HW 6 (Electronic Component)

Q1 Maximum Likelihood Estimation

10 Points

We will begin with a short derivation. Consider a probability distribution with a domain that consists of $|X|$ different values. We get to observe N total samples from this distribution. We use n_i to represent the number of the N samples for which outcome i occurs. Our goal is to estimate the probabilities θ_i for each of the events $i = 1, 2 \ldots |X| - 1$. The probability of the last outcome, $|X|$, equals $1 - \sum_{i=1}^{|X|-1} \theta_i$.

In *maximum likelihood estimation* , we choose the θ_i that maximize the likelihood of the observed samples,

$$L(\text{samples}, \theta) \propto (1 - \theta_1 - \theta_2 - \ldots - \theta_{|X|-1})^{n_{|X|}} \prod_{i=1}^{|X|-1} \theta_i^{n_i}$$

For this derivation, it is easiest to work with the log of the likelihood. Maximizing log-likelihood also maximizes likelihood, since the quantities are related by a monotonic transformation. Taking logs we obtain

$$\theta^{\text{ML}} = \underset{\theta}{\operatorname{argmax}}\, n_{|X|} \log(1 - \theta_1 - \theta_2 - \ldots - \theta_{|X|-1}) + \sum_{i=1}^{|X|-1} n_i \log \theta_i$$

Setting derivatives with respect to θ_i equal to zero, we obtain $|X| - 1$ equations in the $|X| - 1$ unknowns, $\theta_1, \theta_2, \ldots, \theta_{|X|-1}$:

$$\frac{-n_{|X|}}{1-\theta_1^{\text{ML}}-\theta_2^{\text{ML}}-\ldots-\theta_{|X|-1}^{\text{ML}}} + \frac{n_i}{\theta_i^{\text{ML}}} = 0$$

Multiplying by $\theta_i(1 - \theta_1 - \theta_2 - \ldots - \theta_{|X|-1})$ makes the original $|X| - 1$ nonlinear equations into $|X| - 1$ linear equations:

$$-n_{|X|}\theta_i^{\text{ML}} + n_i \left(1 - \theta_1^{\text{ML}} - \theta_2^{\text{ML}} - \ldots - \theta_{|X|-1}^{\text{ML}}\right) = 0$$

That is, the maximum likelihood estimate of θ can be found by solving a linear system of $|X| - 1$ equations in $|X| - 1$ unknowns. Doing so

shows that the maximum likelihood estimate corresponds to simply the count for each outcome divided by the total number of samples. I.e., we have that:

$$\theta_i^{\mathrm{ML}} = \frac{n_i}{N}$$

Now, consider a sampling process with 3 possible outcomes: R, G, and B. We observe the following sample counts:

outcome	R	G	B
count	0	3	10

What is the total sample count N?

13

EXPLANATION

$3 + 10 = 13$

What are the maximum likelihood estimates for the probabilities of each outcome?

$\theta_R^{\mathrm{ML}} =$

0

$\theta_G^{\mathrm{ML}} =$

3/13

$\theta_B^{\mathrm{ML}} =$

10/13

EXPLANATION

As derived above, the maximum likelihood estimate for an outcome is simply the count of that outcome divided by the total count:

$\theta_R^{\mathrm{ML}} = \frac{0}{13} = 0$

$\theta_G^{\mathrm{ML}} = \frac{3}{13} = 0.2307$

$\theta_B^{\mathrm{ML}} = \frac{10}{13} = 0.7692$

Now, use Laplace smoothing with strength $k = 4$ to estimate the probabilities of each outcome.

$\theta_R^{\mathrm{LAP},4} =$

4/25

$\theta_G^{\mathrm{LAP},4} =$

7/25

$\theta_B^{\mathrm{LAP},4} =$

14/25

EXPLANATION

For Laplace smoothing with strength k, increase the count for each outcome by k. In this case, the count for each outcome increases by 4, and the overall outcome increases by 12.

