Final Exam - Moed B

Total time for the exam: 3 hours!

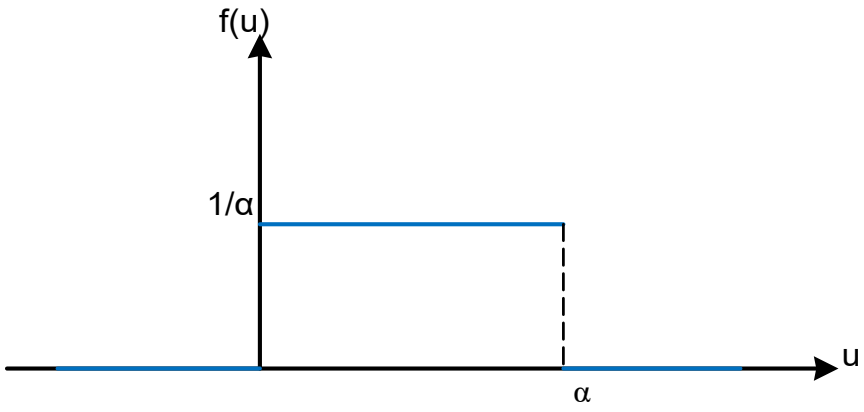
Important: For **True / False** questions, copy the statement to your notebook and write clearly true or false. You should prove the statement if true, and provide counterexample otherwise.

1) **Entropy and differential entropy of continuous random variable (28 Points)**: Let U be a uniform random variable on the interval $[0, \alpha]$, and let f_U be its density.

   a) (4 points) Draw the density f_U.
   b) (4 points) Differential entropy of U is defined as
   $$h(U) = \int_{-\infty}^{+\infty} -f_U(u) \log f_U(u) du.$$
   Compute the differential entropy of U.
   c) (4 points) The quantized version of U is given by X_N, where N is the number of the quantizations. Specifically, define $\Delta = \frac{\alpha}{N}$, and then
   $$X_N = i, \quad \text{if} \quad i\Delta \leq U < (i+1)\Delta.$$
   What is the PMF (probability mass function) of X_N, draw it and compute the entropy $H(X_N)$ as a function of N.
   d) (4 points) What is $\lim_{N\to\infty} H(X_N)$.
   e) (4 points) **True or False:** The entropy of a continuous random variable is infinite. (Explain your answer.)
   f) (4 points) What is $\lim_{N\to\infty}(H(X_N) + \log \Delta)$.
   g) (4 points) Suggest an algorithm to estimate differential entropy using entropy.

Solution Q1:

a)
$$f(u) = \begin{cases} \frac{1}{\alpha}, & 0 \leq u \leq \alpha \\ 0, & \text{otherwise} \end{cases}$$

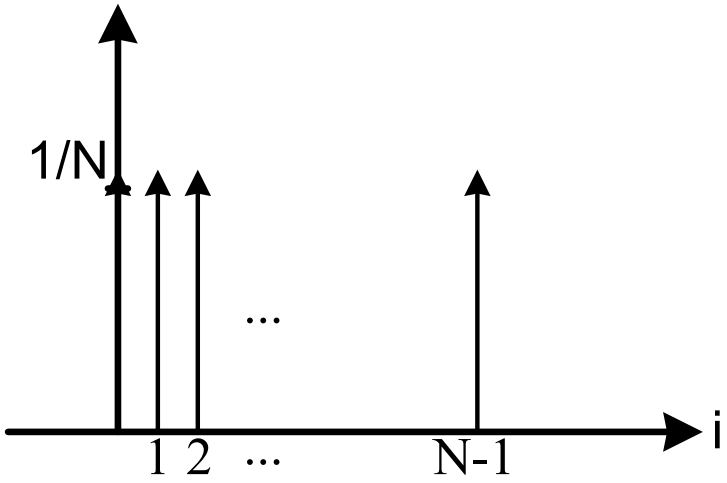


b) Consider the following
$$\begin{aligned} h(U) &= \int_{-\infty}^{+\infty} -f_U(u) \log f_U(u) du \\ &= \int_{0}^{+\alpha} -\frac{1}{\alpha} \log\left(\frac{1}{\alpha}\right) du \\ &= \log(\alpha). \end{aligned}$$

c) The PMF of X_N is determined by

$$\begin{aligned}\Pr[X_N = i] &= \Pr[i\Delta \le U < (i+1)\Delta] \\ &= \int_{i\Delta}^{(i+1)\Delta} f_U(u)du \\ &= \int_{i\Delta}^{(i+1)\Delta} \frac{1}{\alpha}du \\ &= \frac{\Delta}{\alpha} \\ &= \frac{1}{N}.\end{aligned}$$

