CHAPTER 2

Universal coding I: parametric setting

2.1. Introduction

In the preceding chapter, we realized that provided the source statistics are known (and computable), the best compression ratio, is well-defined and there exists effective methods that can approach arbitrarily well this best compression ratio. Note that this does not solve the practical problem faced by system engineers and digital library managers. Even if they may assume that texts are produced by some (stochastic stationary ergodic) source, they cannot assume that they do know the source statistics. Actually they should not assume that they deal with a single stationary source: a digital library may contain texts in English and Italian. At the very best, there are two sources corresponding to the two languages. Nevertheless, text compression algorithms like `compress`, `gzip` or `bzip2` deal with those sources and their performances do not seem to degrade when applied to languages that were completely unknown to their designers.

Those text compression algorithms are actually *universal* in the following sense: for a large class of sources, which have a well-defined entropy rate, those algorithms eventually almost surely achieve the best possible compression ratio (i.e. the entropy rate).

The Chapter is organized as follows. In Section 2.4, we present most straightforward approach to coding with unknown statistics: postulate a parametric model Θ, estimate the parameters using the data that have to be compressed, code the estimate and the data using the estimated probability as a coding probability. The redundancy of the plug-in methodology is easily related with the hardness of estimating parameters in Θ. Despite its simplicity, the plug-in methodology calls for improvements. In the next Section, we point out that parametric data compression problems may be analyzed in both Bayesian and minimax perspectives, as a matter of fact solutions to both problems coincide. In Section 2.9, we describe Krichevsky-Trofimov mixtures for memoryless sources, provide bounds on their pointwise regret. In Section **??**, compression problems are investigated in a non-parametric context: the compressor now just knows the source belongs to (or is well approximated by) one model among many.

The compression problem is now closely related to model selection issues in non-parametric statistics. Ideas stemming from data compression like the Minimum Description Length principle (Rissanen) and the Context-Tree-Weighting method (Tjalkens, Willems and Shtarkov) have deeply influenced non-parametric statistics.

2.2. Motivations : the price of misspecified probability

From a statistical viewpoint, one of the charms of arithmetic coding, or rather of using nearly ideal codeword lengths pertains to the fact that it makes easy the determination of the redundancy of codes tailored towards some source $\mathbb{Q}$ (a consistent family of probability distributions $(Q^n)_n$) while the true source distribution is $\mathbb{P}$ (defined by the consistent family $(P^n)_n$).

The probability distribution $\mathbb{Q}$ is called a (the) *coding probability* associated with (f, ϕ), if the code (f, ϕ) has nearly ideal codelength against $\mathbb{Q}$, that is:

$$\ell(f(w)) \leq -\log \mathbb{Q}(w) + c$$

for some constant c,

Definition 2.2.1. [Redundancy of a code with respect to a source] The difference between the entropy of P^n and the mean (f, ϕ)codeword length under P^n is called the nth order redundancy of (f, ϕ) with respect to $P^n(\mathbb{P})$.

If if the code (f, ϕ) has nearly ideal codelength against Q^n for inputs of length n, then the redundancy with respect to P^n satisfies:

$$\mathbb{E}_{P^n}\left[-\log Q^n\{X_1^n\} + c + \log P^n\{X_1^n\}\right] = D(P^n \parallel Q^n) + c.$$

What we have just noted is a simple and tight relationship between the redundancy and relative entropy. This will prove useful when trying to assess the performances of adaptive coding techniques .

2.3. Prefix codes for integers

It is often useful to have a good (uniquely-decodable) code for the integers (that is for $\mathbb{N}$). It is tempting to code an integer n by its binary expansion $n = \sum_{i=0}^{imax} n_i 2^i$ where n_i equals either 0 or 1, and $imax$ satisfies $imax \leqslant \log_2 n < imax + 1$, for $n \geqslant 1$ and $imax = 0$ for $n = 0$ (that is $imax = \lfloor \log_2(n \vee 1) \rfloor$). The length of the binary expansion $\text{bin}(n) = n_0 n_2 \ldots n_{imax}$ is just $imax + 1$. Unfortunately, binary expansions do not constitute a uniquely decodable code: for example 101 may be parsed either as the binary expansion of 5, or as the concatenation of the binary expansions of 2 and 1. Not too surprinsingly if we denote by $\ell(n)$ the length of the binary expansion of n, the sequence $\ell(n) = 1 + \lfloor \log_2(n \vee 1) \rfloor$ does not satisfy the Kraft-McMillan condition.

A simple device allows to get around this difficulty. Encode n as

$$n_0 n_0 n_1 n_1 \ldots n_{imax} n_{imax} 01$$

that is by repeating twice every bit in the binary expansion and terminating by a sequence of distinct bits. Such a code is prefix, and the codeword lengths do satisfy the Kraft-McMillan inequality. However the coderword length is $2(1 + \lfloor \log_2(n \vee 1) \rfloor) + 2 = 2\ell(n) + 2$.

Rather than applying this doubling device to the binary expansion of n, it is wise to apply it to the length of the binary expansion $\ell(n)$, and to concatenate the prefix encoding of $\ell(n)$ with the binary expansion of n. This provides us with a prefix encoding of integers and the length of the encoding of n is now

$$\ell(n) + 2\ell(\ell(n)) + 2 = 1 + \lfloor \log_2(n \vee 1) \rfloor + 2(1 + \lfloor \log_2(\ell(n) + 1) \rfloor).$$

The codeword length of n is thus around $\log_2 n + 2 \log_2 \log_2 n$.