$\theta_R^{\mathrm{LAP},4} = \frac{0+4}{13+12} = 0.16$

$\theta_G^{\mathrm{LAP},4} = \frac{3+4}{13+12} = 0.28$

$\theta_B^{\mathrm{LAP},4} = \frac{10+4}{13+12} = 0.56$

Now, consider Laplace smoothing in the limit $k \to \infty$. Fill in the corresponding probability estimates.

$\theta_R^{\mathrm{LAP},\infty} =$

1/3

$\theta_G^{\mathrm{LAP},\infty} =$

1/3

$\theta_B^{\mathrm{LAP},\infty} =$

1/3

EXPLANATION

For Laplace smoothing with $k = \infty$, the original counts become negligible compared to k, and the likelihood estimate for each outcome becomes $\frac{\infty}{3\infty} = \frac{1}{3}$

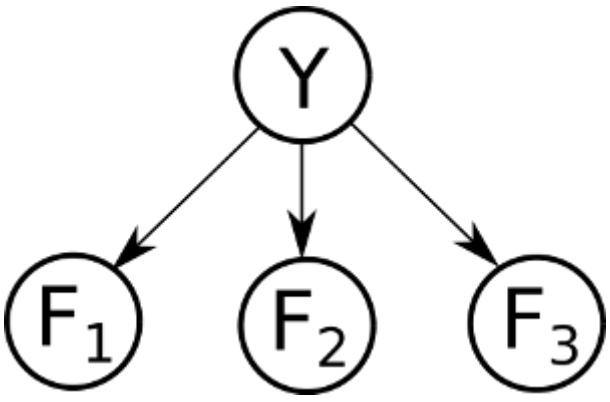
✔ Correct

Submit Last submitted on **Aug 08 at 1:27 PM**

Q2 Naïve Bayes

20 Points

In this question, we will train a Naive Bayes classifier to predict class labels Y as a function of input features F_i.



We are given the following 15 training points:

F_1	0	0	1	1	0	0	1	0	0	0	0	1	0	1	0
F_2	0	1	0	1	0	0	1	0	1	1	0	0	0	0	0
F_3	1	1	1	0	1	0	0	0	0	1	1	1	0	1	0
Y	A	A	A	A	A	A	A	A	A	A	A	B	C	C	C

Q2.1

5 Points

What is the maximum likelihood estimate of the prior $P(Y)$?

$P(Y = A)$:

11/15

$P(Y = B)$:

1/15

$P(Y = C)$:

3/15

EXPLANATION

The maximum likelihood estimate (MLE) of the prior for each label is simply the count of that label over the total number of samples.

What are the maximum likelihood estimates of the conditional probability distributions? Fill in the probability values below (the tables for the second and third features are done for you).

$P(F_1 = 0|Y = A)$:

8/11

$P(F_1 = 1|Y = A)$:

3/11

$P(F_1 = 0|Y = B)$:

0

$P(F_1 = 1|Y = B)$:

1

$P(F_1 = 0|Y = C)$:

2/3

$P(F_1 = 1|Y = C)$:

1/3

EXPLANATION

The MLE of the conditional probability distribution $P(F_i|Y)$ is the count of F_i and Y occurring together over the count of the label Y .

F_2	Y	$P(F_2\|Y)$
0	A	0.545
1	A	0.455
0	B	1.000
1	B	0.000
0	C	1.000
1	C	0.000

F_3	Y	$P(F_3\|Y)$
0	A	0.455
1	A	0.545
0	B	0.000
1	B	1.000
0	C	0.667
1	C	0.333

✔ Correct

Submit Last submitted on **Aug 08 at 1:34 PM**

Q2.2

5 Points

Now consider a new data point (F_1=1, F_2=1, F_3=1). Use your classifier to determine the joint probability of causes Y and this new data point, along with the posterior probability of Y given the new data:

$P(Y = A, F_1 = 1, F_2 = 1, F_3 = 1)$

0.049595

$P(Y = B, F_1 = 1, F_2 = 1, F_3 = 1)$

0

$P(Y = C, F_1 = 1, F_2 = 1, F_3 = 1)$

0

EXPLANATION

$P(Y, F_1, F_2, F_3) = P(F_1, F_2, F_3|Y)P(Y)$

Because of the conditional independence guarantees provided by the structure of the bayes net used for Naive Bayes, this can be simplified to $P(F_1|Y)P(F_2|Y)P(F_3|Y)P(Y)$.

