Universal Source Coding

I-Hsiang Wang

Department of Electrical Engineering
National Taiwan University

ihwang@ntu.edu.tw

December 20, 2024

Recall the fixed-to-fixed lossless source coding result:

> *For a DMS $S \sim \mathsf{P}_S$, the minimum compression ratio is the entropy of the source $\mathrm{H}(S)$.*

Recall the fixed-to-variable lossless source coding result:

> *For a DMS $S \sim \mathsf{P}_S$, the minimum expected codeword length for perfect reconstruction is roughly the entropy $\mathrm{H}(S)$ of the source.*

The coding schemes (typicality-based, Shannon-type, Huffman, etc.) introduced so far all depend on the distribution of the source, P_S.

What if P_S is unknown?

Codes that work for a family of sources

If we know that the source distribution belongs to a family of distributions

$$\mathcal{P}_\Theta := \{\mathsf{P}_\theta \,|\, \theta \in \Theta\} \subset \mathcal{P}(\mathcal{S}),$$

we may hope to propose a coding scheme that can reconstruct the source losslessly for all $\mathsf{P} \in \mathcal{P}_\Theta$.

Is this difficult?

It depends on how complicated $\mathcal{P}_\Theta$ is. The simplest case is $|\Theta| < \infty$. In this case, one can choose the range of the decoding function (a high probability set) as $\mathcal{A} = \bigcup_{\theta\in\Theta} \mathcal{A}_\delta^{(n)}(\mathsf{P}_\theta)$, the (finite) union of (weakly) typical sets defined in Lecture 1. The cardinality of this set can be easily controlled:

$$|\mathcal{A}| \le \sum_{\theta\in\Theta} |\mathcal{A}_\delta^{(n)}(\mathsf{P}_\theta)| \le |\Theta| 2^{n(\max_{\theta\in\Theta} \mathrm{H}(\mathsf{P}_\theta)+\delta)} \underset{\text{for } n \text{ large}}{\le} 2^{n(\max_{\theta\in\Theta} \mathrm{H}(\mathsf{P}_\theta)+\delta')}$$

Hence, we can guarantee all compression rates $\mathrm{R} > \max_{\theta\in\Theta} \mathrm{H}(\mathsf{P}_\theta)$.

For more complicated $\mathcal{P}_\Theta$, can we still have a universal coding scheme that can guarantee almost lossless reconstruction (vanishing error probability) for all $\mathsf{P} \in \mathcal{P}_\Theta$, for any code rate $\mathrm{R} > \sup_{\theta\in\Theta} \mathrm{H}(\mathsf{P}_\theta)$?

The answer is yes (as shown in later slides).

Instead of using typical sets, we use **the method of types** that provides a sharper analysis.

Type

Definition 1 (Type/Empirical Distribution)

For a sequence $x^n \in \mathcal{X}^n$ where $\mathcal{X} = \{a_1, a_2, \ldots, a_d\}$, the number of occurrence of a symbol $a \in \mathcal{X}$ in the sequence is defined as

$$n(a|x^n) := \sum_{i=1}^{n} \mathbb{1}\{x_i = a\}, \quad a \in \mathcal{X}.$$

The *type* (empirical distribution/histogram) of the sequence is a d-dimensional probability vector defined as the collection of all d frequencies of occurrence:

$$\hat{\boldsymbol{p}}(x^n) := \frac{1}{n}(n(a_1|x^n), n(a_2|x^n), ..., n(a_d|x^n)).$$

It belongs to the probability simplex $\mathcal{P}_d$. Hence, when the context is about probability laws, we interchangeably use $\hat{\mathrm{P}}_{x^n}$ to represent its type.

Note: The type of a sequence is a deterministic function of the sequence. Nothing random here.

Note that the possible types of sequences of length $n \in \mathbb{N}$ just form a strict subset of the probability simplex, and the number of them is polynomial in n.

Definition 2 (Collection of Types)

For a given length $n \in \mathbb{N}$, we use $\hat{\mathcal{P}}_{n,d}$ to denote the collection of all possible "n-types" of length-n sequences:

$$\hat{\mathcal{P}}_{n,d} := \left\{ \left(\tfrac{i_1}{n}, \tfrac{i_2}{n}, ..., \tfrac{i_d}{n}\right) \middle| \forall\, i_1, ..., i_d \in \mathbb{N} \cup \{0\},\ i_1 + \cdots + i_d = n \right\}.$$

Exercise 1

Show that

$$\left|\hat{\mathcal{P}}_{n,d}\right| = \binom{n+d-1}{d-1} \leq (n+1)^d.$$

Type class

For a type $\hat{\mathsf{P}} \in \hat{\mathcal{P}}_{n,d}$, there are many length-$n$ sequences with the same type $\hat{\mathsf{P}}$. The collection of such sequences is called a *type class*.

Definition 3 (Type Class)

For a type $\hat{\mathsf{P}} \in \hat{\mathcal{P}}_{n,d}$, the type class of $\hat{\mathsf{P}}$, $\mathcal{T}_n(\hat{\mathsf{P}})$, is the collection of length-n sequences with type $\hat{\mathsf{P}}$:

$$\mathcal{T}_n(\hat{\mathsf{P}}) := \left\{ x^n \in \mathcal{X}^n \,\middle|\, \hat{\mathsf{P}}_{x^n} = \hat{\mathsf{P}} \right\}.$$

It is important to count the number of sequences with the same type. The size of the type class of $\mathcal{T}_n(\hat{\boldsymbol{p}})$ can be characterized explicitly as the following combinatorial expression which gives little insight:

$$|\mathcal{T}_n(\hat{\boldsymbol{p}})| = \frac{n!}{(n\hat{p}_1)!(n\hat{p}_2)!\cdots(n\hat{p}_d)!}, \text{ where } \hat{\boldsymbol{p}} = (\hat{p}_1, ..., \hat{p}_d).$$

It turns out that $|\mathcal{T}_n(\hat{\mathsf{P}})|$ is roughly $2^{n\mathrm{H}(\hat{\mathsf{P}})}$, to within a poly-$n$ factor.

Probability of an i.i.d. sequence

Type is a natural way to classify sequences when we are concerned about the probability of an i.i.d. generated sequence.

