, the counts are calculated for each symbol, and computation complexity is $O(n\log m)$. Then the encoder initially has the location and values of the non-zero counts.

The first pass is to code the counts using the tilted Stirling ratio distribution by

arithmetic coding [39]. This requires recursively calculating the cumulative probabilities to the left of $N_1, \ldots, N_j$ as follows. Define the cumulative marginal probability of P_a as $P_{a,1}^{cum}(k) = \sum_{i=0}^{k-1} P_a(i)$. First, $P_{a,1}^{cum}(N_1)$ is 0 for $N_1 = 0$ and otherwise $P_{a,1}^{cum}(N_1) = \sum_{i=0}^{N_1-1} P_a(i)$. Then for $j \geq 1$,

$$P_{a,j+1}^{cum}(N_1, \ldots, N_j, N_{j+1})$$

$$= \begin{cases} P_a^{cum(N_\cdot, N_\cdot)} P_{a,j+1}^{cum}(N_1, \ldots, N_j, N_{j+1}) & \text{if} \quad N_{j+1} = 0 \\ P_a^c \begin{cases} P_{a,j}^{cum}(N_1, \ldots, N_j) & N_{j+1} > 0 \\ + \begin{cases} P_{a,j}^{cum}(N_1, \ldots, N_j) \\ + Q_a^j(N_1, \ldots, N_j) P_{a,1}^{cum}(N_{j+1}) \end{cases} \end{cases} \end{cases}$$

If $n = o(m)$, it is only at th[illegible] the cumulative probability needs to be updated. To retrieve those values of the positive counts, one needs to know the positions of those positive counts. This requires $O(n \log m)$ computational complexity. If $m = o(n)$, only linear time in m is needed.

The second pass is to use arithmetic coding to encode the string given the counts. Initialize with $P(X_1|N_1, \ldots, N_m) = N_{X_1}/n$, which is evaluated at X_1. The corresponding cumulative probability to the left of X_1 is

$$F_-(X_1|N_1, \ldots, N_m) = \frac{L_{X_1}}{n},$$

where L_{X_1} is the counts of symbols to the left of X_1. For the next step, the relevant counts are for $X_2, \ldots, X_n$. Accordingly we decrement the count of N_{X_1} and decrement the cumulative counts L_x for all $x > X_1$. Then for $i \geq 1$, having decremented by 1 the counts $N_{X_i}^{rem}$ and the cumulative counts L_x^{rem} for $x > X_i$, we proceed to set the conditional probability of the next symbol given the past and the counts (as given

in Section 2.3.6) to be the relative frequency of x in the remaining string

$$Prob(X_{i+1}|X_1,\ldots,X_i,(N_1,\ldots,N_m)) = \frac{N^{rem}_{X_{i+1}}}{n-i}.$$

where $N^{rem}_{X_{i+1}} = N_{X_{i+1}} - N_{X_{i+1},i}$. And this associate cumulative conditional probability to the left of X_{i+1} is

$$F_{-}(X_{i+1}|X_1,\ldots,X_i,(N_1,\ldots,N_m)) = \frac{L^{rem}_{X_{i+1}}}{n-i}.$$

Arithmetic coding requires calculation of the following probabilities

$$\begin{aligned} & Q^{cum}(X_1,\ldots,X_i,X_{i+1}|(N_1,\ldots,N_m)) \\ = \; & Q^{cum}(X_1,\ldots,X_i|(N_1,\ldots,N_m)) \\ & +P_i(X_1,\ldots,X_i|(N_1,\ldots,N_m)) \\ & F_{-}(X_{i+1}|X_1,\ldots,X_i,(N_1,\ldots,N_m)). \end{aligned}$$

Note that for each i, what is needed is the value of $L^{rem}_{X_{i+1}}$ which requires the position of X_{i+1} in the sorted list of the remaining symbols. This requires $\log n$ computation time for each symbol. Therefore the computational complexity is $O(n\log n + n\log m)$ if $n = o(m)$, and $O(n\log n)$ if $m = o(n)$. These calculations are scaled each step as in Pasco [40] or Rissanen, Langdon [41] to avoid underflow or outflow.

In a nutshell, the total computation complexity for this two pass code is $O\left(n(\log n + \log m)\right)$.

