TUTORIAL I

1 Performance of repetition codes

We consider a Binary Symmetric Channel with parameter f.

1. How many times do I need to repeat to make the error per bit go below some p_b?

   A: *We can approximate the probability of error by the probability that $\lceil N/2 \rceil$ bits are flipped, which is equal to (for N odd):*

   $$\binom{N}{\lceil N/2 \rceil} f^{(N+1)/2}(1-f)^{(N-1)/2}$$

   Using Stirling's approximation, we have:

   $$p_b \simeq \frac{2^N}{\sqrt{\pi N/2}} f[f(1-f)]^{(N-1)/2} \simeq \frac{1}{\sqrt{\pi N/8}} f[4f(1-f)]^{(N-1)/2}$$

   The equation in N is thus :

   $$N \simeq 2\frac{p_b + \log \frac{\sqrt{\pi N/8}}{f}}{\log 4f(1-f)} + 1$$

   To find a good solution, we can start by over-estimate N by dropping the $\log \sqrt{N}$ term, which gives us $N = 2\frac{p_b}{\log 4f(1-f)} + 1$ and then iterate this equation until a stable solution is found.

   Numerical application: for $p_b = 10^{-15}$, initial N is equal to 69, and the iterated solution is $N \simeq 60.9$. Hence a block size of 61 gives us a probability of error less than 10^{-15} (which is 30 times bigger than the optimal solution given by Shanon's theorem).

2. Justify that the majority vote is the optimal decoder for the average error probability (i.e., we take a uniform prior on the message).

   A: *Given a binary string s of size n, containing k 0's and $n-k$ 1's, we compute the probability that the initial message m was a 0 using Bayes' theorem:*

   $$\mathbf{P}_m(0|s) = \frac{\mathbf{P}_{out}(s|m=0) \times \mathbf{P}(m=0)}{d}$$

   Here the denominator d isn't important since we want to compare $\mathbf{P}_m(0|s)$ with $\mathbf{P}_m(1|s)$, and both gives the same d, so we only have to compare $\mathbf{P}_{out}(s|m=0)\mathbf{P}(m=0)$ with $\mathbf{P}_{out}(s|m=1)\mathbf{P}(m=1)$.
   We made the assumption that the message has a uniform prior, so $\mathbf{P}(m=0) = \mathbf{P}(m=1) = 1/2$ and we only have left to compare $\mathbf{P}_{out}(s|m=0)$ with $\mathbf{P}_{out}(s|m=1)$. Those probabilities can be computed in a direct way:

   $$\mathbf{P}_{out}(s|m=0) = f^{n-k}(1-f)^k \quad \text{and} \quad \mathbf{P}_{out}(s|m=1) = f^k(1-f)^{n-k}$$

   We now study the function $g(x) = f^{n-x}(1-f)^x$ since we have $\mathbf{P}_{out}(s|m=0) = g(k)$ and $\mathbf{P}_{out}(s|m=1) = g(n-k)$. We have $g'(x) = (\log(1-f) - \log(f))g(x)$ and since $g(x) > 0$ for $x > 0$ the sign of $g'(x)$ depends only of the expression $\log(1-f) - \log(f)$. By studying the associated function, we can see that this expression is negative for $f < 1/2$ and positive for $f > 1/2$. Thus, if we take $f < 1/2$, the function $g(x)$ is non-increasing, and hence the highest probability is the one associated with the maximum between k and $n-k$, i.e. it coincides with the majority bit of s.

2 Weighing problem

You are given 12 balls, all equals in weight except for one that is either heavier or lighter. You are also given a classical two-pan balance which allow you only to compare two subset of balls (you are not given any reference weight). Your task is to design a strategy to determine which is the odd ball *and* whether it is heavier or lighter, using as few uses of the balance as possible.

1. What is the amount of uncertainty of a configuration ?

   A: *There are 24 possible configurations. Since a single use of the balance can result in three different ways, we take the logarithm in base 3, hence the uncertainty is :* $\log_3(24) \simeq 2,89$.

2. How much average information can a single use of the balance give ? What is the minimum number of weighing one can hope to achieve ?

   A: *Each use of the balance can give in average one 3-bit of information (in the case of a uniform distribution), thus one can hope to achieve a strategy with only 3 weighing.*

3. Show that if we start by weighing balls 1-6 against 7-12, we don't get enough information to achieve the optimal solution.

