UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

CS440/ECE448 Artificial Intelligence

Conflict Exam 1

Spring 2023

Spring 2023

Your Name: ______________________________

Your NetID: ______________________________

Instructions

- Please write your name on the top of every page.
- Have your ID ready; you will need to show it when you turn in your exam.
- This will be a CLOSED BOOK, CLOSED NOTES exam. You are permitted to bring and use only one 8.5x11 page of notes, front and back, handwritten or typed in a font size comparable to handwriting.
- No electronic devices (phones, tablets, calculators, computers etc.) are allowed.
- Make sure that your answer includes only the variables that it should include, but DO NOT simplify explicit numerical expressions. For example, the answer $x = \frac{1}{1+\exp(-0.1)}$ is MUCH preferred (much easier for us to grade) than the answer $x = 0.524979$.

Possibly Useful Formulas

$$P(X=x|Y=y)P(Y=y) = P(Y=y|X=x)P(X=x)$$
$$P(X=x) = \sum_y P(X=x, Y=y)$$
$$E\left[f(X,Y)\right] = \sum_{x,y} f(x,y)P(X=x,Y=y)$$

Precision,Recall $= \frac{TP}{TP+FP}, \frac{TP}{TP+FN}$

MPE=MAP: $f(x) = \arg\max\left(\log P(Y=y) + \log P(X=x|Y=y)\right)$

Naive Bayes: $P(X=x|Y=y) \approx \prod_{i=1}^{n} P(W=w_i|Y=y)$

Laplace Smoothing: $P(W=w_i) = \frac{k + \text{Count}(W=w_i)}{k + \sum_v (k + \text{Count}(W=v))}$

Fairness: $P(Y|A) = \frac{P(Y|\hat{Y},A)P(\hat{Y}|A)}{P(\hat{Y}|Y,A)}$

Linear Regression: $\varepsilon_i = f(x_i) - y_i = b + w@x_i - y_i$

Mean Squared Error: $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i^2$

Linear Classifier: $f(x) = \underset{k}{\arg\max}\, w_k@x + b$

Cross-Entropy: $\mathscr{L} = -\frac{1}{n}\sum_{i=1}^{n}\log f_{y_i}(x_i)$

Softmax: $\underset{c}{\text{soft}\max}(w@x+b) = \frac{\exp(w_c@x+b_c)}{\sum_{k=0}^{V-1}\exp(w_k@x+b_k)}$

Softmax Error: $\varepsilon_{i,c} = \begin{cases} f_c(x_i) - 1 & c = y_i \\ f_c(x_i) - 0 & \text{otherwise} \end{cases}$

Gradient Descent: $w \leftarrow w - \eta\nabla_w\mathscr{L}$

Neural Net: $h = \text{ReLU}\left(b_0 + w_0@x\right), \quad f = \text{soft}\max\left(b_1 + w_1@h\right)$

Back-Propagation: $\frac{\partial\mathscr{L}}{\partial h_j} = \sum_k \frac{\partial\mathscr{L}}{\partial f_k}\times\frac{\partial f_k}{\partial h_j}, \quad \frac{\partial\mathscr{L}}{\partial w_{0,k,j}} = \frac{\partial\mathscr{L}}{\partial h_k}\times\frac{\partial h_k}{\partial w_{0,k,j}}$

Question 1 ***(13 points)***

Ctulhoids are small slug-like animals with many eyes (they are very cute). 40% of all ctulhoids have 3 blue eyes, while 60% have 4 blue eyes. The number of orange eyes a ctulhoid has is either one less than the number of its blue eyes (with probability $1-a$) or one more than the number of its blue eyes (with probability a).

(a) (6 points) What is a ctulhoid's expected total number of eyes, including both blue eyes and orange eyes?

Solution:

$$E[\#\text{ eyes}] = 0.4(1-a)\times 5 + 0.4a\times 7 + 0.6(1-a)\times 7 + 0.6a\times 9$$

(b) (7 points) Let $A = 1$ if a ctulhoid has more orange eyes than blue eyes, and let $A = 0$ otherwise. Let $Y = 1$ if somebody adopts the ctulhoid as a pet, and let $Y = 0$ otherwise. People like orange eyes: $P(Y = 1|A = 1) = \frac{2}{3}$, but $P(Y = 1|A = 0) = \frac{1}{3}$. You have decided that this bias in favor of orange-eyed ctulhoids is unfair, so you have created an algorithm that makes pet recommendations ($\hat{Y}$) with perfect demographic parity ($P(\hat{Y} = 1|A = 1) = P(\hat{Y} = 1|A = 0) = \frac{1}{2}$), and with perfect equal opportunity ($P(\hat{Y} = 1|Y = 1, A = 1) = P(\hat{Y} = 1|Y = 1, A = 0) = p$). In terms of p, what is $P(Y = 1|\hat{Y} = 1, A = 1)$?

Solution:

$$\begin{aligned} P(Y = 1|\hat{Y} = 1, A = 1) &= \frac{P(\hat{Y} = 1|Y = 1, A = 1)P(Y = 1|A = 1)}{P(\hat{Y} = 1|A = 1)} \\ &= \frac{p\left(\frac{2}{3}\right)}{\left(\frac{1}{2}\right)} \end{aligned}$$

Question 2 ***(12 points)***

Lakes are frozen in both Milwaukee and Chicago. Lakes A, B, and C in Milwaukee have ice that is 9, 6, and 10 inches thick, respectively. Lakes D, E, and F in Chicago have ice that is 7, 4, and 2 inches thick, respectively. Lake X has ice that is 7.5 inches thick. Your goal is to design a k-nearest neighbors algorithm that guesses which city is closest to lake X ($f(x) =$ Milwaukee or $f(x) =$ Chicago).

