Introduction to Information Theory : list of topics/projects for the oral examination

mailto:jean-pierre.tillich@inria.fr

1 Theoretical subject where there are things to prove

If you choose one of the projects of this section you are asked to give an oral presentation explaining the proof of a result in Information Theory taken from the following list.

1.1 Topic 1: proof of Shannon's second theorem for linear codes

The alphabet $\mathcal{A}$ for the symbols that are transmitted through a noisy channel is assumed to be of cardinality q which is a power of a prime number p, i.e. $q = p^m$ for a certain positive integer m. This implies that there exists a finite field $\mathbb{F}_q$ with exactly q elements. Show that if we are interested in using *linear codes* over $\mathbb{F}_q$, then there exists a Shannon theorem adapted to this restricted class of codes where we have to replace the capacity of a channel by $I(U;Y)$, where U is a uniformly distributed random variable over $\mathbb{F}_q$ and Y is the corresponding channel output.

1.2 Topic 2: Upper bound on the bit error probability after decoding

Consider the following scenario. We are interested here in transmitting bits through a memoryless channel of capacity C. We want to transmit information at a rate R that is above capacity, i.e. $R > C$. Of course this means that we have to tolerate large probabilities of error after decoding. We encode here k bits of information $u = u_1, u_2, \ldots, u_k$ with a codeword $x = x_1, x_2, \ldots, x_n$ of length n. Here $R = \frac{k}{n}$. Let y be the received word when we transmit x through the noisy channel and let $\hat{u} = \hat{u}_1, \ldots, \hat{u}_k$ be the binary output of the decoder. The average bit error probability is given by

$$p_b = \frac{1}{k} \sum_{i=1}^{k} \mathbf{prob}(u_i \neq \hat{u}_i)$$

Show that

$$1 - h(p_b) \leq \frac{C}{R}$$

where h is the binary entropy function $h(x) = -x \log_2 x - (1-x) \log_2 (1-x)$.

Hint.
Show that

$$I(x, y) \geq I(u; \hat{u})$$

by using Section 2.8 of the Cover and Thomas' book *Information Theory.* Then show that

$$I(x; y) \leq nC$$

Finally show that

$$I(u; \hat{u}) \geq k(1 - h(p_b))$$

by using a certain result that you can find in Chapter 13 "'Rate distortion theory" in Cover and Thomas' book.

2 Projects where you have to present an article

For the projects of this list you have to present what you understood from the following articles (you choose of course only one topic). **Important note:** for topic 1, you are allowed to implement the Burrows-Wheeler compression algorithm (instead of presenting the article) and to present your implementation (this is in the spirit of a "projet informatique").

2.1 Topic 1 : Source coding with the Burrows-Wheeler transform

This algorithm is more recent and is generally better than Lempel-Ziv is many cases (it is used in `bzip2` for instance).

References.
The article of Burrows Wheeler `http://www.hpl.hp.com/techreports/Compaq-DEC/SRC-RR-124.pdf`
An article about this article `https://marknelson.us/posts/1996/09/01/bwt.html`

2.2 Topic 2: Proof of the optimality of Lempel-Ziv 1978

There are lecture notes by Shor at the MIT proving the optimality of Lempel-Ziv in the case of a memoryless source. A proof in a much general setting is given in Chapter 10 (see Subsection 10.2) in Cover and Thomas' book.

References

`papers/lempel_ziv_notes.pdf`

2.3 Topic 3: Relationship between error correction in the worst case and error correction for "typical" errors

This is also a simple and nice tutorial about using the probabilistic method in coding theory.

References. `http://people.ee.ethz.ch/~loeliger/localpapers/MasseyBirthday.pdf`

2.4 Topic 4: LDPC codes

A tutorial about an important family of error correcting codes.

References. `http://arizona.openrepository.com/arizona/handle/10150/607470`

2.5 Topic 5: Codes for distributed storage systems

The codes presented here are used in the Ceph storage platform.

References. `https://www.usenix.org/system/files/conference/fast18/fast18-vajha.pdf` You can also have a look a more mathematically oriented oriented article `https://arxiv.org/abs/1605.08630`.

2.6 Topic 6: Information Theory and Statistical Mechanics

The work which is proposed here is to present Chapter 6 "Random code ensemble" of Andrea Montanari and Marc Mézard's book *Information, Physics and Computation* (2009, Oxford University Press) which can be found at `http://www.stanford.edu/~montanar/RESEARCH/book.html` The chapter which is of interest to you is in `http://www.stanford.edu/~montanar/RESEARCH/BOOK/partB.pdf`

2.7 Topic 7: Quantum Codes

The toric code belongs to the family of quantum surface codes and is particularly popular family of quantum codes due to its simple structure and good error correction capacity. You might find a short description of this code p.18-20 in `http://www.math.u-bordeaux.fr/~zemor/QuantumCodes.pdf`. You are asked to present this family of codes, prove their minimum distance and explain as rigorously as possible how they can be decoded from the sketchy description given in these notes.