$P(Y = A|F_1 = 1, F_2 = 1, F_3 = 1)$

1

$P(Y = B|F_1 = 1, F_2 = 1, F_3 = 1)$

0

$P(Y = C|F_1 = 1, F_2 = 1, F_3 = 1)$

0

EXPLANATION

These come from normalizing the joint probabilities to sum to 1.

What label does your classifier give to the new data point? (Break ties alphabetically)

- (•) A
- ( ) B
- ( ) C

EXPLANATION

The label chosen is the label with the highest value $P(Y|F_1, F_2, F_3)$.

✔ **Correct**

Submit Last submitted on **Aug 08 at 1:59 PM**

Q2.3

5 Points

The training data is repeated here for your convenience:

F_1	0	0	1	1	0	0	1	0	0	0	0	1	0	1	0
F_2	0	1	0	1	0	0	1	0	1	1	0	0	0	0	0
F_3	1	1	1	0	1	0	0	0	0	1	1	1	0	1	0
Y	A	A	A	A	A	A	A	A	A	A	A	B	C	C	C

Now use Laplace Smoothing with strength k = 2 to estimate the prior $P(Y)$ for the same data.

$P(Y = A)$:

13/21

$P(Y = B)$:

3/21

$P(Y = C)$:

5/21

Use Laplace Smoothing with strength k = 2 to estimate the conditional probability distributions below (again, the second two are done for you).

$P(F_1 = 0|Y = A)$:

2/3

$P(F_1 = 1|Y = A)$:

1/3

$P(F_1 = 0|Y = B)$:

2/5

$P(F_1 = 1|Y = B)$:

3/5

$P(F_1 = 0|Y = C)$:

4/7

$P(F_1 = 1|Y = C)$:

3/7

F_2	Y	$P(F_2\|Y)$
0	A	0.533
1	A	0.467
0	B	0.600
1	B	0.400
0	C	0.714
1	C	0.286

F_3	Y	$P(F_3\|Y)$
0	A	0.467
1	A	0.533
0	B	0.400
1	B	0.600
0	C	0.571
1	C	0.429

✔ Correct

Submit Last submitted on **Aug 08 at 1:50 PM**

Q2.4

5 Points

Now consider again the new data point (F_1=1, F_2=1, F_3=1). Use the Laplace-Smoothed version of your classifier to determine the joint probability of causes Y and this new data point, along with the posterior probability of Y given the new data:

$P(Y = A, F_1 = 1, F_2 = 1, F_3 = 1)$

0.0513625873

$P(Y = B, F_1 = 1, F_2 = 1, F_3 = 1)$

0.02057142857

$P(Y = C, F_1 = 1, F_2 = 1, F_3 = 1)$

0.01251979591

$P(Y = A|F_1 = 1, F_2 = 1, F_3 = 1)$

0.60817370131

$P(Y = B|F_1 = 1, F_2 = 1, F_3 = 1)$

0.24358200223

$P(Y = C|F_1 = 1, F_2 = 1, F_3 = 1)$

0.14824429645

What label does your (Laplace-Smoothed) classifier give to the new data point? (Break ties alphabetically)

- (•) A
- ( ) B
- ( ) C

EXPLANATION

For Laplace Smoothing, all of the same steps are taken as in part 1, except that each possible value is treated as if it were seen k (in this case 2) more times.