Indeed this device can be iterated until we meet a number with binary encoding reduced to 1 symbol. Let $\ell^{(k+1)}(n) = \ell\left(\ell^{(k)}(n)\right)$ and $\ell^{(0)}(n) = n$. Let $r(n)$ denote the smallest integer k such that $\ell^{(k)}(n) \leqslant 2$ (note that for $n > 2$, $\ell(n) < n$). The iterated procedure would assign the following codeword length to n:

$$\underbrace{\text{bin}(\ell^{(0)}(n))\,\text{bin}(\ell^{(1)}(n))\,\ldots\,\text{bin}(\ell^{(r(n)-1)}(n))} + 01 + \text{repeated}\ \ \text{bin}(\ell^{(r(n))}(n)).$$

For $n = 100$, the binary expansion is 1100100, $\ell^{(1)}(n) = 7$, the binary expansion of $\ell^{(1)}(n)$ is 111, hence $\ell^{(2)}(n) = 3$ has binary expansion 11, $\ell^{(3)}(n) = 2$, hence $r(n) = 3$. The encoding is

$$1100100\,111\,11\,01\,1100.$$

For such a value of n, the overhead is quite significant...

Henceforth we will code integers using the sub-optima prefixed code of length $\ell(n) + 2 + 2\ell^{(2)}(n)$.

2.4. Plug-in codes codes

Consider the following problem:

> Design a code (f, ϕ) that takes as input words generated by a memoryless source $\mathbb{P}_\theta$ in such a way that for any choice of θ among the probability laws on $\mathcal{X}$, the expected length of $f(X_1^n)$ under $\mathbb{P}_\theta$ is within $O(\log n)$ from $nH(\mathbb{P}_\theta)$, i.e. such that (f, ϕ) has redundancy $O(\log n)$,with respect to all memoryless sources.

The plug-in approach proceeds by first estimating $\mathbb{P}_\theta$ using the empirical distribution $\mathbb{P}_{\hat{\theta}}$, i.e.

$$\mathbb{P}_{\hat{\theta}}\{a\} = \frac{1}{n} \sum_{i \leq n} \mathbb{1}_{x_i = a} \ ,$$

encode a representation of $\mathbb{P}_{\hat{\theta}}$ (using a prefix code) and then code X_1^n using $\mathbb{P}_{\hat{\theta}}^{\otimes n}$ as the coding distribution. Note that we take advantage of the fact that Huffman coding and arithmetic coding take source distributions as parameters.

Assume that in the last step we use arithmetic coding. Then according to the previous remarks, the redundancy of the code with respect to $\mathbb{P}_\theta$ is upper-bounded by

$$D(\mathbb{P}_\theta^n \parallel \mathbb{P}_{\hat{\theta}}^n)$$

(here we consider the image of $\mathbb{P}_\theta$ on the first n symbols) plus the length of the representation of $\mathbb{P}_{\hat{\theta}}$.

Note that the latter may be represented by $|\mathcal{X}| - 1$ numbers from $\{0, \ldots n\}$. Hence, $\mathbb{P}_{\hat{\theta}}$ may be represented by

$$(|\mathcal{X}| - 1)(\lceil \log_2(n) \rceil + 2 \log_2(1 + \log_2(n)) + 1) \text{ bits,}$$

using a simple prefix code for the integers (see Exercise).

To check whether our goal is achieved, we just have to notice that as $\mathbb{P}_{\hat{\theta}}$ is a Maximum-Likelihood estimator: for each x_1^n,

$$-\log \frac{\mathbb{P}_{\hat{\theta}}\{x_1^n\}}{\mathbb{P}_\theta\{x_1^n\}} \leq 0$$

Hence, for memoryless sources over alphabets of size $d + 1$, the redundancy of plug-in compressors is less than

$$(|\mathcal{X}| - 1)(\lceil \log_2(n) \rceil + 2 \log_2(1 + \log_2(n)) + 1) = (|\mathcal{X}| - 1)(\lceil \log_2(n) \rceil) + o(\log n) \text{ bits.}$$

This redundancy should be compared with the optimal average codelength for words of length n. By Shannon noiseless coding theorem, the latter belongs to the interval $[nH(P_\theta), nH(P_\theta) + 1]$. Hence the redundancy of our two-steps would-be universal coder is much smaller than the average optimal codelength.

We may naturally wonder whether such a result could be extended to more sophisticated models (for example Markov chains of order k). We should also check whether such a redundancy can be achieved without resorting to Maximum-Likelihood-Estimators. Finally, we would like to avoid a two-pass method, and perform coding in one pass.

EXERCISE 2.5. Check that the maximum likelihood estimator coincides with the empirical distribution (this can be viewed as a consequence of the positivity of relative entropy).

2.6. Redundancies

Let us introduce some criteria that could be used to assess the performance of universal coding schemes. In this section Θ denotes a (compact) subset of $\mathbb{R}^k$ that parametrizes a class of sources $(\mathbb{P}_\theta)_{\theta\in\Theta}$. f denotes a lossless uniquely decodable coding function.