Proposition 1 (Probability of an i.i.d. sequence depends only on its type)

For an i.i.d. generated sequence X^n with $X_i \overset{\text{i.i.d.}}{\sim} \mathsf{P},\ i = 1, 2, ..., n$,

$$\mathsf{P}^{\otimes n}\{X^n = x^n\} = 2^{-n\left(\mathrm{H}\left(\hat{\mathsf{P}}_{x^n}\right)+\mathrm{D}\left(\hat{\mathsf{P}}_{x^n}\middle\|\mathsf{P}\right)\right)}.$$

pf: For notational convenience, let us denote $\mathsf{P}\{X = a\}$ as $\mathsf{P}(a)$. We prove this proposition by direct calculation:

$$\begin{aligned}
\mathsf{P}^{\otimes n}(x^n) = \prod_{i=1}^n \mathsf{P}(x_i) = \prod_{l=1}^d \mathsf{P}(a_l)^{n(a_l|x^n)} &= 2^{-n\sum\limits_{l=1}^d \frac{n(a_l|x^n)}{n}\log\frac{1}{\mathsf{P}(a_l)}} \\
&= 2^{-n\sum\limits_{l=1}^d \left\{\hat{\mathsf{P}}_{x^n}(a_l)\log\frac{1}{\hat{\mathsf{P}}_{x^n}(a_l)}+\hat{\mathsf{P}}_{x^n}(a_l)\log\frac{\hat{\mathsf{P}}_{x^n}(a_l)}{\mathsf{P}(a_l)}\right\}} \\
&= 2^{-n\left[\mathrm{H}\left(\hat{\mathsf{P}}_{x^n}\right)+\mathrm{D}\left(\hat{\mathsf{P}}_{x^n}\middle\|\mathsf{P}\right)\right]}.
\end{aligned}$$

□

Size of a type class

The size of a type class can be approximately characterized by its entropy, which is quite similar to the set of typical sequences.

Proposition 2 (Bounds on the Size of a Type Class)

For a type $\hat{\mathsf{P}} \in \hat{\mathcal{P}}_{n,d}$ *,* $\frac{1}{|\hat{\mathcal{P}}_{n,d}|} 2^{n\mathrm{H}(\hat{\mathsf{P}})} \le |\mathcal{T}_n(\hat{\mathsf{P}})| \le 2^{n\mathrm{H}(\hat{\mathsf{P}})}$.

pf: Our proof is based on a probabilistic argument. Construct an i.i.d. sequence $X^n \sim \hat{\mathsf{P}}^{\otimes n}$. By Proposition 1, we have

$$\hat{\mathsf{P}}^{\otimes n}(x^n) = 2^{-n\left[\mathrm{H}(\hat{\mathsf{P}}_{x^n}) + \mathrm{D}(\hat{\mathsf{P}}_{x^n} \| \hat{\mathsf{P}})\right]}.$$

If $x^n \in \mathcal{T}_n(\hat{\mathsf{P}})$, then $\hat{\mathsf{P}}_{x^n} = \hat{\mathsf{P}}$ and $\hat{\mathsf{P}}^{\otimes n}(x^n) = 2^{-n\mathrm{H}(\hat{\mathsf{P}})}$. Hence,

$$1 \ge \sum_{x^n \in \mathcal{T}_n(\hat{\mathsf{P}})} \hat{\mathsf{P}}^{\otimes n}(x^n) = |\mathcal{T}_n(\hat{\mathsf{P}})| 2^{-n\mathrm{H}(\hat{\mathsf{P}})} \implies |\mathcal{T}_n(\hat{\mathsf{P}})| \le 2^{n\mathrm{H}(\hat{\mathsf{P}})},$$

and the upper bound is proved.

To prove the lower bound, observe that for any type $\hat{\mathsf{Q}} \in \hat{\mathcal{P}}_{n,d}$, the probability that X^n (following $\hat{\mathsf{P}}^{\otimes n}$) falls into this type class $\mathcal{T}_n(\hat{\mathsf{Q}})$ is maximized when $\hat{\mathsf{Q}} = \hat{\mathsf{P}}$:

$$\hat{\mathsf{P}}^{\otimes n}(\mathcal{T}_n(\hat{\mathsf{P}})) \geq \hat{\mathsf{P}}^{\otimes n}(\mathcal{T}_n(\hat{\mathsf{Q}})), \ \forall\, \hat{\mathsf{Q}} \in \hat{\mathcal{P}}_{n,d}. \tag{1}$$

The proof of this is left as exercise (Imre Csiszár: Simple algebra.)

By (1) we have

$$\begin{aligned} 1 &= \textstyle\sum_{\hat{\mathsf{Q}} \in \hat{\mathcal{P}}_{n,d}} \hat{\mathsf{P}}^{\otimes n}(\mathcal{T}_n(\hat{\mathsf{Q}})) \leq |\hat{\mathcal{P}}_{n,d}|\, \hat{\mathsf{P}}^{\otimes n}(\mathcal{T}_n(\hat{\mathsf{P}})) \\ &= |\hat{\mathcal{P}}_{n,d}|\, |\mathcal{T}_n(\hat{\mathsf{P}})|\, 2^{-n\mathrm{H}(\hat{\mathsf{P}})} \implies |\mathcal{T}_n(\hat{\mathsf{P}})| \geq \tfrac{1}{|\hat{\mathcal{P}}_{n,d}|} 2^{n\mathrm{H}(\hat{\mathsf{P}})}. \end{aligned}$$

□

Remark: Since $|\hat{\mathcal{P}}_{n,d}|$ polynomial in n, the size of the typical set $|\mathcal{T}_n(\hat{\mathsf{P}})|$ has the same exponential order as $2^{n\mathrm{H}(\hat{\mathsf{P}})}$, that is, $|\mathcal{T}_n(\hat{\mathsf{P}})| \doteq 2^{n\mathrm{H}(\hat{\mathsf{P}})}$.

Exercise 2

Prove (1) by direct calculation and note that $m!/n! \geq n^{m-n}$ for $m, n \in \mathbb{N}$.

Probability of a type class

Proposition 1 tells us the probability of a sequence of a given type under i.i.d. distribution. Proposition 2 gives exponentially tight bounds for the size of a type class. As a result, we have exponentially tight bounds for the probability of a type class under i.i.d. distribution as follows.

Proposition 3 (Probability of a type class and information divergence)

For an i.i.d. generated sequence X^n *with* $X_i \overset{\text{i.i.d.}}{\sim} \mathsf{P},\ i = 1, 2, ..., n$,

$$\frac{1}{|\hat{\mathcal{P}}_{n,d}|} 2^{-n\mathrm{D}(\hat{\mathsf{P}}\|\mathsf{P})} \leq \mathsf{P}^{\otimes n}\left\{X^n \in \mathcal{T}_n(\hat{\mathsf{P}})\right\} \leq 2^{-n\mathrm{D}(\hat{\mathsf{P}}\|\mathsf{P})}.$$

In other words, $\mathsf{P}^{\otimes n}(\mathcal{T}_n(\hat{\mathsf{P}})) \doteq 2^{-n\mathrm{D}(\hat{\mathsf{P}}\|\mathsf{P})}$.