Thus, X_N has a discrete uniform distribution on $\{0, \ldots, N-1\}$, that can be described as follows:



Obviously, the entropy of a uniform random variable is:

$$\begin{aligned}H(X_N) &= \log|\mathcal{X}_N| \\ &= \log(N).\end{aligned}$$

d) From the above item, $\lim_{N\to\infty} H(X_N) = \lim_{N\to\infty} \log N = \infty$.

e) **True**. The number of bits that can be stored on a continuous random variable is infinity.
More precisely, assume that X has a continuous density function $f(x)$. We quantize the support of X into bins of size Δ. By the mean value theorem, we know that there exists x_i in bin i such that

$$\begin{aligned}P(X = i) &= \int_{i\Delta}^{(i+1)\Delta} f(x)dx \\ &= \Delta x_i.\end{aligned}$$

Then, the entropy of X_Δ is:

$$\begin{aligned}H(X_\Delta) &= -\sum_i (\Delta x_i)\log(\Delta x_i) \\ &= -\log(\Delta) - \sum_i (\Delta x_i)\log(x_i),\end{aligned}$$

where the second equality follows from the PMF definition. The entropy goes to infinity as $\Delta \to 0$. Note that the remaining sum is a Riemann sum that tends to the differential entropy as $\Delta \to 0$ (assuming the limit exists).

f)

$$\begin{aligned}\lim_{N\to\infty}(H(X_N) + \log\Delta) &= \lim_{N\to\infty}\left[\log(N) + \log\left(\frac{\alpha}{N}\right)\right] \\ &= \lim_{N\to\infty}(\log\alpha) \\ &= h(U).\end{aligned}$$

g) From part (f) we can see that for large enough N,

$$h(U) \approx H(X_N) + \log(\Delta)$$

Thus, a possible algorithm is quantize uniformly the support of the continuous random variable to bins of size Δ, calculate the entropy of the quantized random variable. Since it will blow up, we should add a correcting factor of $\log(\Delta)$.

2) **Time - varying BSC (24 Points)** Consider a time-varying memoryless BSC: the probability for flipping a bit at time i is equal to p_i for $i = 1, \ldots, n$ as described in Fig. 1. The channel is memoryless and without feedback as assumed in class.

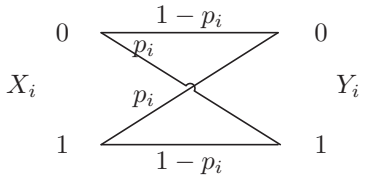


Fig. 1: Time-varying BSC.

a) (6 points) Calculate the capacity that is given by $\max_{p(x_1,\ldots,x_n)} \frac{1}{n} I(X^n; Y^n)$.
b) (6 points) Consider the time-invariant version of this channel with a normalized average parameter, i.e. $\bar{p} = \frac{1}{n}\sum_{i=1}^{n} p_i$. Calculate the capacity that is given by $\max_{p(x_1,\ldots,x_n)} \frac{1}{n} I(X^n; Y^n)$.
c) (6 points) **True or False:** For a fixed $p(x)$ mutual information $I(X;Y)$ is convex in $p(y|x)$.
d) (6 points) Is the time-varying or the time-invariant version has a greater capacity? Prove your answer.

Solution:

a) We will first show an upper bound, and then proceed to show the argument that achieves this upper bound. Consider the following upper bound:

$$\begin{aligned}\max_{p(x_1,\ldots,x_n)} \frac{1}{n} I(X^n; Y^n) &= \max_{p(x_1,\ldots,x_n)} \frac{1}{n} H(Y^n) - H(Y^n|X^n) \\ &\overset{(a)}{\leq} \max_{p(x_1,\ldots,x_n)} \frac{1}{n}\sum_i H(Y_i) - H_2(p_i) \\ &\overset{(b)}{\leq} \frac{1}{n}\sum_i (1 - H_2(p_i)),\end{aligned}$$

where (a) follows from the fact that conditioning reduces entropy and the channel law, and (b) follows from $H(Y_i) \leq 1$ and that the expression does not depend on the maximization.
computing the expression with $P(x^n)$ that is i.i.d. distributed as Bern(0.5) achieves the upper bound.

b) This is a memoryless channel whose average mutual information is equal to the capacity

$$\max_{p(x_1,\ldots,x_n)} \frac{1}{n} I(X^n; Y^n) = 1 - H_2(\bar{p}).$$

c) From Lecture 2.
d) By the convexity of the mutual information in the channel law, the capacity of the time-varying channel is greater (more be precisely, if $I(X;Y)$ is convex in $P(y|x)$, then $I(X^n;Y^n)$ is convex in $P(y^n|x^n)$). Another correct argument is the concavity of the entropy function.