2.6.1. Who plays first? The n-th order *redundancy* of f with respect to θ is defined as:

$$\mathcal{R}(n,\theta,f) \triangleq \mathbb{E}_\theta\left[\ell(f(X_1^n)) - \log\frac{1}{\mathbb{P}_\theta\{X_1^n\}}\right] .$$

The minimax viewpoint may be described as game between a compressor and an inflator. The compressor first chooses a coding function f and then the inflator chooses a source θ so as to maximize the average redundancy of f. The compressor could be a digital library manager and the inflator a vicious writer that possesses a collection of automatic randomized typewriters representing the possible sources and always chooses the worst typewriter.

DEFINITION 2.6.1. [MINIMAX REDUNDANCY] Define the n-th order minimax redundancy over the class Θ as:

$$\begin{aligned}\bar{\mathcal{R}}(n,\Theta) &\triangleq \inf_f \sup_{\theta\in\Theta}\mathbb{E}_\theta\left[\ell(f(X_1^n)) - \log\frac{1}{\mathbb{P}_\theta\{X_1^n\}}\right]\\ &= \inf_f \sup_{\theta\in\Theta}\mathcal{R}(n,\theta,f) \ .\end{aligned}$$

But inflator and compressor may play a different game, where the inflator plays first by picking at random one randomized typewriter in his collection according to a distribution μ, the compressor may optimize f with repect to μ.

Assume that Θ may be provided with a σ-algebra and a probability measure μ such that $\mathbb{P}_\theta\{x_1^n\}$ is measurable.

REMARK. Tthis is obvious if Θ is finite, and requires a little bit care if Θ is bounded subset of $\mathbb{R}^d$. For example if we consider memoryless sources over the binary alphabet $\{0,1\}$, the set of probabilities over $\{0,1\}$ is parameterized by vectors $\theta \in \mathbb{R}^2$ such that $\theta[1] \geqslant 0$,$\theta[2] \geqslant 0$ and $\theta[1]+\theta[2] = 1$, or even by elements of $[0,1]$. In that case putting a probability on Θ, amounts to put a probability on on $[0,1]$. The most obvious choice (but not the most interesting) consists of providing $[0,1]$ with the uniform probability distribution (Lebesgue measure), but any positive integrable function f on $[0,1]$ such that $\int_{[0,1]} f(x)\,\mathrm{d}x = 1$, defines a probability μ on $[0,1]$ through

$$\mu\{A\} = \int_A f(x)\,\mathrm{d}x = \int_{[0,1]} \mathbf{1}_A(x)\,f(x)\,\mathrm{d}x\,.$$

In the latter situation, f is called the density of the probability distribution μ. This construction does not exhaust the set of probability distributions over $[0,1]$, for example, it does not describe the discrete probability distributions which support set is included in $[0,1]$.

In the sequel μ denotes a probability distribution over Θ. It will be called a prior distribution over Θ, or a prior. Assuming that $\mathbb{P}_\theta\{x_1^n\} = P_\theta^n\{x_1^n\}$ is measurable "with respect to μ" (the latter expression is not technically correct), implies that μ defines a probability distribution say Q^n over $\mathcal{X}^n$:

$$Q^n\{x_1^n\} = \mathbb{E}_\mu\left[P_\theta^n\{x_1^n\}\right]\,.$$

If $\mathcal{X} = \{0,1\}$ andμ is defined by a density on $[0,1]$, if n_1 denotes the number of occurrences of 1 in x_1^n:

$$Q^n\{x_1^n\} = \int_{[0,1]} \left(\prod_{i=1}^n P_\theta\{x_i\}\right) f(\theta)\,\mathrm{d}\theta = \int_{[0,1]} \theta^{n_1}(1-\theta)^{n-n_1}\,f(\theta)\,\mathrm{d}\theta.$$

One can check that $(Q^n)_n$ defines a consistent family of probability distributions over the set of words over alphabet $\mathcal{X}$. Hence $(Q^n)_n$ defines a source over $\mathcal{X}$, but this mixture of memoryless sources is not memoryless...

DEFINITION 2.6.2. [AVERAGE REDUNDANCY WITH RESPECT TO PRIOR] The *average redundancy with respect to* μ is:

$$\underline{\mathcal{R}}(n,\Theta,\mu) \triangleq \inf_f \mathbb{E}_\mu\left[\mathbb{E}_\theta\left[\ell(f(X_1^n)) - \log\frac{1}{\mathbb{P}_\theta\{X_1^n\}}\right]\right]\,.$$

And the n-th order maximin average redundancy rate is:

$$\underline{\mathcal{R}}(n,\Theta) \triangleq \sup_\mu \underline{\mathcal{R}}(n,\Theta,\mu).$$

Clearly for all μ, $\underline{\mathcal{R}}(n,\Theta,\mu) \leq \overline{\mathcal{R}}(n,\Theta)$, and thus:

$$\underline{\mathcal{R}}(n,\Theta) \leq \overline{\mathcal{R}}(n,\Theta)$$

Those definitions allow us to formulate a list of questions:

1 How are $\underline{\mathcal{R}}(n,\Theta)$ and $\overline{\mathcal{R}}(n,\Theta)$ related with the rate of convergence of estimators for Θ ?

2 Are the maximin and minimax average redundancy asymptotically equivalent ?

3 Is it possible to approach the maximin and minimax average redundancy rates using feasible coding strategies?

THEOREM 2.6.3. [OPTIMALITY OF MIXTURES WITH RESPECT TO AVERAGE REDUNDANCY] *The average redundancy with respect to prior μ over the set Θ, $\underline{\mathcal{R}}(n,\Theta,\mu)$ is achieved by the mixture probability Q^n over $\mathcal{X}^n$ defined by*

$$Q^n\{x_1^n\} = \mathbb{E}_\mu\left[P_\theta^n\{x_1^n\}\right]$$

for all $x_1^n \in \mathcal{X}^n$.