✔ **Correct**

Submit Last submitted on **Aug 08 at 1:58 PM**

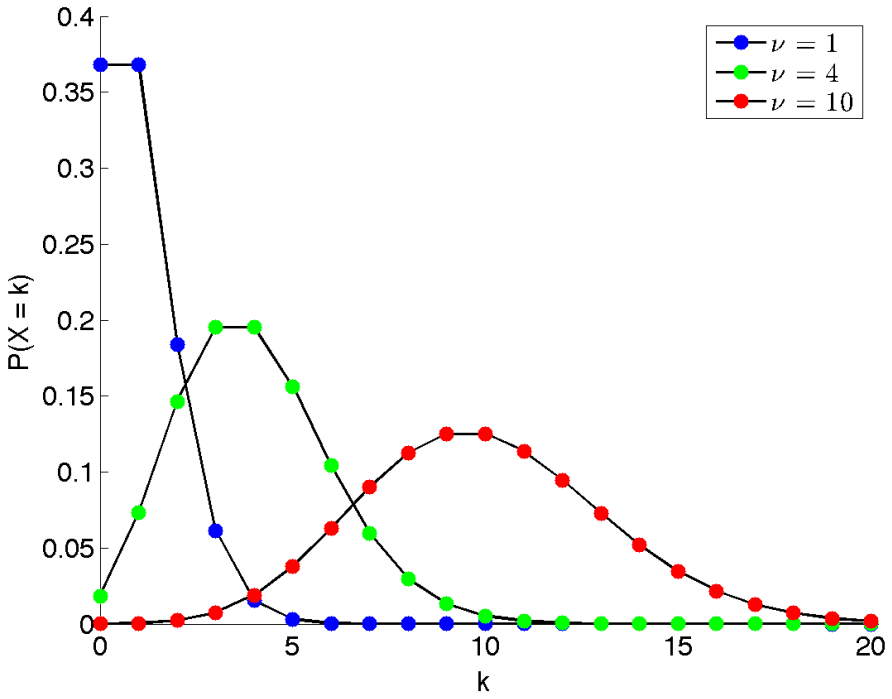
Q3 Poisson Parameter Evaluation

5 Points

We will now consider maximum likelihood estimation in the context of a different probability distribution. Under the Poisson distribution, the probability of an event occuring $X = k$ times is:

$$P(X = k) = \frac{\nu^k e^{-\nu}}{k!}$$

Here ν is the *parameter* we wish to estimate. The distribution is plotted for several values of ν below.



On a sheet of scratch paper, work out the maximum likelihood estimate for ν, given observations of several k_i. Hints: start by taking the product of the equation above over all the k_i, and then taking the log. Then, differentiate with respect to ν, set the result equal to 0, and solve for ν in terms of the k_i.

You observe the samples $k_1 = 5, k_2 = 6, k_3 = 2, k_4 = 2, k_5 = 5$. What is your maximum likelihood estimate of ν?

4

EXPLANATION

The log-likelihood, which is the log of the product of the above equation over all the k_i, is $\sum_{i=1}^{5} [k_i \log \nu - \log(k_i!) - \nu]$

Taking the derivative with respect to ν and setting this equal to 0, we get $\sum_{i=1}^{5} [\frac{k_i}{\nu} - 1] = 0$

Plugging in the values for each k_i and solving results in $\nu = \frac{20}{5} = 4.0$

✔ Correct

Submit Last submitted on **Aug 08 at 2:20 PM**

Q4 Datasets

5 Points

When training a classifier, it is common to split the available data into a training set, a hold-out set, and a test set, each of which has a different role.

Which data set is used to learn the conditional probabilities?

- (•) Training Data
- ( ) Hold-Out Data
- ( ) Test Data

Which data set is used to tune the Laplace Smoothing hyperparameters?

- ( ) Training Data
- (•) Hold-Out Data
- ( ) Test Data

Which data set is used for quantifying performance results?

- ( ) Training Data
- ( ) Hold-Out Data
- (•) Test Data

EXPLANATION

Multiple data sets are used in order to avoid overfitting, which is when the classifier performs very well on the data set used to train it, but then performs poorly on other data because the training data did not accurately represent all data.

✔ Correct

Submit Last submitted on **Aug 08 at 2:15 PM**

Q5 Multiclass Perceptron

25 Points

In this problem, we will train a Multi-class perceptron on data of the form $(f(X) \in \mathbb{R}^2, Y \in \{A, B, C\})$. In particular, we will use training data to update three weight vectors, $W_y \in \mathbb{R}^2, y = A, B, C$.

