Introduction to Information Theory and Machine Learning July 7th, 2022.
(Prof. Permuter Haim, Mr. Bashar Huleihel, and Mr. Omer Luxembourg)

Final Exam - Moed A
Total time for the exam: 3 hours!

Please copy the following sentence and sign it: " I am respecting the rules of the exam: Signature:_________ "

Important: For **True / False** questions, copy the statement to your notebook and write clearly true or false. You should prove the statement if true, or disprove it, e.g. by providing a counter-example, otherwise.

1) **Uncertainty about true distribution (24 Points):** Consider a source U with alphabet $\mathcal{U} = \{a_1, \dots a_m\}$ and suppose we know that the true distribution of U is either P_1 or P_2, but we are not sure which.
   a) (8 points) **True/False:** There is a prefix code where the length of the codeword associated to a_i is $l_i = \left\lceil \log_2\left(\frac{2}{P_1(a_i)+P_2(a_i)}\right)\right\rceil$.
   Solution:
   Let $l_i = \left\lceil \log_2\left(\frac{2}{P_1(a_i)+P_2(a_i)}\right)\right\rceil$, and compute the Kraft sum:
$$\begin{aligned}\sum_{m=1}^{M} 2^{-l_i} &\leq \sum_{m=1}^{M} 2^{-\log_2\left(\frac{2}{P_1(a_i)+P_2(a_i)}\right)} \\ &= \sum_{m=1}^{M} \frac{P_1(a_i)+P_2(a_i)}{2} \\ &= 1.\end{aligned}$$
   Accordingly, the Kraft sum at most to 1, and therefore, there exists a prefix-free code where the length of the codeword associated to a_i is l_i.
   b) (8 points) Show that the average (computed using the true distribution) length $\bar{l}$ of the code constructed in item (a) satisfies $H(U) \leq \bar{l} \leq H(U) + 2$.
   Solution:
   Since the constructed code in item (a) is a prefix code, then $l \geq H(U)$. To prove the upper bound, let P^* be the true distribution (which is either P_1 or P_2). Then, clearly, the inequality $P^*(a_i) \leq P_1(a_i) + P_2(a_i)$ holds for all $1 \leq i \leq M$. Accordingly, we get:
$$\begin{aligned}\bar{l} &= \sum_{m=1}^{M} P^*(a_i) l_i \\ &= \sum_{m=1}^{M} P^*(a_i)\left\lceil \log_2\left(\frac{2}{P_1(a_i)+P_2(a_i)}\right)\right\rceil \\ &= \sum_{m=1}^{M} P^*(a_i)\left(1+\log_2\left(\frac{2}{P_1(a_i)+P_2(a_i)}\right)\right) \\ &= 2 + \sum_{m=1}^{M} P^*(a_i)\log_2\left(\frac{1}{P_1(a_i)+P_2(a_i)}\right) \\ &\leq 2 + \sum_{m=1}^{M} P^*(a_i)\log_2\left(\frac{1}{P^*(a_i)}\right) \\ &= 2 + H(U).\end{aligned}$$
   c) (8 points) Now assume that the true distribution of U is one of k distributions $P_1, \dots, P_k$, but we don't know which. Show that there exists a prefix code satisfying $H(U) \leq \bar{l} \leq H(U) + \log_2(k) + 1$.
   Solution:
   Now let $l_i = \left\lceil \log_2\left(\frac{k}{P_1(a_i)+\cdots+P_k(a_i)}\right)\right\rceil$, and let us compute the Kraft sum for this scenario:
$$\begin{aligned}\sum_{m=1}^{M} 2^{-l_i} &\leq \sum_{m=1}^{M} 2^{-\log_2\left(\frac{k}{P_1(a_i)+\cdots+P_k(a_i)}\right)} \\ &= \sum_{m=1}^{M} \frac{P_1(a_i)+\cdots+P_k(a_i)}{k} \\ &= 1.\end{aligned}$$

Thus, the code is a prefix code which implies that $\bar{l} \geq H(U)$. Here too, let P^* denote the true distribution. Therefore, $P^*(a_i) \leq P_1(a_i) + \cdots + P_k(a_i)$ for all $1 \leq i \leq M$. Following the same proof steps as for the previous item, it is easy to show that:

$$\bar{l} \leq 1 + \log_2(k) + H(U).$$

2) **GMM (18 points):** We will derive the EM update rules for a univariate Gaussian Mixture Model with two mixture components. The mean μ will be shared between the two mixture components, but each component will have its own standard deviation σ_k. The model will be defined as follows:

$$z \sim Bernoulli(\theta),$$
$$p(x|z=k) \text{ is } \mathcal{N}(\mu, \sigma_k).$$

a) (4 points) Write the density defined by this model (i.e. the probability of x, with z marginalized out)