The proof is a simple consequence of the non-negativity of relative entropy.

PROOF. Let $Q_\mu$denote the probability on $\mathcal{X}^n$ defined by:

$$Q_\mu\{x_1^n\} = \mathbb{E}_\mu\left[P_\theta^n\{x_1^n\}\right] = \mathbb{E}_\mu\left[\mathbb{E}_\theta\left[\mathbb{1}_{\{x_1^n\}}\right]\right] ,$$

Q_μ is called the mixture generated by μ. Let ϵdenote a non-negative real, then there exists some coding probability Q^n on $\mathcal{X}^n$ such that :

$$\begin{aligned}
\underline{\mathcal{R}}(n,\Theta,\mu) &\geq \mathbb{E}_\mu\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\{X_1^n\}}{Q^n\{X_1^n\}}\right]\right] - \epsilon \\
&= \mathbb{E}_\mu\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\{X_1^n\}}{Q_\mu\{X_1^n\}} + \log\frac{Q_\mu\{X_1^n\}}{Q^n\{X_1^n\}}\right]\right] - \epsilon \\
&= \mathbb{E}_\mu\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\{X_1^n\}}{Q_\mu\{X_1^n\}}\right]\right] + \mathbb{E}_\mu\left[\mathbb{E}_\theta\left[\log\frac{Q_\mu\{X_1^n\}}{Q^n\{X_1^n\}}\right]\right] - \epsilon \\
&= \mathbb{E}_\mu\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\{X_1^n\}}{Q_\mu\{X_1^n\}}\right]\right] + D\left(Q_\mu \,\|\, Q^n\right) - \epsilon \\
&\geq \mathbb{E}_\mu\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\{X_1^n\}}{Q_\mu\{X_1^n\}}\right]\right] - \epsilon \; .
\end{aligned}$$

The last term is the average redundancy of $Q_\mu$with respect to μ. □

Hence we already know that the maximin redundancy of order n has the following form:

$$\sup_\mu \mathbb{E}_\mu\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\{X_1^n\}}{Q_\mu\{X_1^n\}}\right]\right] .$$

It can be checked that

$$\mu \mapsto \mathbb{E}_\mu\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\{X_1^n\}}{Q_\mu\{X_1^n\}}\right]\right]$$

is concave with respect to μ(as should be any infimum over linear functionals).

Moreover for any μ':

$$\begin{aligned}\mathbb{E}_{\mu'}\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\left\{X_1^n\right\}}{Q_{\mu'}\left\{X_1^n\right\}}\right]\right]-\mathbb{E}_{\mu}\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\left\{X_1^n\right\}}{Q_{\mu}\left\{X_1^n\right\}}\right]\right]\\ =-D(Q_{\mu'}\|Q_\mu)+\mathbb{E}_{\mu'}\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\left\{X_1^n\right\}}{Q_{\mu}\left\{X_1^n\right\}}\right]\right]-\mathbb{E}_{\mu}\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\left\{X_1^n\right\}}{Q_{\mu}\left\{X_1^n\right\}}\right]\right]\end{aligned}$$

which entails that for any probability μ' over Θ:

$$0\leqslant\mathbb{E}_{\mu'}\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\left\{X_1^n\right\}}{Q_{\mu}\left\{X_1^n\right\}}\right]\right]-\mathbb{E}_{\mu}\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\left\{X_1^n\right\}}{Q_{\mu}\left\{X_1^n\right\}}\right]\right].$$

Little computations reveal that for anyθ:

$$D(P_\theta^n\|Q_\mu)\leqslant\mathbb{E}_{\mu}\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\left\{X_1^n\right\}}{Q_{\mu}\left\{X_1^n\right\}}\right]\right].$$

Thus

$$\bar{\mathcal{R}}(n,\Theta)\leqslant\sup_{\theta\in\Theta}D(P_\theta^n\|Q_\mu)\leqslant\underline{\mathcal{R}}(n,\Theta),$$

which entails

$$\bar{\mathcal{R}}(n,\Theta)=\underline{\mathcal{R}}(n,\Theta)$$

2.6.2. Another look at maximin versus minimax. Note that choosing f and thus $\ell(f(\cdot))$ amounts to define a positive vector in $\mathbb{R}^{|\mathcal{X}|^n}$ and that choosing μ amounts to choose an element in a convex compact space (probabilities over probabilities over a finite space equipped with weak convergence topologies). Now recall Sion Minimax Theorem:

THEOREM 2.6.4. *Let g denote a function from topological vector spaces $E\times F$ to $\mathbb{R}$ such that g is convex and continuous with respect to its first argument and concave and continuous with respect to its second argument, assume $A\subseteq E$ is compact, then*

$$\inf_{x\in A}\sup_{y\in E}g(x,y)=\sup_{y\in E}\inf_{x\in A}g(x,y).$$

Note that

$$(\mu,Q^n)\mapsto\mathbb{E}_{\mu}\left[\mathbb{E}_\theta\left[\log\frac{P_\theta\left\{X_1^n\right\}}{Q^n\left\{X_1^n\right\}}\right]\right]$$

is concave with respect to μ and convex with respect to Q^n, Q^ntakes its values in a compact set. Hence we may now conclude:

THEOREM 2.6.5. *The maximin and the minimax redundancy are equal*

$$\underline{\mathcal{R}}(n,\Theta)=\overline{\mathcal{R}}(n,\Theta).$$

From this Theorem and the optimality of mixtures with respect to the maximin redundancy, we get that if μ_n solves the nth maximin redundancy problem:

$$\mathbb{E}_\theta\left[\log\frac{P_\theta\left\{X_1^n\right\}}{Q_{\mu_n}\left\{X_1^n\right\}}\right]=\underline{\mathcal{R}}(n,\Theta)=\overline{\mathcal{R}}(n,\Theta)\,,$$

the optimal mixture Q_{μ_n} equalizes the redundancy.