The proof is straightforward and hence omitted.

Codes that work for all sources w/ entropy $<$ the rate

In fixed-to-fixed source coding, compression rate R is fixed *a priori*.

For a DMS, when P_S is unknown, if the source entropy $\mathrm{H}(S)$ turns out to be *greater than* the chosen R, the strong converse of Shannon's lossless source coding theorem tells us the performance is guaranteed to be terrible.

So the best thing we can hope for is that, if we are so lucky that $\mathrm{H}(S) < \mathrm{R}$, then our scheme (in fact, a sequence of schemes) can guarantee vanishing error probability for all such sources.

Indeed, we can do that, as shown in the following theorem.

Theorem 1

For any source alphabet $\mathcal{S}$, $|\mathcal{S}| < \infty$ and $\mathrm{R} > 0$, there exists a sequence of $(n, \lfloor n\mathrm{R} \rfloor)$ lossless source codes such that

$$\lim_{n\to\infty} \mathsf{P}_{\mathsf{e}}^{(n)} = 0 \quad \forall\, S \text{ with } \mathrm{H}(S) < \mathrm{R}\,.$$

pf: Recall: fixed-to-fixed source coding
$\approx$ finding a high-probability $\mathcal{A} \subset \mathcal{S}^n$ with small cardinality.

For a sequence generated from a DMS, its probability is determined by its type. Hence, it is natural to construct $\mathcal{A} = \bigcup_{\hat{\mathsf{P}} \text{ satisfies some property}} \mathcal{T}_n(\hat{\mathsf{P}})$, and therefore

$$|\mathcal{A}| = \sum_{\hat{\mathsf{P}} \text{ satisfies some property}} |\mathcal{T}_n(\hat{\mathsf{P}})| \le \sum_{\hat{\mathsf{P}} \text{ satisfies some property}} 2^{n\mathrm{H}(\hat{\mathsf{P}})}.$$

Since we would like to have $|\mathcal{A}| \le 2^{\lfloor n\mathrm{R} \rfloor}$, we may choose those types with $\mathrm{H}\left(\hat{\mathsf{P}}\right) \le \mathrm{R}_n := \mathrm{R} - |\mathcal{S}| \frac{\log(n+1)}{n} - \frac{1}{n}$, so that

$$\begin{aligned} |\mathcal{A}| &\le \sum_{\hat{\mathsf{P}}\colon \mathrm{H}(\hat{\mathsf{P}}) \le \mathrm{R}_n} 2^{n\mathrm{H}(\hat{\mathsf{P}})} \\ &\le (n+1)^{|\mathcal{S}|} 2^{n\mathrm{R}_n} \\ &= 2^{n\mathrm{R}-1} \le 2^{\lfloor n\mathrm{R} \rfloor}. \end{aligned}$$

It remains to show that the probability of this set converges to 1 as $n \to \infty$, when the underlying source distribution P_S has entropy $\mathrm{H}(\mathsf{P}_S) < \mathrm{R}$.

Using the method of types, for a DMS sequence $S^n \sim \mathsf{P}_S^{\otimes n}$,

$$\begin{aligned}
\Pr\{S^n \notin \mathcal{A}\} &= \textstyle\sum_{\hat{\mathsf{P}}:\, \mathrm{H}(\hat{\mathsf{P}})>\mathrm{R}_n} \Pr\{S^n \in \mathcal{T}_n(\hat{\mathsf{P}})\} \\
&\le \textstyle\sum_{\hat{\mathsf{P}}:\, \mathrm{H}(\hat{\mathsf{P}})>\mathrm{R}_n} 2^{-n\mathrm{D}(\hat{\mathsf{P}}\|\mathsf{P}_S)} \\
&\le (n+1)^{|\mathcal{S}|} 2^{-n \inf_{\mathsf{P}:\, \mathrm{H}(\mathsf{P})>\mathrm{R}_n} \mathrm{D}(\mathsf{P}\|\mathsf{P}_S)} \\
&= \exp\left\{-n(\ln 2)\left(\inf_{\mathsf{P}:\, \mathrm{H}(\mathsf{P})>\mathrm{R}_n} \mathrm{D}(\mathsf{P}\|\mathsf{P}_S) - |S|\tfrac{\log(n+1)}{n}\right)\right\}. \qquad (2)
\end{aligned}$$

Observe that $\mathrm{R}_n \nearrow \mathrm{R}$ as $n \to \infty$. Hence, if $\mathrm{H}(\mathsf{P}_S) < \mathrm{R}$, $\mathrm{H}(\mathsf{P}_S) < \mathrm{R}_n$ for sufficiently large n. As a result,

$$\inf_{\mathsf{P}:\, \mathrm{H}(\mathsf{P})>\mathrm{R}_n} \mathrm{D}(\mathsf{P}\|\mathsf{P}_S) > 0$$

for sufficiently large n, and (2) $\to 0$ as $n \to \infty$. □

Remark: The above proof also says that the probability of error vanishes exponentially fast with error exponent $\min_{\mathsf{P}:\, \mathrm{H}(\mathsf{P})\ge\mathrm{R}} \mathrm{D}(\mathsf{P}\|\mathsf{P}_S)$, the value of an information projection problem!

Universal lossy source coding

The results can be extended to lossy source coding.

Idea: no need to enumerate all type classes with significant probability. Instead, use much fewer sequences in $\hat{\mathcal{S}}^n$ to *cover* type classes in $\mathcal{S}^n$.

Given a target distortion D, for a type $\hat{\mathsf{P}} \in \hat{\mathcal{P}}_n(\mathcal{S})$, let us use some $\mathcal{B} \subseteq \hat{\mathcal{S}}^n$ to *cover* the type class $\mathcal{T}_n(\hat{\mathsf{P}})$, so that

$$\min_{\hat{s}^n \in \mathcal{B}} d(s^n, \hat{s}^n) \le \mathsf{D}, \ \forall\, s^n \in \mathcal{T}_n(\hat{\mathsf{P}}).$$

In words, for each s^n with type $\hat{\mathsf{P}}$, $\exists\, \hat{s}^n \in \mathcal{B}$ that is a good "representative".

Question: how small can $\mathcal{B}$ be? It turns out for any $\delta > 0$, for n large enough, there exists $\mathcal{B}$ with

$$|\mathcal{B}| \le 2^{n\left(\mathrm{R}(\mathsf{D};\hat{\mathsf{P}})+\delta\right)}.$$

Here $\mathrm{R}(\mathsf{D};\mathsf{P})$ denotes the rate distortion function for DMS $S \sim \mathsf{P}$. The proof of this *type covering lemma* is omitted here.