   A: *If we compute the probability distribution of the outcome of this weighing, we see that:*

   $$\mathbf{P}(1-6>7-12) = \mathbf{P}(1-6<7-12) = \frac{1}{2} \quad \textit{and} \quad \mathbf{P}(1-6=7-12) = 0$$

   The information obtained is :

   $$-\frac{1}{2}\log_3\frac{1}{2} - \frac{1}{2}\log_3\frac{1}{2} - 0\log_3 0 = \log_3 2 \simeq 0.63$$

   The remaining uncertainty is thus : $2.89 - 0.63 = 2.26 > 2$*, and so we can't determine the odd ball using only two more rounds.*

4. Describe an optimal strategy.

   A: *We want to maximise the amount of information obtained at every round, so we have to choose the weighing which gives a probability distribution for the outcome the closest for the uniform distribution. For the first round, we can take any two 4-subsets of balls, 1234 and 5678 for instance, which gives exactly the uniform distribution, so we obtain one 3-bit of information.*
   For the second round, we can't achieve a uniform distribution since we have 8 remaining situations, which is not a multiple of 3. We thus divide in 3-3-2, which will give us in average $-\frac{3}{4}\log_3\frac{3}{8} - \frac{1}{4}\log_3\frac{1}{4} \simeq 0.985$ *bits of information. The last round is quite straightforward.*

5. Compute the exact information obtained during the process depending on the result of the second round (3 or 2 remaining situations).

   A: *The first round gives 1 bit of information.*

   - *First case (3 situations remaining): the probability is* $3/8$*, thus the information obtained on the second round is* $\log_3(8/3) \simeq 0.89$*. For the last round, we can achieve a uniform distribution, which gives another 3-bit of information.*
     Overall, we have obtained $1 + 0.89 + 1 = 2.89$ *bits of information.*
   - *Second case (2 situations remaining) : the probability is* $2/8$*, thus the information obtained is* $\log_3(4) \simeq 1.26$*. In the last round, only two outcome are possible, thus the information obtained is only* $\log_3(2) \simeq 0.63$*.*
     Overall, we have obtained $1 + 1.26 + 0.63 = 2.89$ *bits of information.*

   In both case, the information obtained is exactly the uncertainty of a configuration.

3 Find query

We consider a list of 32 elements and we want to find the position of the element in the list. We assume that all the positions are equiprobable. Our strategy is to test the first element, then the second element, ... until the wanted element is found.

1. How many bit do we need to describe a position/what is the amount of uncertainty ?

   A: $\log 32 = 5$ *bits of uncertainty.*

2. What is the probability distribution of the outcome on the n^{th} test ?

   A: *On the first test, we have* $P_{X_0}(1) = \frac{1}{32}$ *and* $P_{X_0}(0) = \frac{31}{32}$.
   On the second test, we have $P_{X_1|X_0=0}(1) = \frac{1}{31}$ *and* $P_{X_1|X_0=0}(0) = \frac{30}{31}$.
   More generally, we have $P_{X_n|X_0=\ldots=X_{n-1}=0}(1) = \frac{1}{32-n}$

3. How much information do we obtain on the first test if we find the element ? And if we don't find it ?

   A: *We investigate the two possible cases:*

   - *If we find the element : we have obtained* $h_{X_0}(1) = \log 32 = 5$ *bits of information, which is normal because we have removed all the uncertainty, since we know the position of the element.*
   - *If we don't find the element : we have obtained* $h_{X_0}(0) = \log 32 - \log 31 \simeq 0.0458$ *bits of information.*

4. Same question for the second test.

   A: *We investigate the two possible cases:*

   - *If we find the element : we have obtained* $h_{X_1}(1) = \log 31$ *bits of information on this round, and a total of* 5 *bits.*
   - *If we don't find the element : we have obtained* $h_{X_1}(0) = \log 31 - \log 30$ *bits of information.*

5. Gave a general formula for the total amount of information obtained in the case $X_0 = X_1 = \ldots = X_n = 0$

   A: *Using question 2, the amount of information obtain during the* n^{th} *round is* $h_{X_n}(0) = \log(32-n) - \log(31-n)$. *When we add up this telescoping series, we find* $\log(32) - \log(31-n)$ *bits of information obtained.*

6. How much information has been obtained for $n = 15$. Give an interpretation of this result.

   A: *Using previous formula, we have* $\log(32) - \log(16) = \log(2) = 1$ *bit of information obtained. This can be viewed as the first bit describing the position of the element, since we know that the element doesn't belong to the first half. Having 1 bit information reduce our problem to the same one with only 16 positions available.*

7. How much information is obtained in the case $X_0 = X_1 = \ldots = X_n = 0, X_{n+1} = 1$?

   A: *We already have computed the information obtained for* $X_0 = X_1 = \ldots = X_n = 0$, *so we only have to compute the information obtained for* $X_{n+1} = 1$ *and to add up those numbers. Using question 2, we have* $h_{X_{n+1}}(1) = \log(32-(n+1))$. *The total number of information bits is thus :* $\log(32) - \log(31-n) + \log(32-(n+1)) = \log(32) = 5$.