3) **Neural network initialization (23 Points)**: Consider a neural network with L layers. The first layer is the input, denoted by $\mathbf{x}$, and the last layer is the output. The network sizes are $[N_1, N_2, .., N_L]$. All activation functions are the sigmoid function. The initial weights are drawn i.i.d. from some distribution that has mean (expected value) μ_w, and variance σ_w^2.
Recall that we defined z_i^l as the input to the sigmoid of the i'th neuron in the l'th layer. The question is mostly on z_i^2 which is a linear function of the input $\mathbf{x}$.

a) (6 Points) Assume input vector of N_1 ones, i.e. $\mathbf{x} = [\underbrace{1, 1, .., 1}_{N_1}]$. Calculate the mean and the variance of z_i^2 for $i = 1, 2, .., N_2$. The answer should be a function of μ_w, σ_w.
b) (10 Points) What are the best values for the mean and variance of the activation inputs z_i^2 for $i = 1, 2, .., N_2$, explain your answer.
c) (7 Points) A student suggested to initialize the weights with $\mathcal{N}(0, \frac{1}{N_1})$. Repeat question a), and check if it satisfies the values you suggested in b).

Solution Q3:

a) using the fact that $\mathbf{x} = [\underbrace{1, 1, .., 1}_{N_1}]$:

$$\mathbb{E}\left[z_i^2\right] = \mathbb{E}\left[\sum_{i=1}^{N_1} w_i x_i\right] \overset{(a)}{=} \sum_{i=1}^{N_1} \mathbb{E}[w_i] = N_1 \mu_w.$$

$$\text{Var}[z_i^2] = \text{Var}\left[\sum_{i=1}^{N_1} w_i x_i\right] \overset{(b)}{=} \sum_{i=1}^{N_1} \text{Var}[w_i] = N_1 \sigma_w^2.$$

Where in (a) we used the linearity of the expectation, and in (b) we used the fact that the weights are drawn i.i.d.

b) The best values for the mean and variance of the activation inputs z_i^2 will be the ones that achieve zero mean and a small variance. For example - $\mathcal{N}(0,1)$. The reason that we would want to have activation values with zero mean and small variance while using the Sigmoid function is to avoid saturation. Namely, to keep $\sigma(z_i^l)$ in the linear region so that $\sigma'(z_i^l)$ values aren't too small, which could cause what is called the "Vanishing Gradient". The "Vanishing Gradient" can ruin the learning process, and we might get stuck in saturation, hence fail to learn.

Additional explanation: Let's assume we're using some cost function (for example the MSE cost function, or the Cross-Entropy cost function). As shown in class:

$$\delta_j^l = \sum_k w_{kj}^{l+1} \delta_k^{l+1} \sigma'(z_j^l).$$

$$\frac{\partial C}{\partial w_{jk}^l} = \delta_j^l a_k^{l-1}.$$

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l.$$

If our activations, $\sigma(z_j^l)$, aren't in the linear region, the term $\sigma'(z_j^l)$ will be small, resulting in a small error, δ_j^l. This is a phenomenon that gets worse the more we go deeper into our network during Backpropagation. The result will be a very small gradient which will cause the model to stick to its current parameters and not learn well.