This general reasoning does not tell us what $\overline{\mathcal{R}}(n,\Theta)$ is.

2.7. Regrets

Finally another notion has recently acquired extreme importance. It provides a bridge between the stochastic modeling attitude of Information Theory and the Individual Sequences perspective.

Definition 2.7.1. [pointwise regret] The maximum pointwise regret of a code (f,ϕ) with respect to a model Θ is:

$$\max_{x_1^n}\max_{\theta\in\Theta}\left[\ell(f(x_1^n))-\log\frac{1}{P\left\{x_1^n\right\}}\right]\ .$$

A coding probability achieves minimax pointwise regret over words of length n its maximum pointwise regret is minimal. The following exercise shows that coding probabilities achieving minimax pointwise regret have a simple definition. Note that the minimax pointwise regret is at least as large as the minimax redundancy $\bar{\mathcal{R}}(n,\Theta)$. Any upper bound on the pointwise regret provides an upper-bound on the minimax redundancy.

Definition 2.7.2. [Normalized Maximum Likelihood] Let Q^n be defined as

$$Q^n\left\{x_1^n\right\}=\frac{\max_{P^n}P^n\{x_1^n\}}{\sum_{x_1^n\in\mathcal{X}^n}\max_{P^n}P^n\{x_1^n\}}$$

where maximization is performed over all memoryless sources over alphabet $\mathcal{X}$.The distribution Q^n is called the nth Normalized Maximum Likelihood (NML) coding probability.

Theorem 2.7.3. *The NML coding probability achieves the minimax pointwise regret, moreover Q^n equalizes the regret over the possible words.*

Proof. Let Θdenote a model such that

$$\sum_{x_1^n\in\mathcal{X}^n}\max_{P^n\in\Theta}P^n\{x_1^n\}$$

is finite, then for any x_1^n, the pointwise regret of Q^n on x_1^n equals

$$\log\left(\sum_{x_1^n\in\mathcal{X}^n}\max_{P^n} P^n\{x_1^n\}\right) .$$

On the other hand, let R^ndenote another probability on $\mathcal{X}^n$, for some input $x_1^n R^n\left\{x_1^n\right\} \leq Q^n\left\{x_1^n\right\}$, hence the regret of R^non x_1^n is larger than

$$\log\left(\sum_{x_1^n\in\mathcal{X}^n}\max_{P^n} P^n\{x_1^n\}\right) .$$

□

EXERCISE 2.8. Provide an expression for the minimax pointwise regret for memoryless sources on the binary alphabet.

When dealing with memoryless sources over alphabet $\mathcal{X}=\{0,1\}$, we may evaluate the minimax pointwise regret. For any $x_1^n\in\{0,1\}^n$, such that there are n_1 occurrences of 1 in x_1^n, the maximum likelihood is equal to

$$\left(\frac{n_1}{n}\right)^{n_1}\left(\frac{n-n_1}{n}\right)^{n-n_1} .$$

There are exactly $\binom{n}{n_1}$ words with exactly n_1 occurrences of 1 over n. The minimax pointwise regret is thus equal to:

$$\log\left(\sum_{n_1=0}^{n}\binom{n}{n_1}\left(\frac{n_1}{n}\right)^{n_1}\left(\frac{n-n_1}{n}\right)^{n-n_1}\right) .$$

In order to get a user-friendly upper-bound on this expression, it is useful to use the Robbins-Stirling approximations to the factorial. This leads to:

$$\binom{n}{n_1}\left(\frac{n_1}{n}\right)^{n_1}\left(\frac{n-n_1}{n}\right)^{n-n_1} \leqslant \frac{1}{\sqrt{2\pi}}\sqrt{\frac{n}{n_1(n-n_1)}}\,\mathrm{e}^{1/(12n+1)} .$$

Now

$$\begin{aligned}\sum_{n_1=0}^{n}\binom{n}{n_1}\left(\frac{n_1}{n}\right)^{n_1}\left(\frac{n-n_1}{n}\right)^{n-n_1} &\leqslant \frac{1}{\sqrt{2\pi}}\mathrm{e}^{1/(12n+1)}\sum_{n_1=0}^{n}\sqrt{\frac{n}{n_1(n-n_1)}}\\ &\leqslant \frac{\mathrm{e}^{1/(12n+1)}}{\sqrt{2\pi}}\sqrt{n}\sum_{n_1=0}^{n}\frac{1}{n}\sqrt{\frac{1}{(n_1/n)(1-(n_1/n))}}\\ &\leqslant \frac{\mathrm{e}^{1/(12n+1)}}{\sqrt{2\pi}}\sqrt{n}\sum_{n_1=0}^{n}\frac{1}{n}\sqrt{\frac{1}{(n_1/n)(1-(n_1/n))}}.\end{aligned}$$