Now we can propose a universal lossy source coding scheme similar to the lossless one to guarantee that, for a code rate R, any source $S \sim \mathsf{P}$ with rate distortion function $\mathrm{R}(\mathrm{D}; \mathsf{P}) < \mathrm{R}$, the average distortion is $\leq \mathrm{D}$ asymptotically.

Let $\mathcal{A} := \bigcup_{\hat{\mathsf{P}}:\, \mathrm{R}(\mathrm{D};\hat{\mathsf{P}}) \leq \mathrm{R}_n} \mathcal{T}_n(\hat{\mathsf{P}})$, with R_n defined as in the lossless case.

By the method of types, for $S_i \overset{\text{i.i.d.}}{\sim} \mathsf{P}$,

$$\Pr\{S^n \notin \mathcal{A}\} \leq (n+1)^{|\mathcal{S}|} 2^{-n\left(\inf_{\mathsf{Q}:\, \mathrm{R}(\mathrm{D};\mathsf{Q}) > \mathrm{R}_n} \mathrm{D}(\mathsf{Q}\|\mathsf{P})\right)}$$

which converges to 0 as $n \to \infty$ since $\mathrm{R}_n \nearrow \mathrm{R}$.

On the other hand, for a type $\hat{\mathsf{P}}$ with $\mathrm{R}(\mathrm{D}; \hat{\mathsf{P}}) \leq \mathrm{R}_n$, by the type covering lemma in the previous slide, we can find $\mathcal{B}_{\hat{\mathsf{P}}} \subseteq \hat{\mathcal{S}}^n$ such that $|\mathcal{B}_{\hat{\mathsf{P}}}| \dot{\leq} 2^{n(\mathrm{R}_n + \delta)}$ and

$$\min_{\hat{s}^n \in \mathcal{B}_{\hat{\mathsf{P}}}} d(s^n, \hat{s}^n) \leq \mathrm{D}, \ \forall\, s^n \in \mathcal{T}_n(\hat{\mathsf{P}}).$$

Finally, similar to the lossless case, we have

$$\left| \bigcup_{\hat{\mathsf{P}}:\, \mathrm{R}(\mathrm{D};\hat{\mathsf{P}}) \leq \mathrm{R}_n} \mathcal{B}_{\hat{\mathsf{P}}} \right| \leq 2^{n\mathrm{R}-1} \leq 2^{\lfloor n\mathrm{R} \rfloor}.$$

Summary

For fixed-to-fixed source coding, universality can be guaranteed by refining the achievability baed on the method of types.

The universal guarantee is that, for any sources with a fundamental limit (entropy or rate-distortion function) smaller than the rate of the scheme, the reconstruction criterion can be met without knowing the source distribution.

So even if we have some prior knowledge about the source such as its distribution belongs to a certain family, the best rate that can have universal guarantee is given by the worst source in that family.

Next: fixed-to-variable source coding.

1 Universal Fixed-to-Fixed Source Coding

2 Universal Fixed-to-Variable Source Coding

Compressing an individual sequence

Sometimes we just see a sequence of data without much knowledge about the underlying distribution. How to represent such an individual sequence?

In this case, fixed-to-fixed source coding does not make much sense, so we turn to *fixed-to-variable* source coding.

Recall: when the source distribution (distribution of S^n) is known, we can use

1. One-shot code construction: Assign ascending-length codewords to symbols in the order of descending probability (optimal).
2. Uniquely decodable (prefix-free) code construction: Shannon-type code, Huffman code (optimal).

It turns out that the expected codeword length $\mathsf{E}\left[k_{\mathrm{enc}}(S^n)\right] = \mathrm{H}(S^n) + O(1)$. Hence, the source entropy $\mathrm{H}(S^n)$ is a good baseline to compare with.

When the source distribution is completely unknown, we shall introduce universal coding schemes that achieves $\mathsf{E}\left[k_{\mathrm{enc}}(S^n)\right] = \mathrm{H}(S^n) + O(\log n)$.

A two-stage compression scheme

Lynch (1966) and Davisson (1966) proposed a simple two-stage method to compress an individual sequence s^n:

1. First, compute the *type* $\hat{\mathsf{P}}_{s^n}$ of the sequence s^n and use $\log_2|\hat{\mathcal{P}}_n(\mathcal{S})|$ bits to represent this type.
2. Second, use $\log_2|\mathcal{T}_n(\hat{\mathsf{P}}_{s^n})|$ bits to represent which sequence s^n is.

Expected codeword length:

$$\text{Part 1}: \quad \log|\hat{\mathcal{P}}_n(\mathcal{S})| \le (|\mathcal{S}|-1)\log(n+1) \qquad \text{(nothing random)}$$

$$\text{Part 2}: \quad \mathsf{E}\Big[\log|\mathcal{T}_n(\hat{\mathsf{P}}_{S^n})|\Big] \le n\mathsf{E}\Big[\mathrm{H}\Big(\hat{\mathsf{P}}_{S^n}\Big)\Big] \le n\mathrm{H}\Big(\mathsf{E}[\hat{\mathsf{P}}_{S^n}]\Big) \underset{\text{if } S_i \overset{\text{i.i.d.}}{\sim} \mathsf{P}}{=} n\mathrm{H}(\mathsf{P})$$

Hence, for S^n generated from DMS $S \sim \mathsf{P}$,

$$\underbrace{\mathsf{E}[k_{\text{enc}}(S^n)] - \mathrm{H}(S^n)}_{\text{expected redundancy}} \le (|\mathcal{S}|-1)\log(n+1) = O(\log n).$$

Expected redundancy

$$\mathrm{Red}_n := \mathsf{E}[k_{\mathrm{enc}}(S^n)] - \mathrm{H}(S^n)\,.$$

- When the distribution is known, the expected redundancy is $O(1)$.
- When the distribution is unknown, the expected redundancy is $O(\log n)$.

How to systematically design good coding schemes?

Let us focus on finding a good $\mathsf{Q} \in \mathcal{P}(\mathcal{S}^n)$ and, as before, use Shannon-type coding for example, to construct a code with

$$\mathsf{E}[k_{\mathrm{enc}}(S^n)] = \mathsf{E}\left[\log \tfrac{1}{\mathsf{Q}(S^n)}\right] + O(1).$$

Note: no loss of optimality to focus on finding Q, if we restrict to prefix-free (uniquely decodable) codes. This is because the code tree must be full, and hence the lengths of the prefix-free code define a distribution in $\mathcal{P}(\mathcal{S}^n)$.