4 Introducing typical sets

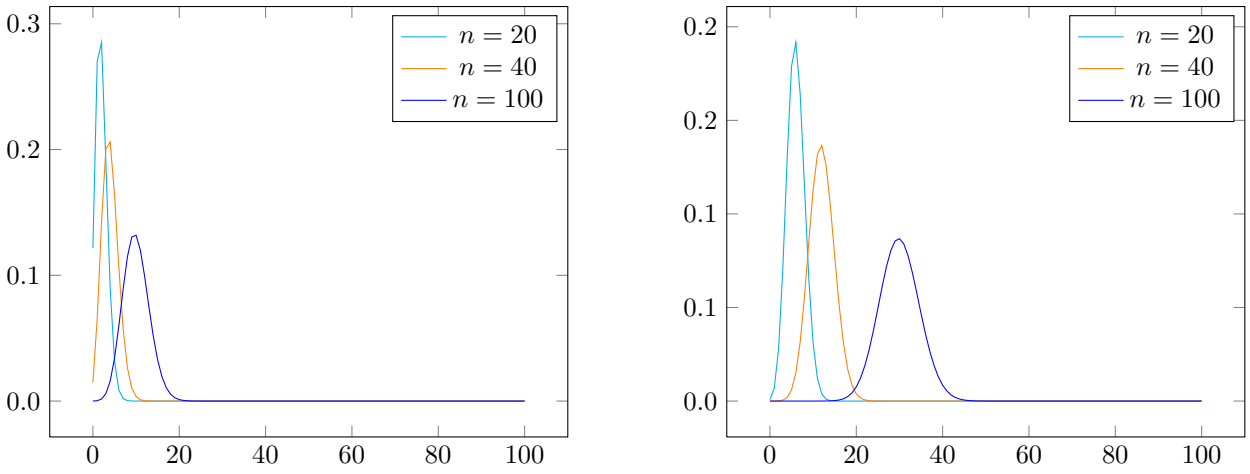
Let $X = X_1 \ldots X_n$ be independent and identically distributed bits with $X_1 \sim \mathrm{Ber}(p)$, i.e., $P_{X_1}(0) = p$ and $P_{X_1}(1) = 1 - p$ (assume wlog that $p > 1/2$). Your task is to prepare a bag $S \subseteq \{0,1\}^n$ with papers with each one having a binary string of length n. You win the game if $X \in S$.

1. Suppose your bag is restricted to have size 1. What would you write on the piece of paper?

A: *The most likely string of size n is $0\dots0$, which gives a probability of success of p^n.*

2. If I want to win with probability 0.9, how small can I make the bag be? In general, as a function of δ, give an approximation of the size of the smallest bag S_δ that can win the game with probability $\geq 1-\delta$.

A: *We use a greedy approach: we take the most likely strings in our bag until the probability is larger than 0.9. The most likely string is $0\dots0$, which has probability of outcome of p^n. Then we have all the strings that have only one 1, which have probability of outcome of $p^{n-1}(1-p)$, and there are n of them. Trying to compute the number of strings added to obtain 0.9 is rather difficult is this way, so we will use an approximation: we can see on the plot of the binomial laws that the strings that add a lot of probability are those which have a number of 1 near the mean of the distribution, which is pn. This means that even though the string $0\dots0$ is the most likely, it's the only one of this form and so its contribution is negligeable.*



Binomial laws with parameters $p = 0.1$ and $p = 0.3$.

For instance, let's take only the string which have pn 1's, there are $\binom{n}{pn}$ of them, which is, using stirling approximation:

$$\binom{n}{pn} \simeq \frac{1}{\sqrt{2\pi np(1-p)}} 2^{n\log n - pn\log(pn) - (1-p)n\log((1-p)n)} \simeq \frac{1}{\sqrt{2\pi np(1-p)}} 2^{nh(p)}$$

Each of these string have a probability of $p^{np}(1-p)^{n(1-p)} = 2^{-nh(p)}$, so it gives a first approximation: we can achieve a probability of $\frac{1}{\sqrt{2\pi np(1-p)}}$ using only $2^{nh(p)}$ strings.

A more precise solution can be found using Chebyshev inequality and the weak law of large numbers. Instead of taking only the strings with highest probability, we will also take strings with almost highest probability: we view the random variable $\frac{1}{n}\log\frac{1}{\mathbf{P}(s)}$ as the average of n informations contents $h_n = \log\frac{1}{\mathbf{P}(s_n)}$, each of which is a random variable with mean $H = h(p)$ and variance σ^2. We use the weak law of large numbers on the set $T_{n,\beta} = \{s \in \{0,1\}^n : |\frac{1}{n}\log\frac{1}{\mathbf{P}(s)} - h(p)| < \beta\}$:

$$\mathbf{P}(s \in T_{n,\beta}) \geq 1 - \frac{\sigma^2}{\beta^2 n}$$

We can compute the size of $T_{n,\beta}$ using the fact that $s \in T_{n,\beta} \Leftrightarrow 2^{-n(h(p)+\beta)} < \mathbf{P}(s) < 2^{-n(h(p)-\beta)}$. Thus, since the total probability that $T_{n,\beta}$ contains can't be bigger than 1, we have:

$$|T_{n,\beta}|2^{-n(h(p)+\beta)} < 1 \quad \Rightarrow \quad |T_{n,\beta}| < 2^{n(h(p)+\beta)}$$

Hence, we can take β as small as we want, for N large enough, the probability will be greater than $1-\delta$, so our first approximation was good: asymptotically, we need only $2^{nh(p)}$ strings in our bags.