Solution:

$$p(x) = \theta\mathcal{N}(x;\mu,\sigma_1) + (1-\theta)\mathcal{N}(x;\mu,\sigma_0)$$

b) (4 points) E-step - Compute the posterior probability $w^{(i)} = Pr(z^{(i)} = 1|x^{(i)})$

Solution:

$$w^{(i)} = \frac{\theta\mathcal{N}(x;\mu,\sigma_1)}{\theta\mathcal{N}(x;\mu,\sigma_1) + (1-\theta)\mathcal{N}(x;\mu,\sigma_0)}$$

c) (5 points) M-Step - Calculate the update rule for μ (for a fixed σ_k)
d) (5 points) M-Step - Calculate the update rule for σ_k (for a fixed μ)

Solution:

At each M-step we optimize the following:

$$\begin{aligned}
\mathcal{L}(\mu,\sigma_0,\sigma_1,\theta) &= \sum_{i=1}^{N} w^{(i)} log\left(\mathcal{N}(x(i)|\mu,\sigma_1)\right) + w^{(i)} log\theta \\
&+ (1-w^{(i)}) log\left(\mathcal{N}(x(i)|\mu,\sigma_0)\right) + (1-w^{(i)}) log(1-\theta) \\
\frac{\partial\mathcal{L}}{\partial\mu} = 0 &\Rightarrow \sum_{i}^{N} (w^{(i)}\frac{x^{(i)}-\mu}{\sigma_1^2} + (1-w^{(i)})\frac{x^{(i)}-\mu}{\sigma_0^2} = 0 \\
&\Rightarrow \sum_{i}^{N} (x^{(i)}-\mu)\left(\frac{w^{(i)}}{\sigma_1^2} + \frac{1-w^{(i)}}{\sigma_0^2}\right) = 0 \\
&\Rightarrow \sum_{i}^{N} (x^{(i)}-\mu)\left(\sigma_0^2 w^{(i)} + \sigma_1^2(1-w^{(i)})\right) = 0
\end{aligned}$$

Thus you get:

$$\begin{aligned}
\mu &= \frac{\sum_i^N x^{(i)}\left(\sigma_0^2 w^{(i)} + \sigma_1^2(1-w^{(i)})\right)}{\sum_i^N \left(\sigma_0^2 w^{(i)} + \sigma_1^2(1-w^{(i)})\right)} \\
\frac{\partial\mathcal{L}}{\sigma_k^2} = 0 &\Rightarrow \sigma_k^2 = \frac{\sum_{i=1}^N w^{(i)}(x^{(i)}-\mu)^2}{\sum_{i=1}^N w^{(i)}}
\end{aligned}$$

3) **Linear Regression (26 Points):** You are tasked with solving a fitting a linear regression model on a set of m datapoints where each feature has some dimensionality d. Your dataset can be described as the set $\{x^{(i)}, y^{(i)}\}_{i=1}^m$, where $x^{(i)} \in \mathbb{R}^d$, $y^{(i)} \in \mathbb{R}$. You initially decide to optimize the loss objective:

$$J = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - x^{(i)^T}\theta)^2,$$

using Batch Gradient Descent - in which each step involves calculations over the **entire** training set. Here, $\theta \in \mathbb{R}^d$ is your weight vector. Assume you are ignoring a bias term for this problem.

a) (4 points) Write each update of the batch gradient descent, $\frac{\partial J}{\partial \theta}$ in **vectorized** form. Your solution should be a single vector (no summation terms) in terms of the matrix X and vectors Y and θ, where

$$X = \begin{bmatrix} {x^{(1)}}^T \\ \vdots \\ {x^{(m)}}^T \end{bmatrix}, Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}.$$

Solution:
Final solution is:

$$\frac{\partial J}{\partial \theta} = \frac{2}{m} X^T(X\theta - Y)$$

Two common approaches are "derive, then vectorize" and "vectorize, then derive". Both get full credit. With the first approach.

$$\frac{\partial J}{\partial \theta} = \frac{2}{m}\sum_i ({x^{(i)}}^T\theta - y^{(i)})x^{(i)} \text{ - derivative step}$$

$$= \frac{\partial J}{\partial \theta} = \frac{2}{m} X^T(X\theta - Y) \text{ - vectorization step}$$

b) (7 points) A coworker suggests you augment your dataset by adding Gaussian noise to your features. Specifically, you would be adding *zero-mean*, Gaussian noise of *known vairance* σ^2 from the distribution

$$\mathcal{N}(0, \sigma^2 I),$$

where $I \in \mathbb{R}^{d\times d}$, $\sigma \in \mathbb{R}$. This modifies your original objective to:

$$J_* = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - (x^{(i)} - \delta^{(i)})^T\theta)^2,$$