Now we can rewrite the expected redundancy, ignoring a constant term, as

$$\mathsf{Red}_n = \mathsf{E}\left[\log \frac{1}{\mathsf{Q}(S^n)}\right] - \mathrm{H}(S^n) = \mathrm{D}\left(\mathsf{P}_{S^n} \| \mathsf{Q}\right).$$

So the problem boils down to finding some $\mathsf{Q} \in \mathcal{P}(\mathcal{S}^n)$ with "good" expected redundancy Red_n. We will comment on what "good" means later.

As a sanity check, we see that it is possible to achieve the redundancy of the Lynch-Davisson two-stage coding scheme with a good $\mathsf{Q} \in \mathcal{P}(\mathcal{S}^n)$ as follows.

Choose $\mathsf{Q}(s^n)$ inversely proportional to the size of the type class s^n belongs to, that is, $\mathsf{Q}(s^n) = |\mathcal{T}_n(\hat{\mathsf{P}}_{s^n})|^{-1}/c_n$ where c_n is the normalizing factor

$$c_n = \textstyle\sum_{s^n \in \mathcal{S}^n} |\mathcal{T}_n(\hat{\mathsf{P}}_{s^n})|^{-1} = |\hat{\mathcal{P}}_n(\mathcal{S})|.$$

The expected redundancy $\mathsf{Red}_n = \mathsf{E}\left[\log|\mathcal{T}_n(\hat{\mathsf{P}}_{S^n})|\right] + \log|\hat{\mathcal{P}}_n(\mathcal{S})| - \mathrm{H}(S^n)$ is the same as that of Lynch-Davisson and hence $O(\log n)$.

Other choices?

Minimax redundancy

(In the next few slides, let's remove the index n as it is not relevant for the moment.) To choose a "good" Q, one natural way is to find one such that the worst-case expected redundancy is minimized:

$$\overline{\mathsf{Red}}^* := \min_{\mathsf{Q}\in\mathcal{P}(\mathcal{S})} \max_{\mathsf{P}\in\mathcal{P}(\mathcal{S})} \mathrm{D}(\mathsf{P}\|\mathsf{Q}) .$$

More generally, the worst case is over a prescribed family of distributions $\{\mathsf{P}_\theta \,|\, \theta\in\Theta\} =: \mathcal{P}_\Theta \subseteq \mathcal{P}(\mathcal{S})$, and the formulated problem becomes

$$\overline{\mathsf{Red}}^* := \min_{\mathsf{Q}\in\mathcal{P}(\mathcal{S})} \max_{\mathsf{P}\in\mathcal{P}_\Theta} \mathrm{D}(\mathsf{P}\|\mathsf{Q}) \equiv \min_{\mathsf{Q}\in\mathcal{P}(\mathcal{S})} \max_{\theta\in\Theta} \mathrm{D}(\mathsf{P}_\theta\|\mathsf{Q}) . \tag{3}$$

It turns out that choosing the encoding probability $\mathsf{Q}\in\mathcal{P}(\mathcal{S})$ as the mixture of distributions in $\mathcal{P}_\Theta$ minimizes the worst-case expected redundancy, that is,

$$\overline{\mathsf{Red}}^* = \min_{\pi\in\mathcal{P}(\Theta)} \max_{\theta\in\Theta} \mathrm{D}(\mathsf{P}_\theta(\cdot)\|\mathsf{E}_{\Theta\sim\pi}[\mathsf{P}_\Theta(\cdot)]) .$$

Mixture achieves the minimax redundancy

Interestingly, the minimax redundancy is equal to capacity of a memoryless channel with Θ and S being the input and output respectively.

Theorem 2 (Gallager; Ryabko)

Consider a family of sources $\{\mathsf{P}_\theta \,|\, \theta \in \Theta\}$ *and a memoryless channel* $\left(\Theta, \mathsf{W}_{S|\Theta}, \mathcal{S}\right)$ *with* $\mathsf{W}_{S|\Theta}(\cdot|\theta) \equiv \mathsf{P}_\theta(\cdot)$*. The minimax redundancy in universal source coding and the channel capacity are equal, that is,*

$$\overline{\mathsf{Red}}^* = \max_{\mathsf{P}_\Theta \in \mathcal{P}(\Theta)} \mathrm{I}(\Theta; S) =: \mathrm{C}.$$

Furthermore, the optimal Q^* *to achieve* (3) *is the output distribution induced by the capacity achieving input distribution of the memoryless channel.*

pf: Let us begin with manipulating (3) based on the following observation:

$$\max_{\theta\in\Theta} \mathrm{D}\left(\mathsf{P}_\theta \| \mathsf{Q}\right) = \max_{\pi\in\mathcal{P}(\Theta)} \mathrm{D}\left(\mathsf{W}_{S|\Theta} \middle\| \mathsf{Q} \middle| \pi\right).$$

This is due to the non-negativity of $\mathrm{D}\left(\mathsf{P}_\theta \| \mathsf{Q}\right)$ for all $\theta \in \Theta$. Hence, (3) becomes

$$\overline{\mathsf{Red}}^* = \min_{\mathsf{Q}\in\mathcal{P}(\mathcal{S})} \max_{\pi\in\mathcal{P}(\Theta)} \mathrm{D}\left(\mathsf{W}_{S|\Theta} \middle\| \mathsf{Q} \middle| \pi\right). \tag{4}$$

Before we proceed, let us set up some notations. Let $\pi^* := \arg\max_{\mathsf{P}_\Theta} \mathrm{I}(\Theta; S)$ denote the capacity-achieving input distribution, and $\mathsf{Q}^* \in \mathcal{P}(\mathcal{S})$ denote the corresponding output distribution:

$$\Theta \sim \pi^* \to \boxed{\mathsf{W}_{S|\Theta}} \to S \sim \mathsf{Q}^*.$$

Also, with $\mathsf{W}_{S|\Theta}$ being fixed, consider $\mathrm{D}\left(\mathsf{W}_{S|\Theta} \middle\| \mathsf{Q} \middle| \pi\right) \equiv g(\mathsf{Q}, \pi)$ as a bivariate function of (Q, π).

