Binary Additive White-Gaussian-Noise Channel

Tomáš Filler

December 11, 2009

In this handout, we give a short summary of the Binary Additive White-Gaussian-Noise Channel (abbreviated as BAWGNC). This channel is often used as a practical model in many digital communication schemes (such as transmission of data over a pair of wires). In practice, many types of noise sources are additive and independent of each other and thus, when added together, can be approximated by a zero-mean Gaussian random variable with some variance, say σ^2. This approximation is justified by the central limit theorem.

The BAWGNC(σ) channel, as depicted in Figure 1, accepts a realization of a random variable $X \in \{-1, +1\}$ on its input and outputs a realization of a random variable $Y = X + Z$, where Z is a zero-mean Gaussian random variable with variance σ^2. When combined together, we obtain the following conditional pdfs of Y

$$
\begin{aligned}
P(Y = y|X = +1) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y-1)^2}{2\sigma^2}\right] \\
P(Y = y|X = -1) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y+1)^2}{2\sigma^2}\right].
\end{aligned}
$$

This channel is fully described by the noise variance σ^2.

The only component that is necessary to have when running the Belief Propagation (BP) decoder over an *arbitrary* binary memoryless channel is the log-likelihood ratio calculated for *every* received value (variable node). For a received value y_i (y_i is binary if BSC was used, or a real number such as in BAWGNC), the log-likelihood ratio

$$LLR(y_i) = \log \frac{P(Y_i = y_i|\text{bit 0 was sent})}{P(Y_i = y_i|\text{bit 1 was sent})}$$

is used in the BP algorithm to drive the ith variable node. For the BAWGNC, the log-likelihood ratio is of the following simple form

$$LLR(y_i) = \log \frac{P(Y_i = y_i|\text{bit 0 was sent})}{P(Y_i = y_i|\text{bit 1 was sent})} = \log \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y-1)^2}{2\sigma^2}\right]}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y+1)^2}{2\sigma^2}\right]} = \log \exp\left[-\frac{(y-1)^2}{2\sigma^2} + \frac{(y+1)^2}{2\sigma^2}\right] = \frac{2}{\sigma^2} y.$$

The capacity of this channel is given by the following formula

$$C_{BAWGNC}(\sigma) = -\int_{-\infty}^{+\infty} \Phi(y, \sigma^2) \log_2 \Phi(y, \sigma^2) dy - \frac{1}{2} \log_2(2\pi e \sigma^2),$$

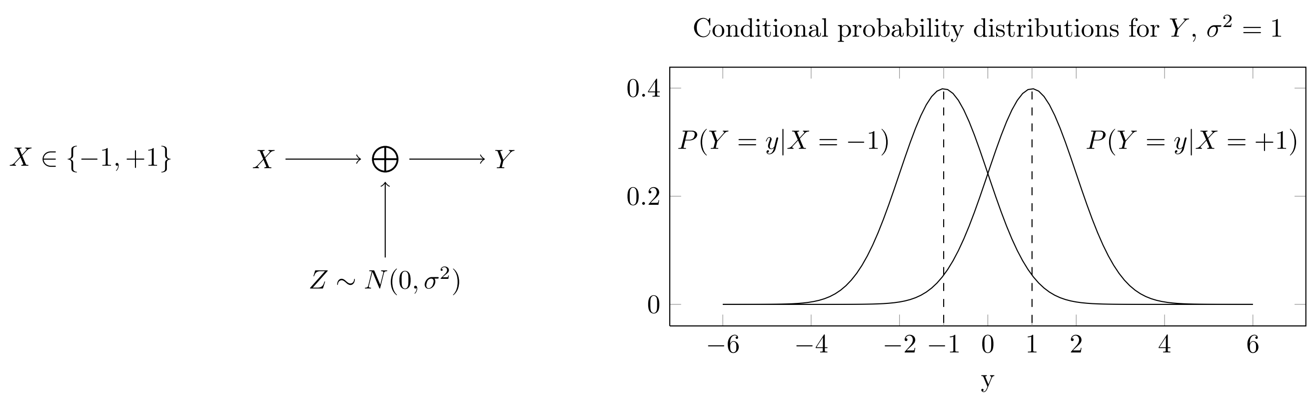

Figure 1: Binary Additive White-Gaussian-Noise Channel (BAWGNC).

