lecture01

December 18, 2024

0.0.1 Entrory (in bits - amount of information)

[4]:
```
p  = 0.99; @show -log2(p );
p  = 0.01; @show -log2(p );
```

```
-(log2(p )) = 0.014499569695115089
-(log2(p )) = 6.643856189774724
```

0.0.2 Joint Entropy

$H(X,Y) = -E[log\, p(X,Y)] = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}$

0.0.3 Proof of DPI using Marko Chains

$P(X,Z|Y) \overset{\text{(br)}}{=} \frac{P(X,Y,Z)}{P(Y)} \overset{\text{(br)}}{=} \frac{P(Z|Y,X)\,P(X,Y)}{P(Y)} \overset{\text{(mc+br)}}{=} P(Z|Y)P(X|Y)$

Rules Coding

- (br) => Bayes Rules
- (cr) => Chain Rules
- (cre) => Conditioning Reduces Entropy
- (maxH) => Maximum Entropy

$E[\mathcal{L}(\xi(X^n))]$

0.0.4 Shannon's Coding Theorem :

Proof:

- Let us build a random code

- We generate \$ M \$ code words \$ X^n(1), X^n(2), …, X^n(M) \$.
  > According to \$ P(X^n) = _{i=1}^n P(X_i), ; X^n = (X_1, X_2, …, X_n) \$

- Assume that message \$ i \$ is transmitted (\$ X^n(i) \$ is transmitted).

- Decoder receives \$ Y^n \$.

- Decoder finds message \$ i \$ such that \$ (X^n(i), Y^n) \$ is jointly typical.
  > Let's see what joint typicality (J.T.) means.

[2]:

```julia
using Random, Statistics, Distributions

# Function to generate random codewords
function generate_codewords(M, n, p_dist)
    # Generate M codewords, each of length n, using distribution p_dist
    return [rand(p_dist, n) for _ in 1:M]
end

# Function to simulate channel (add noise to transmitted codeword)
function simulate_channel(Xn, noise_dist)
    return Xn .+ rand(noise_dist, length(Xn))
end

# Joint typicality check (simplified for i.i.d. Gaussian noise)
function is_jointly_typical(Xn, Yn, threshold)
    n = length(Xn)
    joint_average = mean(Xn .* Yn)
    return abs(joint_average - mean(Xn) * mean(Yn)) < threshold
end

# Decoder: Find the message index based on joint typicality
function decode(Yn, codebook, threshold)
    for (i, Xn) in enumerate(codebook)
        if is_jointly_typical(Xn, Yn, threshold)
            return i  # Return the index of the jointly typical codeword
        end
    end
    return nothing  # If no codeword is jointly typical, return nothing
end

# Parameters
M = 4               # Number of codewords
n = 10              # Length of each codeword
p_dist = Bernoulli(0.5)  # Probability distribution for codewords
noise_dist = Normal(0, 0.5)  # Noise distribution (Gaussian)
threshold = 0.1     # Threshold for joint typicality

# Generate random codewords
codebook = generate_codewords(M, n, p_dist)

# Transmit message i
i_transmitted = 2
Xn_transmitted = codebook[i_transmitted]
Yn_received = simulate_channel(Xn_transmitted, noise_dist)

# Decode the received message
i_decoded = decode(Yn_received, codebook, threshold)
```

```julia
# Print results
println("Transmitted message index: ", i_transmitted)
println("Decoded message index: ", i_decoded)
```

```
Transmitted message index: 2
Decoded message index: 1
```

m = 128; n = 1100; Y = sqrt(pi) * randn(m,n)

```
[ ]:
```