To move forward from (4), the key is to show that (Q^*, π^*) is a **saddle point** of $g(\mathsf{Q}, \pi)$, that is,

$$g(\mathsf{Q}^*, \pi) \overset{\text{(a)}}{\leq} g(\mathsf{Q}^*, \pi^*) \overset{\text{(b)}}{\leq} g(\mathsf{Q}, \pi^*) \quad \forall\, \mathsf{Q} \in \mathcal{P}(\mathcal{S}), \pi \in \mathcal{P}(\Theta). \tag{5}$$

Once (5) is established, one can leverage the saddle point theorem to show that $\min_{\mathsf{Q}} \max_{\pi} g(\mathsf{Q}, \pi) = \max_{\pi} \min_{\mathsf{Q}} g(\mathsf{Q}, \pi)$ and $\mathsf{Q}^* \in \arg\min_{\mathsf{Q}} \max_{\pi} g(\mathsf{Q}, \pi)$.

Then, since $\min_{\mathsf{Q}\in\mathcal{P}(\mathcal{S})} g(\mathsf{Q}, \pi) \equiv \min_{\mathsf{Q}\in\mathcal{P}(\mathcal{S})} \mathrm{D}\left(\mathsf{W}_{S|\Theta} \middle\| \mathsf{Q} \middle| \pi\right) = \mathrm{I}(\Theta; S)$, we can conclude that

$$\overline{\mathsf{Red}}^* = \min_{\mathsf{Q}\in\mathcal{P}(\mathcal{S})} \max_{\pi\in\mathcal{P}(\Theta)} \mathrm{D}\left(\mathsf{W}_{S|\Theta} \middle\| \mathsf{Q} \middle| \pi\right) = \max_{\Theta\sim\pi\in\mathcal{P}(\Theta)} \mathrm{I}(\Theta; S) = \mathrm{C},$$

and the capacity-achieving *output* distribution

$$\mathsf{Q}^* \in \arg\min_{\mathsf{Q}\in\mathcal{P}(\mathcal{S})} \max_{\pi\in\mathcal{P}(\Theta)} \mathrm{D}\left(\mathsf{W}_{S|\Theta} \middle\| \mathsf{Q} \middle| \pi\right).$$

Since the channel output distribution is a mixture of $\{\mathsf{W}_{S|\Theta}(\cdot)|\theta) \,|\, \theta \in \Theta\}$, we can conclude that mixture of the source distributions is optimal for encoding.

Now it boils down to proving (5). (b) is straightforward since

$$\mathrm{D}\left(\mathsf{P}_{Y|X}\middle\|\mathsf{Q}_Y\middle|\mathsf{P}_X\right) = \mathrm{D}\left(\mathsf{P}_{Y|X}\middle\|\mathsf{P}_Y\middle|\mathsf{P}_X\right) + \mathrm{D}\left(\mathsf{Q}_Y\|\mathsf{P}_Y\right).$$

For (a), let $\pi_\lambda := \lambda\pi + (1-\lambda)\pi^*$, $\lambda \in (0,1)$, and Q_λ denote the corresponding channel output distribution, that is,

$$\Theta \sim \pi_\lambda \to \boxed{\mathsf{W}_{S|\Theta}} \to S \sim \mathsf{Q}_\lambda.$$

Hence,

$$\begin{aligned}\mathrm{C} \geq \mathrm{D}\left(\mathsf{W}_{S|\Theta}\middle\|\mathsf{Q}_\lambda\middle|\pi_\lambda\right) &= \lambda\mathrm{D}\left(\mathsf{W}_{S|\Theta}\middle\|\mathsf{Q}_\lambda\middle|\pi\right) + (1-\lambda)\mathrm{D}\left(\mathsf{W}_{S|\Theta}\middle\|\mathsf{Q}_\lambda\middle|\pi^*\right)\\ &\geq \lambda\mathrm{D}\left(\mathsf{W}_{S|\Theta}\middle\|\mathsf{Q}_\lambda\middle|\pi\right) + (1-\lambda)\mathrm{C},\end{aligned}$$

where the last inequality is due to (b). This leads $\mathrm{C} \geq \mathrm{D}\left(\mathsf{W}_{S|\Theta}\middle\|\mathsf{Q}_\lambda\middle|\pi\right)$.

The proof is complete by the lower semi-continuity of KL divergence:

$$\mathrm{C} \geq \liminf_{\lambda\to 0} \mathrm{D}\left(\mathsf{W}_{S|\Theta}\middle\|\mathsf{Q}_\lambda\middle|\pi\right) = \mathrm{D}\left(\mathsf{W}_{S|\Theta}\middle\|\mathsf{Q}^*\middle|\pi\right). \qquad \square$$

Asymptotic minimax redundancy for DMS

Now back to DMS generating a sequence of length n and the corresponding minimax redundancy $\overline{\mathrm{Red}}_n^*$. We are interested in its asymptote as $n \to \infty$.

Let the underlying DMS is drawn according to some P_θ in the parametric family $\{\mathsf{P}_\theta \,|\, \theta \in \Theta\}$. Hence, the probability of s^n is just $\mathsf{P}_\theta^{\otimes n}(s^n) = \prod_{i=1}^n \mathsf{P}_\theta(s_i)$.

By Theorem 2, one can use the mixture algorithm to construct the encoding probability $\mathsf{Q} \in \mathcal{P}(\mathcal{S}^n)$ "as if" there is a *prior distribution* π on the unknown parameter Θ:

$$\mathsf{Q}(s^n) = \mathsf{E}_{\Theta\sim\pi}\left[\mathsf{P}_\Theta^{\otimes n}(s^n)\right] = \int_\Theta \pi(\theta) \left(\prod_{i=1}^n \mathsf{P}_\theta(s_i)\right) \mathrm{d}\theta. \qquad \text{(assuming } \pi \text{ is a density)}$$

Clarke and Barron (1996) showed that an asymptotically optimal (least favorable) prior is *Jefferys' prior*

$$\pi^*(\theta) \propto \sqrt{\det \mathbf{J}(\theta)},$$

and it is unique among all positive continuous priors. Here $\mathbf{J}(\theta)$ denotes the *Fisher information matrix* of the parametric family.

The resulting asymptotic minimax redundancy is

$$\overline{\mathsf{Red}}_n^* = \frac{d}{2} \log \frac{n}{2\pi \mathrm{e}} + \log \int_\Theta \sqrt{\det \mathbf{J}(\theta)} \, \mathrm{d}\theta + o(1)$$

for a compact $\Theta \subseteq \mathbb{R}^d$.

For the extreme case, suppose we know nothing about the underlying DMS (except that it is memoryless) – it can be any $\mathsf{P} \in \mathcal{P}(\mathcal{S})$. Let us view P as a $|\mathcal{S}|$-dim probability vector $\boldsymbol{p}$ – an unknown parameter in the probability simplex $\mathcal{P}(\mathcal{S})$.