Hence

$$\log\left(\sum_{n_1=0}^{n}\binom{n}{n_1}\left(\frac{n_1}{n}\right)^{n_1}\left(\frac{n-n_1}{n}\right)^{n-n_1}\right)-\frac{1}{2}\log n$$
$$\leqslant \log\left(\frac{\mathrm{e}^{1/(12n+1)}}{\sqrt{2\pi}}\right)+\log\left(\sum_{n_1=0}^{n}\frac{1}{n}\sqrt{\frac{1}{(n_1/n)(1-(n_1/n))}}\right).$$

The sum in the right-hand-side is an approximation to the (Riemann) integral

$$\int_0^1 \frac{1}{\sqrt{x(1-x)}}\,\mathrm{d}x=\pi .$$

Hence the minimax pointwise regret (and the minimax redundancy) satisfy the following relations

$$\log\left(\sum_{n_1=0}^{n}\binom{n}{n_1}\left(\frac{n_1}{n}\right)^{n_1}\left(\frac{n-n_1}{n}\right)^{n-n_1}\right)\leqslant \frac{1}{2}\log\left(\frac{n\pi}{2}\right)+o(1)\,.$$

This provides with an improvement over the naive bound derived from the plug-in technique (an improvement by a factor of 2). A close inspection of the computations shows that the upper-bound is tight for pointwise regret. It can be checked that this upper-bound is also tight for minimax redundancy. Moreover there is nothing special with the two-symbols alphabet. The reasoning could have been carried out for larger alphabets, it would have led to upper-bounds of the form

$$\frac{(d-1)}{2}\log(n)+\log\frac{\Gamma(1/2)^d}{\Gamma(d/2)}+o(1)\,.$$

The first thing to notice is the $(d-1)/2\log(n)$ term. It should be compared with the $d\log n$ term derived for the plug-in code. This discrepancy can be interpreted in the following way: when building the plug-in codes, we coded the rational numbers defining the maximum likelihood estimator with high accuracy, we could have set-up a sieve for numbers $0, 1/n, 2/n, \ldots, 1$ with mesh $1/\sqrt{n}$ and approximated the coefficients of the maximum likelihood estimation by the closest element in the sieve. Coding a point in the sieve requires roughly $\frac{1}{2}\log n$ bits, and we do not loose to much by replacing $\widehat{\theta}$ by its approximation.

The maximum pointwise regret of NML coding probabilities provides benchmarks. But NML probabilities are mostly of theoretical interest: they do not define a consistent family of probabilty distributions. This is one of the reasons why mixture coders have attracted such much attention.

2.9. Mixture coding

Let us now examine good universal coding probabilities for interesting models. The first model Θ to be considered is the class of memoryless sources over the finite alphabet $\mathcal{X}$. In the language of the previous section this is a $|\mathcal{X}|-1$-dimensional parametric model. We will be satisfied if we can find a coding probability $\mathbb{Q}$ such that $\overline{\mathcal{R}}(n,\theta,\mathbb{Q})/\log_2 n$ tends toward $(|\mathcal{X}|-1)/2$ as n tends toward infinity.

In order to make this mixture coding something useful, we would also like $Q^{n+1}\{x_1^{n+1}\}$ to be easily computable from $Q^n\{x_1^n\}$. We have two goals: a computational and an information-theoretical one.

The following prior distribution on probability distributions over the finite alphabet $\mathcal{X}=\{1,\dots k\}$ plays a distinguished role in universal data compression.

The d-dimensional simplex is the subset of vectors $\mathbf{u}$ from $\mathbb{R}^d$ defined by $u[i]\geqslant 0$ for all i, $1\leqslant i\leqslant d$, and $\sum_{i=1}^d u_i=1$. Each element of the simplex defines a probability on $\mathcal{X}$ when $|\mathcal{X}|=d$.

Recall that the Γ function is defined by:

$$\Gamma(x+1)\triangleq\int_0^\infty e^{-t}t^x dt,$$

and that $\Gamma(x+1)=x\Gamma(x)$. The Gamma function interpolates the factorial numbers. It can also be approximated thanks to the Robbins-Stirling bounds. For all $x>0$:

$$x^{x-\frac{1}{2}}e^{-x}\sqrt{2\pi}\leq\Gamma(x)\leq x^{x-\frac{1}{2}}e^{-x}\sqrt{2\pi}e^{\frac{1}{12x}}\quad. \tag{2.9.1}$$

Recall also that $\Gamma(1/2)=\sqrt{\pi}$.

The beta function $B(\alpha,\beta)$ is defined by

$$B(\alpha,\beta)=\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}.$$

The following relationship plays a fundamental role in subsequent analysis:

$$B(\alpha,\beta)=\int_0^1\theta^{\alpha-1}(1-\theta)^{\beta-1}\,\mathrm{d}\theta\,.$$

LEMMA 2.9.1. *If X and Y are independent random variables distributed according to Gamma distributions with parameters α and β, then $X+Y$ and $X/(X+Y)$ are independent random variables distributed according to a Gamma distribution with parameter $\alpha+\beta$ and to a Beta distribution with parameters α,β.*

Proof. Under the notations of the Lemma, the jont density of X and Y over $\mathbb{R}_+ \times \mathbb{R}_+$ is

$$\frac{x^{\alpha-1}y^{\beta-1}\mathrm{e}^{-x}\mathrm{e}^{-y}}{\Gamma(\alpha)\Gamma(\beta)} .$$