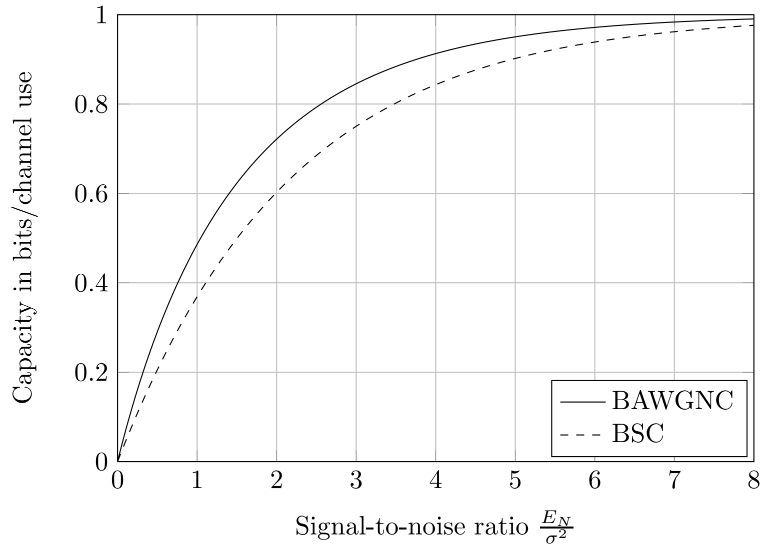

Figure 2: The capacity of the BAWGNC in bits/channel use. Capacity of the BSC(f) with $f = Q\left(\sqrt{E_N/\sigma^2}\right)$ corresponds to a hard-decision decoding, where the decoder uses just $sign(y)$ as its input (instead of the original signal y). On x-axis is the signal-to-noise ratio E_N/σ^2(*not* expressed in dB). This can be interpreted as $1/\sigma^2$, when $E_N = 1$ (this is just a scaling which is assumed for our channel).

where

$$\Phi(y,\sigma^2) = \frac{1}{\sqrt{8\pi\sigma^2}}\left(\exp\left[-\frac{(y-1)^2}{2\sigma^2}\right] + \exp\left[-\frac{(y+1)^2}{2\sigma^2}\right]\right).$$

Figure 2 shows the capacity for different values of the signal-to-noise ratio $1/\sigma^2$.

In the rest, we describe an application of the BAWGNC in digital communication and a common parametrization which is often considered. Unless we state it explicitly, we do not assume any coding in the following paragraphs.

Assume we want to send one bit over a wireless channel using an allowed frequency ω_0. The frequency ω_0 is assigned to us and we are allowed to send radio waves at this (and a slightly different) frequency. A simple way how to "modulate" one bit onto this frequency is to use the BPSK (Binary Phase-Shift Keying[1]). In this modulation scheme, we send a real signal (wave) $a_i(t)$ for a period of time $t \in [0,T)$, where $i \in \{0,1\}$ is the bit we want to send.[2] A common way how to describe the waves $a_i(t)$ is by their "basis function". Let $\Psi(t) = A\cos(\omega_0 t)$ be a real signal for $t \in [0,T)$ and choose constant A in a way that this wave is of a unit energy, i.e., $\int_0^T (\Psi(t))^2\,dt = 1$. If E_N is the energy we are allowed to use for transmitting one bit, we pick

$$a_0(t) = \sqrt{E_N}\Psi(t), \qquad\qquad a_1(t) = -\sqrt{E_N}\Psi(t).$$

In this case, the energy of $a_i(t)$, $\int_0^T (a_i(t))^2\,dt = E_N$ as desired. If another bit needs to be sent, we send the required wave again for another period of T seconds.