To emphasize this view, we may think of $\boldsymbol{p} \equiv \boldsymbol{\theta}$. Since there are only $|\mathcal{S}| - 1$ free parameters in $\boldsymbol{p}$, $\boldsymbol{\theta}$ is actually $(|\mathcal{S}| - 1)$-dimensional.

Using the result by Clarke and Barron, we see that even if we know nothing about the distribution of the underlying DMS, there is an universal coding scheme achieves the following worst-case expected redundancy

$$\overline{\mathsf{Red}}_n = \tfrac{1}{2}(|\mathcal{S}| - 1)\log n + \text{constant} + o(1),$$

Specializing to this case, the Fisher information matrix $\mathbf{J}(\theta)$ can be explicitly derived, and the corresponding Jefferys' prior is the *Dirichlet distribution* with parameter $(\frac{1}{2}, ..., \frac{1}{2})$:

$$\pi^*\left(p_1, ..., p_{|\mathcal{S}|}\right) \propto \left(\prod_{l=1}^{|\mathcal{S}|} p_l\right)^{-\frac{1}{2}}. \tag{6}$$

The above constant can be explicitly characterized as well.

Discussions

- It turns out that just to get $\overline{\mathrm{Red}}_n = \frac{d}{2}\log n + O(1)$, given that the parametric family consists of smooth distributions, a prior density that is strictly positive for all $\theta \in \Theta$ suffices. Such scaling results for Θ being the entire probability simplex were established in 1970-1980's.
- The lower bound (converse part) of the minimax redundancy $\overline{\mathrm{Red}}_n = \frac{d}{2}\log n + \Omega(1)$ is due to J. Rissanen (1984), where he established the results not only for the expected redundancy. Merhav and Feder (1995) improved the converse.
- There is a caveat in Clarke and Barron (1994) that Θ needs to be a compact subset of the ambient space. This rules out the case of Θ being the whole probability simplex. Consequently, the Jefferys' prior in (6) is just asymptotically maximin, not minimax. Xie and Barron (1997) fixed this issue and show that a slight modification of Jeffeys' prior achieves asymptotic minimax optimality.

A good reference:

2124 IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 44, NO. 6, OCTOBER 1998

Universal Prediction

Neri Merhav, *Senior Member, IEEE*, and Meir Feder, *Senior Member, IEEE*

(Invited Paper)

In the special issue that celebrates 50 years of the birth of information theory.

Online compression via arithmetic coding

An issue in the previously mentioned universal source coding schemes: it is offline, that is, the encoder must first see the entire sequence s^n and then it can output the codeword.

Such offline algorithms are not very efficient and incur huge delay and storage cost both in encoding and decoding.

Arithmetic coding: a simple online algorithm due to P. Elias that encodes a source sequence on the fly, given a *coding distribution* $\mathrm{Q} \in \mathcal{P}(\mathcal{S}^n)$ of the sequence. Moreover, for any n, it gives a prefix-free code with individual codeword lengths

$$k_{\mathsf{arith}}(s^n) = \left\lceil \log \frac{1}{\mathrm{Q}(s^n)} \right\rceil + 1.$$

Hence, it is 1 bit longer than a Shannon-type code, and hence the expected codeword length is at most 2 bits longer than $\mathrm{H}(\mathrm{Q})$ if Q is the true probability distribution of the source sequence.

Lexicographical ordering and source intervals

Arithmetic coding relies on lexicographically ordering the source sequence: let $\mathcal{S} = \{1, ..., d\}$ WLOG. Then, the lexicographic order $\prec$ is defined as follows:

$$\tilde{s}^n \prec s^n \iff \tilde{s}_i < s_i \text{ for the first } i \text{ such that the two sequences disagree}$$

Then, for a given coding distribution $\mathsf{Q} \in \mathcal{P}(\mathcal{S}^n)$, we can define a corresponding *source interval*

$$\mathcal{I}(s^n) := [\mathsf{F}(s^n), \mathsf{F}(s^n) + \mathsf{Q}(s^n)) \,,$$

where

$$\mathsf{F}(s^n) := \sum_{\tilde{s}^n \prec s^n} \mathsf{Q}(\tilde{s}^n).$$

By construction, the source intervals form a partition of the unit interval $[0, 1)$.

Codeword assignment

The disjointness of the source intervals hints to find a "code interval"

$$\mathcal{C}(s^n) := \left[c(s^n), c(s^n) + 2^{-l(s^n)}\right) \subseteq \mathcal{I}(s^n) := [\mathsf{F}(s^n), \mathsf{F}(s^n) + \mathsf{Q}(s^n))$$

where $c(s^n)2^{l(s^n)} \in \mathbb{Z}$, that is, the binary representation of $c(s^n)$ has 0's all the way after the $l(s^n)$-th bit below the decimal point.

Using the first $l(s^n)$ below the decimal point of the binary representation of $c(s^n)$ as the codeword $\mathrm{enc}(s^n)$, we are guaranteed to have a prefix-free code.

How to choose $c(s^n)$ and $l(s^n)$ to achieve the above?

- $c(s^n)$ should be the "up-truncation" of $\mathsf{F}(s^n)$ with up to the $l(s^n)$-th bit below the decimal point $\implies c(s^n) = \left\lceil \mathsf{F}(s^n)2^{l(s^n)} \right\rceil 2^{-l(s^n)}$.
- Due to up-truncation, in order to ensure the code interval $\mathcal{C}(s^n) \subseteq \mathcal{I}(s^n)$, we need $2^{-l(s^n)+1} \le \mathsf{Q}(s^n) \implies l(s^n) - 1 \ge \log_2 \frac{1}{\mathsf{Q}(s^n)}$. We shall choose $l(s^n) = \lceil \log \frac{1}{\mathsf{Q}(s^n)} \rceil + 1$.

Sequential Encoding

Summary of codeword assignment: $\mathcal{C}(s^n) = \left[c(s^n), c(s^n) + 2^{-l(s^n)}\right)$, where

$$c(s^n) = \left\lceil \mathsf{F}(s^n) 2^{l(s^n)} \right\rceil 2^{-l(s^n)},\ l(s^n) = \left\lceil \log \frac{1}{\mathsf{Q}(s^n)} \right\rceil + 1.$$

Key quantities: $\mathsf{Q}(s^n)$ and $\mathsf{F}(s^n) = \sum_{\tilde{s}^n \prec s^n} \mathsf{Q}(\tilde{s}^n)$.