The inverse of the one-to-one transformation $(x,y) \mapsto (x+y, x/(x+y))$ is $(x',y') \mapsto (x'y', x'-x'y')$. Its Jacobian has determinant :$-x'$. Hence the joint density of $X+Y$, and $X/(X+Y)$ at (u,v) is

$$\frac{(uv)^{\alpha-1}(u-uv)^{\beta-1}}{\Gamma(\alpha)\Gamma(\beta)}\, u\, \mathrm{e}^{-uv}\mathrm{e}^{-u+uv} = \frac{u^{\alpha+\beta-1}}{\Gamma(\alpha+\beta)}\,\mathrm{e}^{-u}\,\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\,v^{\alpha-1}(1-v)^{\beta-1}$$

The fact that the joint density can be factorized proves the independence of $X+Y$ and $X/(X+Y)$. Furthermore if we integrate out with respect to u, we realize that $Y/(X+Y)$ has a Beta(α,β) density. □

Definition 2.9.2. [Dirichlet distribution over the d-dimensional simplex] Let $\boldsymbol{\alpha} \in \mathbb{R}_+^d$, the Dirichlet distribution over the d-dimensional simplex with parameter $\boldsymbol{\alpha}$ has density

$$\frac{\Gamma(\sum_{j=1}^d \alpha_j)}{\prod_{j=1}^d \Gamma(\alpha_j)} \prod_{j=1}^r u_j^{\alpha_j-1} \text{for every } \mathbf{u} \text{ in the simplex.}$$

For the 2-dimensional case (that is for the binary alphabet), the density equals

$$\frac{\Gamma(\alpha_1+\alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)}\,\theta^{\alpha_1-1}(1-\theta)^{\alpha_2-1} = \frac{\theta^{\alpha_1-1}(1-\theta)^{\alpha_2-1}}{B(\alpha_1,\alpha_2)} .$$

As a special case we can recover the uniform prior (or Laplace prior) by taking $\alpha_1 = \alpha_2 = 1$. But calculation would reveal that it does not perform essentially better than the naive plug-in code. Indeed Dirichlet priors provide an handy way to solve our computational problem (whatever the choice of $\boldsymbol{\alpha}$), this is a consequence of classical results in Bayesian statistics. But, our information-theoretical problem will be solved by choosing $\alpha_i = 1/2$ for all $i \leqslant d$.

Definition 2.9.3. [Dirichlet prior] The k-ary 1/2-Dirichlet prior on the space of probability distributions on $\mathcal{X} = \{1, \ldots k\}$ has density:

$$\nu(\theta_1^{k-1}) = \mathbb{1}\{\sum_{i<k} \theta_i \le 1\} \frac{\Gamma(k/2)}{\Gamma(1/2)^k \prod_{i=1}^k \theta_i^{1/2}} ,$$

with respect to the Lebesgue measure on $[0,1]^{k-1}$, with the convention that $\theta_k = 1 - \sum_{i<k} \theta_i$.

When the alphabet is binary, each probability on $\mathcal{X} = \{0,1\}$ is completely defined by a single number $\theta = \theta_0$ and the Dirichlet prior density (with respect to Lebesgue measure on $[0,1]$ turns out to equal:

$$\frac{1}{\Gamma(1/2)^2\sqrt{\theta(1-\theta)}} = \frac{1}{\pi\sqrt{\theta(1-\theta)}} \ .$$

Definition 2.9.4. [Krichevsky-Trofimov mixture] The Krichevsky-Trofimov mixture distribution over strings of length n is defined by

$$\mathsf{kt}(\mathbf{x}_1^n) \triangleq \int d\theta \left[\nu(\theta)\prod_{i=1}^k \theta_i^{n_i} \mathbb{1}\{\sum_{i<k} \theta_i \leq 1\}\right],$$

where n_i is the number of occurrences of the ith alphabet symbol in the string x_1^n, and integration takes place over $\sum_{i<k}\theta_i \leq 1$ while $\theta_i \geq 0$ for $i < k$ and the convention $\theta_k = 1 - \sum_{i<k}\theta_i$.

The fact that Dirichlet mixtures solve our computational problem is summarized by the following proposition.

Proposition 2.9.5. *The KT-probability of the sequence $\mathbf{x}_1^n$ over alphabet $\mathcal{X}$ of size k satisfies the following formula:*

$$\mathit{kt}\{\mathbf{x}_1^n\} = \frac{\Gamma\left(\frac{k}{2}\right)\prod_{i\in\mathcal{X}}\Gamma\left(n_i+\frac{1}{2}\right)}{\Gamma\left(\frac{1}{2}\right)^k\Gamma\left(n+\frac{|\mathcal{X}|}{2}\right)}. \tag{2.9.2}$$

As $\Gamma(x+1) = x\Gamma(x)$, the KT-conditional probability of observing $X_{n+1} = a$ given $\mathbf{x}_1^n$ is

$$\mathsf{kt}\{X_{n+1} = a \mid X_1^n = \mathbf{x}_1^n\} = \frac{n_a + 1 + \frac{1}{2}}{n + 1 + \frac{|\mathcal{X}|}{2}} \tag{2.9.3}$$

where n_a denotes the number of occurrences of symbol a in $\mathbf{x}_1^n$.