where $\delta^{(i)}$ are **i.i.d.** noise vectors, $\delta^{(i)} \in \mathbb{R}^d$ and $\delta^{(i)} \sim \mathcal{N}(0, \sigma^2 I)$.
Express the expectation of the modified objective J_* over the Gaussian noise, $\mathbb{E}_{\delta\sim\mathcal{N}}[J_*]$, as a function of the original objective J added to a term independent of your data. Your answer should be in the form

$$\mathbb{E}_{\delta\sim\mathcal{N}}[J_*] = J + C,$$

where C is independent of points in $\{x^{(i)}, y^{(i)}\}_{i=1}^m$.
Hint: For a Gaussian random vector δ with zero mean, and convariance matrix $\sigma^2 I$

$$\mathbb{E}_{\delta\sim\mathcal{N}}[\delta\delta^T] = \sigma^2 I, \quad \mathbb{E}_{\delta\sim\mathcal{N}}[\delta] = 0.$$

Solution:

$$\begin{aligned}
J_* &= \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - (x^{(i)} - \delta^{(i)})^T\theta)^2 \\
&= \frac{1}{m}\sum_{i=1}^{m}((y^{(i)} - x^{(i)}) - \delta^{(i)})^T\theta)^2 \\
&= \frac{1}{m}\sum_{i=1}^{m}((y^{(i)} - x^{(i)})^2 - 2(y^{(i)} - x^{(i)})(\delta^{(i)T}\theta) + (\delta^{(i)T}\theta)^2) \\
&= J + \frac{1}{m}\sum_{i=1}^{m}(-2(y^{(i)} - x^{(i)})(\delta^{(i)T}\theta) + (\delta^{(i)T}\theta)^2)
\end{aligned}$$

$$\mathbb{E}_{\delta\sim\mathcal{N}}[J_*] = J + \mathbb{E}_{\delta\sim\mathcal{N}}\left[\frac{1}{m}\sum_{i=1}^{m}(-2(y^{(i)} - x^{(i)})(\delta^{(i)T}\theta) + (\delta^{(i)T}\theta)^2)\right]$$

From Linearity of Expectation, we can take the expectation individually for each sample:

$$\mathbb{E}_{\delta\sim\mathcal{N}}\left[\frac{1}{m}\sum_{i=1}^{m} -2(y^{(i)} - x^{(i)})(\delta^{(i)T}\theta)\right] = -2(y^{(i)} - x^{(i)})\mathbb{E}\left[\delta^{(i)T}\theta\right] = 0$$

$$\text{and} \quad \mathbb{E}_{\delta\sim\mathcal{N}}\left[(\delta^{(i)T}\theta)^2\right] = \sigma^2\|\theta\|_2^2 \quad \text{(from the hint)}$$

Thus, we get:

$$\mathbb{E}_{\delta\sim\mathcal{N}}[J_*] = J + \sigma^2\|\theta\|_2^2. \tag{1}$$

c) (4 points) What effect would adding noise have on model overfitting/underfitting? Explain why.
Remember that the weights update rule is derived from the loss function, which is the expectation of J_*.
Solution:
Adding noise to the model will prevent overfiting because the model wouldn't be able to remember a specific point mapping from $x^{(i)}$ to $y^{(i)}$ due to the noise inserted to $x^{(i)}$. Alternatively, in expectation, the new objective would help regularize the model, due to the similar $L2$ regularization term (see the answer for next question) and it is known that $L2$ regularization method prevents overfiting.

d) (4 points) Is this method similar to a regularization method we studied in class? If so, specify the regularization method and prove it and if not, explain why?
Solution:
Yes. It is similar to L_2 regularization, but with a scalar multiplicative σ^2 (see equation 1) which is the λ in $L2$ regularization method.

e) (3 points) Consider the limits $\sigma \to 0$ and $\sigma \to \infty$. What impact would these extremes in the value of σ have on model training (relative to no noise added)? Explain why.
Solution:
$\sigma \to 0$: Less regularizing/no effect.
$\sigma \to \infty$: All weights get pushed to zero / model underfits.

f) (4 points) Suggest a cost function and a noise that is related to *Dropout*.
Solution:

$$J_* = \frac{1}{m}\sum_{i=1}^{m}(y^{(i)} - (x^{(i)} \cdot \gamma^{(i)})^T\theta)^2$$

where $\gamma^{(i)} \in \mathbb{R}^d$ are i.i.d. noise vectors. $\gamma_j^{(i)} \sim Ber(p)$ (i.i.d.) for $j \in \{1, 2, ..., d\}$, and the probability $1-p$ is the rate of the dropout.