The question is: can they be computed sequentially? Observe the following:

$$\begin{aligned}
\mathsf{Q}(s^n) &= \mathsf{Q}(s_n|s^{n-1})\mathsf{Q}(s^{n-1}) \\
\mathsf{F}(s^n) &= \sum_{\tilde{s}^n \prec s^n} \mathsf{Q}(\tilde{s}^n) \\
&= \sum_{\tilde{s}^n : \tilde{s}^{n-1} \prec s^{n-1}} \mathsf{Q}(\tilde{s}^n) + \left(\sum_{i<s_n} \mathsf{Q}(i|s^{n-1})\right) \mathsf{Q}(s^{n-1}) \\
&= \sum_{\tilde{s}^{n-1} \prec s^{n-1}} \mathsf{Q}(\tilde{s}^{n-1}) + \left(\sum_{i<s_n} \mathsf{Q}(i|s^{n-1})\right) \mathsf{Q}(s^{n-1}) \\
&= \mathsf{F}(s^{n-1}) + \left(\sum_{i<s_n} \mathsf{Q}(i|s^{n-1})\right) \mathsf{Q}(s^{n-1}).
\end{aligned}$$

As long as the conditional probability $\mathsf{Q}(\cdot|s^{n-1})$ can be computed efficiently, the whole encoding process can be done efficiently in a sequential manner.

Sequential probability computation

If the underlying Q is memoryless (i.i.d.) or finite-order Markov, then it is straightforward to compute $\mathsf{Q}(\cdot|s^{n-1})$.

But for the mixture distribution used in universal source coding, it does not have finite-depth memory. Can $\mathsf{Q}(\cdot|s^{n-1})$ be computed efficiently, if the joint $\mathsf{Q}(s^n) = \int_\Theta \pi(\theta)\mathsf{P}_\theta(s^n)\,\mathrm{d}\theta$ is a mixture distribution?

It turns out that there is an elegant way to do so (we use subscript $()_\pi$ to emphasize the dependency on prior π):

$$\begin{aligned}\mathsf{Q}_\pi(s_n|s^{n-1}) &= \frac{\mathsf{Q}_\pi(s^n)}{\mathsf{Q}_\pi(s^{n-1})} = \frac{1}{\mathsf{Q}_\pi(s^{n-1})}\int_\Theta \pi(\theta)\mathsf{P}_\theta(s^n)\,\mathrm{d}\theta \\ &= \int_\Theta \frac{\pi(\theta)\mathsf{P}_\theta(s^{n-1})}{\int_\Theta \pi(\theta')\mathsf{P}_{\theta'}(s^{n-1})\,\mathrm{d}\theta'}\,\mathsf{P}_\theta(s_n|s^{n-1})\,\mathrm{d}\theta.\end{aligned}$$

It is just the mixture of conditional distributions, mixed with (weighted by) the *posterior* density $\pi(\theta|s^{n-1}) = \frac{\pi(\theta)\mathsf{P}_\theta(s^{n-1})}{\int_\Theta \pi(\theta')\mathsf{P}_{\theta'}(s^{n-1})\,\mathrm{d}\theta'}$.

Hence the only thing need to be efficiently computed is $\mathsf{P}_\theta(s_n|s^{n-1})$. For a given θ, usually S^n has finite depth of memory (otherwise we lose the point of doing mixture), and hence $\mathsf{P}_\theta(s_n|s^{n-1})$ is simple to compute.

In particular, for a mixture of memoryless distributions, $\mathsf{P}_\theta(s_n|s^{n-1}) = \mathsf{P}_\theta(s_n)$.

For the Jefferys' prior given in (6), it turns out that there is a simple explicit expression for the mixture conditional distribution:

$$\mathsf{Q}_{\pi^*}(i|s^{n-1}) = \frac{n(i|s^{n-1}) + \frac{1}{2}}{(n-1) + \frac{d}{2}}.$$

Summary: for the mixture algorithm with Jeffery's prior (and many other priors), the encoding part can be done in an online fashion by using arithmetic coding, and achieves worst-case expected redundancy

$$\overline{\mathsf{Red}}_n = \tfrac{1}{2}(|\mathcal{S}| - 1)\log n + \text{constant} + o(1).$$

Codebook-free sequential decoding

Prefix-free code can be instantaneously decoded. So for a fixed n, sequential decoding is possible if there are multiple length-n blocks.

However, the main point of online algorithms like arithmetic coding is that, n is dynamic, and hence there is no fixed codebook (in the case of prefix-free code, it is the code tree) to be stored at the decoder.

How to decode without a codebook? It turns out there is a companion online decoding algorithm for arithmetic coding.

To see this, make the following observation:

$$\begin{aligned}\mathsf{F}(s^n) &= \mathsf{F}(s^{n-1}) + \underbrace{\left(\sum_{i<s_n} \mathsf{Q}(i|s^{n-1})\right)}_{f_n(s_n|s^{n-1})} \mathsf{Q}(s^{n-1}) \\ &= f_1(s_1) + \mathsf{Q}(s_1) f_2(s_2|s_1) + \ldots + \mathsf{Q}(s^{n-1}) f_n(s_n|s^{n-1})\end{aligned}$$

$$\implies \mathsf{F}(s^n) \in [f_1(s_1), f_1(s_1+1)).$$

Hence, if we know $\mathsf{F}(s^n)$, we can first find s_1 by finding the unique $s_1 \in \mathcal{S} = \{1, ..., d\}$ such that $\mathsf{F}(s^n) \in [f_1(s_1), f_1(s_1 + 1))$ and then do

$$\begin{aligned}\frac{\mathsf{F}(s^n) - f_1(s_1)}{\mathsf{Q}(s_1)} &=: \mathsf{F}(s_2^n | s_1) \\ &= f_2(s_2|s_1) + \mathsf{Q}(s_2|s_1) f_3(s_3|s_1, s_2) ... + \mathsf{Q}(s_2^{n-1}|s_1) f_n(s_n|s^{n-1}).\end{aligned}$$

Then compute $\mathsf{Q}(\cdot|s_1)$, update the marginal of s_2 to this, and continue the previous process. Recursively we can decode $s_1, s_2, ...s_n$.

Caveat: at the decoder, we do not have the exact $\mathsf{F}(s^n)$. Instead, we have the codeword $c(s^n)$, which is an "up-truncation" of $\mathsf{F}(s^n)$.

Fortunately, since

$$f_1(s_1) \le \mathsf{F}(s^n) \le c(s^n) < \mathsf{F}(s^n) + \mathsf{Q}(s^n) \le f_1(s_1 + 1),$$

we still have $c(s^n) \in [f_1(s_1), f_1(s_1 + 1))$. Hence, the above procedure can be applied with $\mathsf{F}(s^n)$ replaced by its truncation, $c(s^n)$.