B.3 Approximation of c

$$\begin{aligned} c &= \frac{m}{k}\left(\frac{\frac{\partial^2 C_a}{\partial a^2}}{C_a} - \left(\frac{\frac{\partial C_a}{\partial a}}{C_a}\right)^2\right)\Bigg|_{a=a_\eta^*} \\ &\leq \eta\left(\frac{\frac{\partial^2 C_a}{\partial a^2}}{C_a}\right)\Bigg|_{a=a_\eta^*}. \end{aligned}$$

A similar argument as in the proof of Lemma A.2 yields an upper bound for the first term

$$\begin{aligned} \frac{\frac{\partial^2 C_a}{\partial a^2}}{C_a} &\leq \frac{3}{(2a)^2} + \frac{1}{\sqrt{2\pi}}\left(\frac{3}{e}\right)^{3/2}\frac{1}{2a} \\ &\leq 3C_a^4 + \frac{1}{\sqrt{2\pi}}\left(\frac{3}{e}\right)^{3/2} C_a^2 \\ &< \frac{7}{2}C_a^4. \end{aligned}$$

The second last inequality is by Lemma A.1.

Hence, we have an upper bound for the Laplace approximation of the regret

$$\begin{aligned} &\log e^{a_\eta^* k} C_{a_\eta^*}^m + \log\frac{m}{2} + \frac{1}{2}\log\frac{k}{2\pi} + \frac{1}{2}\log\left(\frac{7}{4}\eta C_{a_\eta^*}^4\right) \\ <\ &\log e^{a_\eta^* k} C_{a_\eta^*}^m + \frac{3}{2}\log\frac{m}{2} + 2\log C_{a_\eta^*}. \end{aligned}$$

Thus, the extra regret above the optimal level by using $Q(\underline{N})$ is approximately no more than $\frac{3}{2}\log\frac{m}{2} + 2\log C_{a_\eta^*}$ bits.

Similar argument can show that averaging over the two parameters tilting distribution $Q_{a,b}$ can lead to a distribution that achieves regret not much larger than the minimizing value if the actual total count and tail sum were known beforehand.

Bibliography

[1] Y. M. Shtarkov, "Universal sequential coding of single messages," *Problems of Information Transmission*, vol. 23, no. 3, pp. 175–186, 1987.

[2] S. Boucheron, A. Garivier, and E. Gassiat, "Coding on countably infinite alphabets," *IEEE Transactions on Information Theory*, vol. 55, no. 1, Jan 2009.

[3] D. Bontemps, "Universal coding on infinite alphabets: exponentially decreasing envelopes," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1466–1478, 2011.

[4] Y. Leng and Y. Wei, Eds., *ZhongHua ZiHai.* Zhonghua Press, 1994.

[5] Wikipedia. (2015, Feb) Chinese characters. [Online]. Available: http://en.wikipedia.org/wiki/Chinese_characters

[6] X. Yang and A. Barron, "Compression and predictive distributions for large alphabet i.i.d and markov models," *IEEE International Symposium on Information Theory*, 2014.

[7] Q. Xie and A. R. Barron, "Minimax redundancy for the class of memoryless sources," *IEEE Transactions on Information Theory*, vol. 43, pp. 646–657, May 1997.

[8] P. Kontkanen and P. Myllymäki, "A linear-time algorithm for computing the multinomial stochastic complexity," *Inf. Process. Lett.*, vol. 103, no. 6, pp. 227–233, Sep. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.ipl.2007.04.003

[9] A. R. Barron, T. Roos, and K. Watanabe, "Bayesian properties of normalized maximum likelihood and its fast computation," *IEEE International Symposium on Information Theory*, 2014.

[10] R. E. Krichevsky and V. K. Trofimov, "The performance of universal encoding," *IEEE Transactions on Information Theory*, vol. 27, no. 2, pp. 199–207, 1981.

[11] W. Feller, *An Introduction to Probability Theory and Its Applications.* Wiley, 1950, vol. 1.

[12] I. Csiszar, "I-divergence geometry of probability distributions and minimization problems," *The Annals of Probability*, vol. 3, no. 1, pp. 146–158, Feb 1975.

[13] ——, "Sanov property, generalized I-projection and a conditional limit theorem," *The Annals of Probability*, vol. 12, no. 3, pp. 768–793, Jan 1984.