Proof. We sketch thet proof for the binary alphabet. Under the Dirichlet mixtures, thanks to the fundamental connection between the Beta function and the Gamma function, the probability of a word with $n_1$occurrences of 1 and $n-n_1$ occurrences of 0:

$$\int_0^1 \frac{\Gamma(\alpha_1+\alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)}\theta^{n_1+\alpha_1-1}(1-\theta)^{n-n_1+\alpha_2-1}\,\mathrm{d}\theta = \frac{B(n_1+\alpha_1, n-n_1+\alpha_2)}{B(\alpha_1,\alpha_2)} \ .$$

But

$$B(n_1+\alpha_1, n-n_1+\alpha_2) \ = \ \frac{\prod_{i=1}^n (n-i+\alpha_1)}{\prod_{i=1}^{n_1}(n_1-i+\alpha)\prod_{i=1}^{n-n_1}(n-n_1-i+\alpha)} B(\alpha_1,\alpha_2) \, .$$

□

THEOREM 2.9.6. *The worst-case pointwise regret of the* KT*-mixture with respect to the class* Θ *of memoryless coding distributions is upper-bounded by:*

$$\widehat{\mathcal{R}}\big(n, \mathbf{x}, \Theta\big) \leq \log \frac{\Gamma(n+\frac{k}{2})\Gamma(\frac{1}{2})}{\Gamma(n+\frac{1}{2})\Gamma(\frac{k}{2})} \leq \frac{k-1}{2}\log n - \log\frac{\Gamma(k/2)}{\Gamma(1/2)} + o(1/n) \ .$$

PROOF. The maximum-likelihood is achieved for a tuple $\hat{\theta}_1, \ldots \hat{\theta}_k$ that matches the empirical distribution of the sequence $\mathbf{x}_1^n$, let n_i denote the number of occurrences of symbol $i \in \mathcal{X}$ in $\mathbf{x}_1^n$. The maximum likelihood among memoryless sources is thus

$$\prod_{i=1}^{k}\left(\frac{n_i}{n}\right)^{n_i} .$$

The regret of the KT-mixture is thus

$$\log\frac{\Gamma\left(\frac{1}{2}\right)^k\Gamma\left(n+\frac{k}{2}\right)\prod_{i\leq k}\left(\frac{n_i}{n}\right)^{n_i}}{\Gamma\left(\frac{k}{2}\right)\prod_{i\leq k}\Gamma\left(n_i+\frac{1}{2}\right)}$$

It is enough to check that:

$$\prod_{i=1}^{k}\left(\frac{n_i}{n}\right)^{n_i} \leq \prod_{i=1}^{k}\frac{\Gamma\left(n_i+\frac{1}{2}\right)}{\Gamma(\frac{1}{2})}\frac{1}{\Gamma\left(n+\frac{1}{2}\right)} \ .$$

Note that right-hand-side can be rewritten as

$$\begin{aligned}
&\frac{\prod_{i=1}^{k}\left[(n_i-\frac{1}{2})(n_i-\frac{3}{2})\ldots\frac{1}{2}\right]}{(n-\frac{1}{2})(n-\frac{3}{2})\ldots\frac{1}{2}} \\
= \quad & \frac{\prod_{i=1}^{k}\left[n_i(n_i-\frac{1}{2})(n_i-1)(n_i-\frac{3}{2})\ldots 1\frac{1}{2}\right]}{n_i!}\frac{n!}{n(n-\frac{1}{2})(n-1)(n-\frac{3}{2})\ldots 1\frac{1}{2}} \\
= \quad & \prod_{i=1}^{k}\frac{(2n_i)!}{2^{2n_i}n_i!}\frac{2^{2n}n!}{(2n)!} \\
= \quad & \prod_{i=1}^{k}\frac{(2n_i)!}{n_i!}\frac{n!}{(2n)!} \\
= \quad & \frac{\prod_{i=1}^{k}\left[(2n_i)(2n_i-1)\ldots(n_i+1)\right]}{(2n)(2n-1)\ldots(n+1)} \ .
\end{aligned}$$

There are n factors at the denominator and n terms at the numerator. There are also n factors in $\prod_{i=1}^{k}\left(n/n_i\right)^{n_i}$. The desired result will be obtained by matching factors on both sides in an appropriate way. Note that

$$\frac{n_i}{n} \leq \frac{n_i+j}{n+\ell} \tag{2.9.4}$$

is satisfied for all j such that $n_i\ell/n \leq j \leq n_i$. Note that there are at least $n_i(1-\ell/n)$ of them. Hence for a given choice of ℓ, the number of pairs (i,j) such that Inequality (2.9.4) holds is larger than

$$\sum_{i=1}^{k} n_i \left(1 - \frac{\ell}{n}\right) = n - \ell.$$

This motivates the following greedy allocation method. For all ℓ starting from $\ell = n$ downto $\ell = 1$, choose any (i,j) such that inequality (2.9.4) is satisfied call that pair (i_ℓ, j_ℓ). Note that at each step there is at least on such pair which is available.

By construction, we have:

$$\prod_{\ell=1}^{n} \left(\frac{n_{i_\ell}}{n}\right) = \prod_{\ell=1}^{n} \frac{n_{i_\ell} + j_\ell}{n+\ell} \quad .$$

This terminates the proof of the first part of the Theorem.

The proof of the remaining part of Theorem 2.9.6 boils down to applying classical bounds on the Γ function □

Remark 2.9.7. The pointwise regret is attained on samples formed by repeating a single symbol.