To emphasize this important matter, consider this graph:

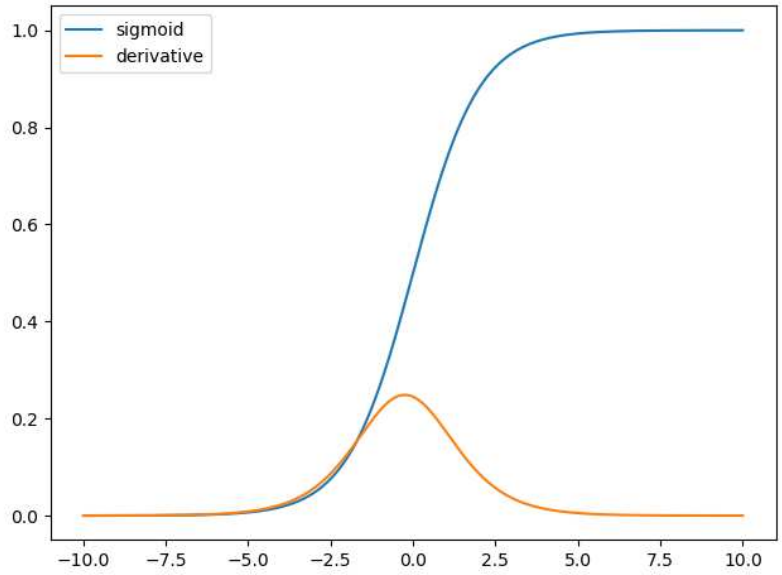


Fig. 2: Sigmoid and Sigmoid Prime

c) Repeating part (a) we achieve:

$$\mathbb{E}[z_i^2] = 0.$$

$$\text{Var}[z_i^2] = 1.$$

Which satisfies (b).

4) **Linear regression (30 Points)**: Given training set $\{(x_i, y_i)\}_{i=1}^{N}$ where $(x_i, y_i) \in \mathbb{R}^2$.
First, we use a linear regression method to model this data, assume the model has no bias, i.e. b=0. To test our linear regressor, we choose at random some data records to be a training set, and choose at random some of the remaining records to be a test set. Now let us increase the training set size gradually.
   a) (7 points) As the training set size increases, what do you expect will happen with the mean training error? explain your answer.
   b) (7 points) As the training set size increases, what do you expect will happen with the mean test error? explain your answer.

Now we have prior knowledge of our dataset's distribution $y_i \sim \mathcal{N}(\log(wx_i), 1)$.
   c) (10 points) We now perform a maximum likelihood estimation of w. Which of the following conditions is sufficient and necessary for a maximum likelihood estimation of w:
      i) $\sum_i x_i \log(wx_i) = \sum_i x_i y_i \log(wx_i)$
      ii) $\sum_i x_i y_i = \sum_i x_i y_i \log(wx_i)$
      iii) $\sum_i x_i y_i = \sum_i x_i \log(wx_i)$
      iv) $\sum_i y_i = \sum_i \log(wx_i)$
   d) (6 points) Provide a pseudocode (or a matlab code) for estimating w.

Solution Q4:

a) Using the same linear regression model with more training data will result in the following: The model will minimize the MSE cost function with respect to a larger amount of samples, resulting in a higher training error mean value.

b) More training data increases the likeliness that the training set will 'represent' the test set, resulting in a decreased mean test error.

c) We could write the log likelihood as:

$$LL = \log\left(\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(y_i-\log(wx_i))^2}{2\sigma^2}\right)\right) = \log\left(\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(y_i-\log(wx_i))^2}{2}\right)\right) =$$

$$\sum_{i=1}^{n}\log\left(\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(y_i-\log(wx_i))^2}{2}\right)\right).$$

Now derive the expression with respect to w:

$$\frac{\partial LL}{\partial w} = 0 \Longrightarrow \frac{\partial\sum_{i=1}^{n}(y_i-\log(wx_i))^2}{\partial w} = 0$$

$$\sum_{i=1}^{n}\frac{x_i}{wx_i}(y_i-\log(wx_i)) = 0 \Longrightarrow \sum_{i=1}^{n}y_i = \sum_{i=1}^{n}\log(wx_i).$$

d) First, we'll expand the term from (c) in order to isolate w. Then we'll provide the algorithm:

$$\sum_{i=1}^{n}y_i = \sum_{i=1}^{n}\log(wx_i) = n\log(w) + \sum_{i=1}^{n}\log(x_i)$$

$$n\log(w) = \sum_{i=1}^{n}(y_i-\log(x_i))$$

$$w = \exp\left(\frac{1}{n}\sum_{i=1}^{n}(y_i-\log(x_i))\right).$$

Algorithm 1 Maximum likelihood estimator for w

```
function LIKELIHOOD ESTIMATOR(X,Y)
    n ← Size(X)
    Sum ← 0
    for i = 1, ..., n do
        Sum ← Sum + (y_i − log(x_i))
    Sum ← (1/n) Sum
    w = exp(Sum)
return w
```

Good Luck!