[14] J. V. Campenhout and T. Cover, "Maximum entropy and conditional probability," *IEEE Transactions on Information Theory*, vol. 27, no. 4, July 1981.

[15] A. Orlistsky, N. P. Santhanam, and J. Zhang, "Always good turing: Asymptotically optimal probability estimation," *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Sciene*, 2003.

[16] W. Szpankowski and M. J. Weinberger, "Minimax redundancy for large alphabets," *Information Theory Proceedings*, June 2010.

[17] L. A. Adamic. Zipf, power-laws and pareto – a ranking tutorial. [Online]. Available: http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html

[18] T. J. Tjalkens, F. M. Willems, and Y. M. Shtarkov, "Multi-alphabet universal coding of memoryless sources," *Problems of Information Transmissions*, vol. 31, no. 2, pp. 114–127, 1995.

[19] Q. Xie and A. R. Barron, "Asymptotic minimax regret for data compression, gambling, and prediction," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 431–445, March 2000.

[20] A. Orlistsky and N. P. Santhanam, "Speaking of infinity," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2215–2230, October 2004.

[21] T. J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, vol. 40, December 1953.

[22] A. B. Wagner, P. Viswanath, and S. R. Kulkarni, "Probability estimation in the rare-events regime," *IEEE Transactions on Information Theory*, vol. 57, no. 6, pp. 3207–3229, June 2011.

[23] S. Kullback, *Information Theory and Statistics.* Wiley, New York, 1959.

[24] H. Robbins, "A remark of stirling's formula," *The American Mathematical Monthly*, vol. 62, no. 1, pp. 26–29, Jan 1955.

[25] N. G. D. Bruijn, *Asymptotic Methods in Analysis.* New York: Dover, 1958.

[26] P. Bartlett, P. Grunwald, P. Harremoes, F. Hedayati, and W. Kotlowski, "Horizon-independent optimal prediction with log-loss in exponential families," *arXiv preprint arXiv:1305.4324*, 2013.

[27] T. Roos and J. Rissanen, "On sequentially normalized maximum likelihood models," in *Workshop on Information Theoretic Methods in Science and Engineering (WITMSE-08)*, August 2008.

[28] Wikipedia. (2015, Feb) Classic of poetry. [Online]. Available: https://en.wikipedia.org/wiki/Classic_of_Poetry

[29] ——. (2015, January) Gb18030 stardard. [Online]. Available: http://zh.wikipedia.org/wiki/GB_18030

[30] (2008, Nov) 2007 report on language use in china. [Online]. Available: http://www.china-language.gov.cn/14/2008_11_17/1_14_3890_0_1226884790921.html

[31] X. Yang and A. Barron, "Large alphabet compression and predictive distributions through poissonization and tilting," *arXiv:1401.3760v1 [stat.ME]*, 2013.

[32] J. Acharya, A. Jafarpour, A. Orlitsky, and A. T. Suresh, "Poissonization and universal compression of envelope classes," *IEEE International Symposium on Information Theory*, 2014.

[33] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: basic properties," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 653–664, 1995.

[34] J. G. Cleary and I. H. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Transactions on Communications*, vol. 32, no. 396-402, 1984.

[35] K. Sadakane, T. Okazaki, and H. Imai, "Implementing the context tree weighting method for text compression," in *Proceedings of the Conference on Data Compression*, ser. DCC '00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 123–. [Online]. Available: http://dl.acm.org/citation.cfm?id=789087.789787

[36] J. Archarya and H. Das, "Tight bounds for universal compression of large alphabets," *IEEE International Symposium on Information Theory*, 2013.

[37] J. Rissanen, "A Universal Data Compression Syetem," *IEEE Transactions on Information Theory*, vol. 29, no. 5, pp. 656–664, 1983.

[38] B. S. Clarke and A. R. Barron, "Jefferys' prior is asymptotically least favorable under entropy risk," *Journal of Statistical Planning and Inference*, vol. 41, pp. 37–60, August 1994.

[39] F. Jelinek, *Probabilistic Information Theory.* McGraw-Hill Book Co., Inc, New York, 1968.

[40] R. C. Pasco, "Source coding algorithms for fast data compression," Ph.D. dissertation, Stanford University, 1976.

[41] J. Rissanen and G. Langdon, "Arithmetic coding," *IBM Journal of Research and Development*, vol. 29, no. 3, pp. 198–203, 1979.