(a) (6 points) Name two different values of K that result in different values of $f(x)$. Specify, for each, the k nearest neighbors of lake X (out of the set $\{A,B,C,D,E,F\}$).

> **Solution:** There are many possible answers. One possible answer: for $k = 1$, the nearest neighbor is D, so KNN classifies lake X as Chicago. For $k = 3$, the nearest neighbors are D, A, and B, so KNN classifies lake X as Milwaukee.

(b) (6 points) You've been given startup funds sufficient to let you measure the ice on a few hundred additional lakes. How can you can use data from these new lakes to choose a value of k that will make the k-nearest neighbors algorithm as accurate as possible, even when tested on lakes that you've never heard of?

Solution: Divide the data into a training set and a development test set. Run the following experiment for many different values of k: Use KNN, with your given training set to classify data in the development test set. Choose the values of k that gives the lowest error rate on the development test set. OPTIONAL: repeat the previous experiment for five folds. In each fold, use 4/5 of the data for training, and 1/5 for test. Choose the k with the lowest average error rate.

Question 3 ***(12 points)***

You have been comparing emoji usage among messages on the Telegram and WhatsApp messaging systems. After extensive research, your Telegram database contains m examples of the rofl emoji, n examples of the halo emoji, and no examples of any other emoji.

(a) (6 points) Use Laplace smoothing to estimate the fraction of all emojis on Telegram that are halo emojis. Note that emojis other than rofl and halo may exist, even though there are none in your training dataset. Your answer should be a function of m, n, and the Laplace smoothing hyperparameter, k.

Solution:

$$P(\text{halo}|\text{telegram}) = \frac{n+k}{m+n+3k}$$

(b) (6 points) Based on extensive research, you conclude that 87% of all text messages are sent via WhatsApp, and 13% are sent via Telegram. The likelihood of rofl on each of these two platforms is $P(X = \text{rofl}|Y = \text{whatsapp}) = p$, and $P(X = \text{rofl}|Y = \text{telegram}) = q$. A journalist shows you a text message containing a rofl emoji (and no other emojis), and asks you to guess whether it came from WhatsApp or Telegram. Under what condition should you say that it came from Telegram? Your answer should be an inequality in terms of p and q.

Solution: You should decide that the emoji is from Telegram if

$$P(Y = \text{telegram}|X = \text{rofl}) > P(Y = \text{whatsapp}|X = \text{rofl}),$$

which is true if

$$P(X = \text{rofl}|Y = \text{telegram})P(Y = \text{telegram}) > P(X = \text{rofl}|Y = \text{whatsapp})P(Y = \text{whatsapp})$$

which happens if

$$0.13q > 0.87p$$

Question 4 ***(13 points)***

You have a machine learning problem in which the input is a 3-dimensional vector, x, and the output is binary, $y \in \{0,1\}$. You are considering two possible solutions: a linear regression algorithm that uses a weight vector w and a bias term b, and a softmax linear classifier algorithm that uses weight vectors w_0 and w_1 and bias coefficients b_0 and b_1. As you know, the stochastic gradient descent algorithm has a similar form in both cases:

$$\textbf{Linear Regression:} w \leftarrow w - \eta \varepsilon_i x_i,$$
$$\textbf{Linear Classifier:} w_c \leftarrow w_c - \eta \varepsilon_{i,c} x_i,$$

where $x_i = [x_{i,0}, x_{i,1}, x_{i,2}]$ and y_i are the stochastically sampled training token, ε_i is the linear regression error term, and $\varepsilon_{i,0}, \varepsilon_{i,1}$ are the linear classifier errors.

(a) (6 points) Consider a linear regression algorithm, whose output is

$$f(x_i) = w@x_i + b$$

Suppose that w is initialized as $w = [1,1,1]$, and b is initialized as $b = 0$. In terms of $x_{i,0}$, $x_{i,1}$, $x_{i,2}$ and y_i, what is ε_i?

Solution:

$$\begin{aligned}\varepsilon_i &= w@x_i - y_i \\ &= x_{i,0} + x_{i,1} + x_{i,2} - y_i\end{aligned}$$

(b) (7 points) Consider a softmax classifier,

$$f_c(x_i) = \underset{c}{\text{softmax}}(w@x_i + b)$$

Suppose that w is initialized to $w_0 = [0,0,0]$ and $w_1 = [1,1,1]$. Suppose that b is initialized to $b_0 = 0$ and $b_1 = 0$. In terms of $x_{i,0}$, $x_{i,1}$, $x_{i,2}$, and y_i, what is $\varepsilon_{i,1}$?

Solution: The target output is: $f_c(x_i)$ should be 1 for $c = y_i$, and 0 for all other c, therefore

$$\begin{aligned}\varepsilon_{i,1} &= f_1(x_i) - y_i \\ &= \begin{cases} \frac{\exp(x_{i,0}+x_{i,1}+x_{i,2})}{1+\exp(x_{i,0}+x_{i,1}+x_{i,2})} - 1 & y_i = 1 \\ \frac{\exp(x_{i,0}+x_{i,1}+x_{i,2})}{1+\exp(x_{i,0}+x_{i,1}+x_{i,2})} & y_i = 0 \end{cases}\end{aligned}$$

This page is scratch paper