At the receiver side, we receive the noisy wave $r(t) = a_i(t) + n(t)$, where $n(t)$ is assumed to be an independent zero-mean Gaussian noise with variance σ^2, $t \in [0,T)$. This choice of the form of the noise can be justified in a similar fashion as in the beginning of this handout. In order to detect what wave was sent, we need to "compare" $r(t)$ with both waves $a_i(t)$, over the time interval $[0,T)$. The best possible linear detector (it minimizes the bit-error probability) is known to be the "correlator" (or "matched filter") calculating the statistics

$$D = \int_0^T r(t)\Psi(t)\,dt = \int_0^T a_i(t)\Psi(t)\,dt + \int_0^T n(t)\Psi(t)\,dt = \pm\sqrt{E_N} + \int_0^T n(t)\Psi(t)\,dt.$$

Since $n(t)$ is assumed to be a Gaussian random variable, so is $\int_0^T n(t)\Psi(t)\,dt$ and therefore the same holds for D

[1]If you had a course on digital communication, then you may find this type of modulation in your notes or textbook.

[2]Think that signal $a_i(t)$ represents voltage we apply to an antenna over the time interval. If the antenna has resistance of 1Ω, then voltage and current are the same. In this case, the dissipated power can be calculated as $\int_0^T (a_i(t))^2\,dt$.

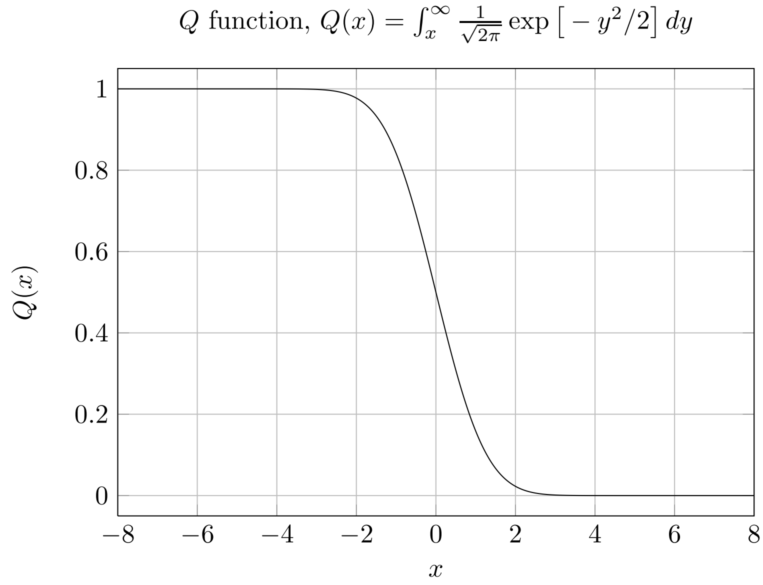

Figure 3: Q function.

when conditioned by the bit we sent. The mean and variance are calculated as follows:

$$E[D] = \pm\sqrt{E_N} + E\left[\int_0^T n(t)\Psi(t)\,dt\right] = \pm\sqrt{E_N} + \int_0^T \underbrace{E\left[n(t)\right]}_{=0}\Psi(t)\,dt = \pm\sqrt{E_N}$$

$$Var[D] = E\left[(D - E[D])^2\right] = E\left[\int_0^T n(t)\Psi(t)\,dt \cdot \int_0^T n(s)\Psi(s)\,ds\right] = E\left[\int_0^T\int_0^T n(t)n(s)\Psi(t)\Psi(s)\,dtds\right]$$

$$= \int_0^T\int_0^T \underbrace{E\left[n(t)n(s)\right]}_{\sigma^2 \text{ if } t=s \text{ and } 0 \text{ otherwise}} \Psi(t)\Psi(s)\,dtds = \sigma^2 \underbrace{\int_0^T \Psi(t)\Psi(t)\,dt}_{=1} = \sigma^2.$$