2.8 Topic 8: Code-based Cryptography

Decoding a linear code in general is considered as a very difficult problem that can be used as a building block for devising cryptographic schemes. Here it is asked to understand and present the notes written by G. Zémor about Alekhnovich's cryptosystems, which is a rather old cryptosystem. But many most modern code-based cryptosystems are a variation of it. The nice feature is that the security of the two cryptosystems presented here can be directly linked to the difficulty of decoding.

3 A mix of programming/theoretical study

In this project, you are asked to find an answer to the following problem that comes from DNA digital storage. This consists in encoding and decoding data to and from synthesized DNA strands. The amount of information that can be encoded in a single gram of DNA is really impressive, it is of order 10^{17} bytes (hundreds of petabytes!). Storing information in this way might very well be a way to handle a few worrisome problems that start to appear in big data storage (amount of energy that is needed, lifetime of standard storage solutions, whether the actual solutions will be able to cope with the big amount of data produced in ten years from now,...).

Encoding of data in DNA strands can be modelled by a one-to-one mapping f from the set $\{0,1\}^*$ of binary strings to the set of words $\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}^*$ over the 4 nucleobases that can be found in DNA, namely `A`, `C`, `G` and `T`. The problem that arises is here that for several reasons not all words in $\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}^*$ are desirable. The set of words that are obtained by f should be *admissible*, meaning that

1. neither `ATG`, nor `TATAAT`, nor `CCCAT` nor `TTGCA` should be subwords of a produced word $f(x)$. For instance, $f(x)$ is not allowed to be equal to `CGGATGCAT` but could be `ACTG`.
2. For any $\varepsilon > 0$ the probability that $\frac{|f(x)|_{\mathtt{C},\mathtt{G}}}{|f(x)|}$ is not in $[\frac{1}{2}-\varepsilon, \frac{1}{2}+\varepsilon]$ when x is chosen uniformly at random in the set $\{0,1\}^n$ of binary words of length n tends to 0 as n tends to infinity. Here $|f(x)|$ is the length of the word $f(x)$ and $|f(x)|_{\mathtt{C},\mathtt{G}}$ is the number of symbols equal to `C` or `G` in it. For instance when $f(x) = \mathtt{AACCTCGC}$, we have $|f(x)| = 8$ and $|f(x)|_{\mathtt{C},\mathtt{G}} = 5$.
3. a symbol is never repeated in a sequence $f(x)$ more than 4 times in a row. For instance `TAAAATTA` is allowed but not `TCGAAAAACC`.

DNA storage works as follows: when one wants to store the binary word w the sequence $f(x)$ is synthesized and stored. To recover the information $f(x)$ is read and $x = f^{-1}(f(x))$ is computed.

Let $\bar{f}_n$ be the expected length of $f(x)$ when x is chosen uniformly in $\{0,1\}^n$. Let $\varphi_n \overset{\text{def}}{=} \frac{\bar{f}_n}{n}$ and $\varphi_\infty \overset{\text{def}}{=} \lim_{n\to\infty} \varphi_n$. You are now asked to answer the following questions.

1. Give the best lower bound you can find for φ_∞ for an f that produces only admissible words. This might involve some programming.
2. Implement an encoding function f for which φ_∞ is close to this lower bound. The program should take as input a binary string and output an admissible sequence in $\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}^*$.

You are also allowed here to give a solution that skips Condition 2, but in this case you are asked to give the limit as n tends to infinity of the ratio $\frac{\mathbb{E}(|f_n(x)|_{\mathtt{C},\mathtt{G}})}{\mathbb{E}(|f_n(x)|)}$ that you obtain. Here $\mathbb{E}(X)$ stands for the expectation of the random variable X.

Subsidiary question: There is an additional problem that arises: when $f(x)$ is read, it might be that some of symbols are read incorrectly and that an error-correcting code might be needed. For this purpose a special kind of encoding function might be used. It first consists in grouping the binary data in binary packets of size m (the length n of the binary sequence is supposed to be a multiple of m). The binary sequence $x = x_1 \cdots, x_n$ of length n is now viewed as a sequence of length n/m over the finite field $\mathbb{F}_{2^m}$: $u = u_1 \cdots u_{n'}$ with $u_i \in \mathbb{F}_{2^m}$ and $n' = n/m$. One chooses now a function g of the form

$$g : \{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}^{m'} \to \mathbb{F}_{2^m}$$

which is such that there is a one-to-one mapping f' that maps *any* sequence $u = u_1 \cdots u_{n'}$ of elements in $\mathbb{F}_{2^m}$ into an *admissible* sequence $v_1 \cdots v_{n'}$ of elements in $\{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}\}$ where we have for all i that $v_i \in g^{-1}(u_i)$. Give an example of such a g and such an f'. Explain how this can be used to perform error-correction. As an additional subsidiary question you might want to find such a g and an m such that the ratio $\frac{m'}{m}$ is close to φ_∞ that you have found in the previous question.