4) **Computable lower bounds (32 Points):** In this question, you will prove a simple lower bound on the capacity of a memoryless channel. Let $p(y|x)$ be a memoryless channel, and let $p(x)$ be a distribution on $\mathcal{X}$. Let $r(x|y)$ be an arbitrary conditional distribution on $\mathcal{X}$ given $\mathcal{Y}$, i.e., for each $x \in \mathcal{X}$ and each $y \in \mathcal{Y}$, $r(x|y) \geq 0$ and $\sum_{\tilde{x}\in\mathcal{X}} r(\tilde{x}|y) = 1$. Define the functional $F(p, r)$ as follows:

$$F(p, r) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y|x)\log_2\left(\frac{r(x|y)}{p(x)}\right).$$

where p in $F(p, r)$ denotes $P(x)$ and $p(y|x)$ is fixed thought the question. Now, for each input distribution p on $\mathcal{X}$, define the conditional distribution r_p as

$$r_p(x|y) = \frac{p(x)p(y|x)}{\sum_{\tilde{x}\in\mathcal{X}} p(\tilde{x})p(y|\tilde{x})}.$$

That is, r_p is the "true" conditional distribution of $\mathcal{X}$ given $\mathcal{Y}$ when p is the input distribution.

a) (8 points) **True/False:** For all conditional distributions r we have $F(p, r) \leq F(p, r_p)$.
Solution:
True. Let us show that the difference is non-negative.

$$\begin{aligned}
F(p, r_p) - F(p, r) &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y|x)\log_2\left(\frac{r_p(x|y)}{r(x|y)}\right) \\
&= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y|x)\log_2\left(\frac{p(x)p(y|x)}{r(x|y)\sum_{\tilde{x}\in\mathcal{X}} p(\tilde{x})p(y|\tilde{x})}\right) \\
&= D(P_1\|P_2) \\
&\geq 0,
\end{aligned}$$

where $P_1(x, y) = p(x)p(y|x)$ and $P_2(x, y) = r(x|y)\sum_{\tilde{x}\in\mathcal{X}} p(\tilde{x})p(y|\tilde{x})$.

b) (4 points) Show that $I(X; Y) = \max_r F(p, r)$.
Solution:

From the previous item we can deduce that $F(p, r_p) = \max_r F(p, r)$. Further,

$$\begin{aligned} F(p, r_p) &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x)p(y|x) \log_2 \left(\frac{r_p(x|y)}{p(x)} \right) \\ &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x)p(y|x) \log_2 \left(\frac{p(x,y)}{p(x)p(y)} \right) \\ &= I(X;Y), \end{aligned}$$

as required.

c) (8 points) **True/False:** The functional $F(p, r)$ is strictly concave in both p and r.
Solution:
True. We can rewrite $F(p, r)$ as follows:

$$F(p, r) = \left(\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x)p(y|x) \log_2(r(x|y)) \right) + \left(\sum_{x \in \mathcal{X}} p(x) \log_2 \left(\frac{1}{p(x)} \right) \right).$$

The first term is linear in p while the second term is strictly concave in p (the function $t \Rightarrow t \log_2 \frac{1}{t}$ is strictly concave). Therefore, $F(p, r)$ is strictly concave in p. In addition, the first term is concave in r (the function $\log_2$ is strictly concave), and the second term is constant with respect to r. Therefore, $F(p, r)$ is strictly concave in r.

d) (6 points) In Algorithm 1 below, we introduce an iterative algorithm for maximizing a two-variable function. Following the previous items, suggest such an iterative algorithm to compute the capacity.
Solution:
Following the previous items, the capacity can be computed as

$$\begin{aligned} C &= \max_p I(X;Y) && (2) \\ &= \max_p \max_r F(p, r). && (3) \end{aligned}$$

Accordingly, in the spirit of Algorithm 1, the following algorithm can be used to compute the capacity:

Algorithm 1 Alternating maximization procedure

---

input: The function $F(p, r)$ that is concave in both p and r
output: A global maximum of $F(p, r)$ (capacity)
set p_0 uniform in $\mathcal{X}$ and solve $r_0 = \arg\max_r F(p_0, r)$
set $i = 1$
while $F(p_i, r_i)$ not converged **do**
    $p_i = \arg\max_p F(p, r_{i-1})$
    $r_i = \arg\max_r F(p_{i-1}, r)$
    compute $F(p_i, r_i)$
    $i = i + 1$
end
return $F(p_i, r_i)$

Note: The Alternating maximization procedure is known to converge to optimal solution when the function $g(x, y)$ is concave in (x, y).

e) (6 points) For a given memoryless channel, let r^* denote the conditional distribution that should be used to obtain the capacity. Write explicitly r^* for the case of a binary symmetric channel with crossover probability 0.2.
Solution:
The optimal input distribution p^* for a binary symmetric channel is a uniform distribution. Accordingly,

$$\begin{aligned} r^*(x|y) &= r_{p^*}(x|y) \\ &= \frac{p^*(x)p(y|x)}{\sum_{\tilde{x} \in \mathcal{X}} p^*(\tilde{x})p(y|\tilde{x})}. \end{aligned}$$

Good Luck!