The detection statistic D conditioned by the bit we sent is thus Gaussian with mean $\pm\sqrt{E_N}$ and variance σ^2. The maximum amount of information can be sent if both bits 0 and 1 are sent equally likely. Since we do not know which bit was sent and have access only to D, the following rule minimizes the probability of error

$$\text{decode as "0" if } D > 0 \text{ and "1" otherwise.} \tag{1}$$

Under this decision rule, the probability that the bit is decoded incorrectly, P_b, can be obtained as follows

$$P_b = P(D \leq 0|\text{bit 0 was sent})P(\text{bit 0 was sent}) + P(D > 0|\text{bit 1 was sent})P(\text{bit 1 was sent}),$$

where

$$P(D \leq 0|\text{bit 0 was sent}) = \int_{-\infty}^0 \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y-1)^2}{2\sigma^2}\right] dy,$$
$$P(D > 0|\text{bit 1 was sent}) = \int_0^\infty \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(y+1)^2}{2\sigma^2}\right] dy.$$

The values of both integrals can be obtained from the tabulated Q function (Matlab command `qfunc`)

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} \exp\left[-y^2/2\right] dy,$$

when the arguments are properly scaled (Can you see how?). See Figure 3 for a plot of the Q function. By the symmetry of the Q function, we have

$$P_b = Q\left(\sqrt{E_N/\sigma^2}\right)\frac{1}{2} + Q\left(\sqrt{E_N/\sigma^2}\right)\frac{1}{2} = Q\left(\sqrt{E_N/\sigma^2}\right).$$

To summarize this example, when sending waves $a_i(t)$, the performance (without any coding so far) is driven by the signal-to-noise ratio (SNR) E_N/σ^2. By increasing the SNR (for example by increasing the energy E_N while keeping σ^2 fixed), we can decrease the probability of error P_b. Unfortunately, in order to decrease P_b to zero, we need $E_N \to \infty$, which is highly impractical. By using the decision rule (1), the communication channel is essentially a BSC(f) with flipping probability $f = Q(\sqrt{E_N/\sigma^2})$. This transformation corresponds to "hard-decision decoding", where we only allow to utilize the sign obtained from the receiver's statistic D. This approach to communicating over the BAWGNC is suboptimal (see Figure 2 for comparison). A better (and optimal way) is the "soft-decision decoding", where we allow the decoder to utilize the receiver's statistic D without any restriction. Both approaches can be realized by LDPC codes when decoded by the BP algorithm initialized with suitable log-likelihood ratios.

Different measures of signal-to-noise ratio

The following approach is often used for comparing coding schemes with different rates over the BAWGNC. Let the channel be of SNR E_N/σ^2, where E_N is the energy we use for sending one bit over the channel. If we do not use any coding method (the rate of the communication is $R = 1$), then we can achieve probability of bit error $P_b = Q(\sqrt{E_N/\sigma^2})$. Now, assume that we are allowed to use a code of rate R. In this case, we send $1/R$ encoded bits for every *information bit* over the channel. The energy used for sending *one information bit* is thus $E_b = E_N/R$. Now the question is what energy to use for comparing coding schemes of different rates?

If we fix the energy E_N, then by a code of rate $R < 1$ we effectively *increase* the power E_b we use for sending one information bit and thus this will not be a fair comparison across different rates R. For this reason, E_b is often used because it makes more sense to fix the energy budget we have for every information bit. Finally, the variance of a Gaussian noise is often being expressed in terms of its power $N_0 = 2\sigma^2$ (parameter N_0 is called a single-sided power-spectral density) and thus obtaining a new (normalized) SNR measure E_b/N_0. This SNR measure is usually given in decibels (dB), $10\log_{10}(E_b/N_0)$ and is widely used.

Consider the problem of channel simulation when E_b/N_0 is given in dBs. First, in our channel, we have $E_N = 1$. Combining all the above equations together, we obtain the variance σ^2 which we should use to simulate the channel

$$\sigma^2 = \frac{1}{2R10^{\gamma/10}},$$

where γ is the SNR, E_b/N_0, expressed in dB, and R is the rate of the code we are using.

The relationship between E_b and E_N can be described by the following diagram.