We begin with the following set of randomly-initialized weight vectors:

Y	$W_{Y,1}$	$W_{Y,2}$
A	-0.82	-0.02
B	-1.63	-0.88
C	0.39	0.65

Q5.1

5 Points

We will now incorporate the training data point $f(X) = (-1.06, 0.95); Y = C$. First fill in the resulting weight-feature dot products.

$W_A \cdot f(X)$

0.8502

$W_B \cdot f(X)$

0.8918

$W_C \cdot f(X)$

0.2041

✔ Correct

Submit Last submitted on **Aug 08 at 2:30 PM**

Q5.2

5 Points

Now update the weight values as necessary for the training point from part 1.

Note: For all of these questions, if a weight vector doesn't get updated, make sure to still write its value in the blank provided.

new $W_{A,1}$

-0.82

new $W_{A,2}$

-0.02

new $W_{B,1}$

-0.57

new $W_{B,2}$

-1.83

new $W_{C,1}$

-0.67

new $W_{C,2}$

1.6

✔ Correct

Submit Last submitted on **Aug 08 at 2:40 PM**

Q5.3
5 Points

We will now incorporate the training data point $f(X) = (0.09, 1.48); Y = A$. Fill in the resulting weight-feature dot product, and update the weight values as necessary.

$W_A \cdot f(X)$

-0.1034

$W_B \cdot f(X)$

-2.7597

$W_C \cdot f(X)$

2.3077

✔ Correct

Submit Last submitted on **Aug 08 at 2:44 PM**

Q5.4
5 Points

new $W_{A,1}$

-0.73

new $W_{A,2}$

1.46

new $W_{B,1}$

-0.57

new $W_{B,2}$

-1.83

new $W_{C,1}$

-0.76

new $W_{C,2}$

0.12

✔ Correct

Submit Last submitted on **Aug 08 at 2:45 PM**

Q5.5

5 Points

We took over from here and ran the perceptron algorithm till convergence. In case you're curious, this data set consisted of 50 data points, and the perceptron algorithm converged after 722 steps. Of these steps, 103 changed the weight vector.

At convergence, we have the following weight vectors:

Y	$W_{Y,1}$	$W_{Y,2}$
A	3.12	0.96

Y	$W_{Y,1}$	$W_{Y,2}$
B	3.11	-0.97
C	-8.29	-0.24

Use the converged perceptron to classify the new data point $f(X) = (-1.35, 0.42)$. Fill in the weight-feature dot product for each value of y.

$W_A \cdot f(X)$

-3.8088

$W_B \cdot f(X)$

-4.6059

$W_C \cdot f(X)$

11.0907

What is the predicted label?

- ○ A
- ○ B
- ◉ C

EXPLANATION

For a multi-class perceptron, the label, Y, is chosen for a data point, X, as

$$Y = argmax_y W_y \cdot f(X).$$

When training a perceptron, if the label chosen by the perceptron matches the label provided with the training data, the weights do not change.

However, if the label differs, say the perceptron classified the data point as Y, but it should have been some other label, Y^*, then the weights must be updated. This update is performed as:

$$W_y = W_y - f(X)$$

and

$$W_{y^*} = W_{y^*} + f(X)$$

✔ **Correct**

Submit Last submitted on **Aug 08 at 3:04 PM**

Q6 A Variant on the Perceptron Algorithm

10 Points

You were recently promoted to the Vice President of Recruiting Science at Pacapalooza Technologies. Pacapalooza is expanding rapidly, and you decide to use machine learning to hire the best and brightest. To do so, you have the following available to you for each candidate i in the pool of candidates I: (i) their GPA, (ii) whether they took CS 164 with Hilfinger and received an A, (iii) whether they took CS 188 and received an A, (iv) whether they have a job offer from Pactronic LLC, (v) whether they have a job offer from Pacmania Corp., and (vi) the number of misspelled words on their resume. You decide to represent each candidate $i \in I$ by a corresponding 6-dimensional feature vector $f(x^{(i)})$. You believe that if you just knew the right weight vector $w \in \mathbb{R}^6$ you could reliably predict the quality of a candidate i by computing $w^T f(x^{(i)})$. To determine w, you sample

pairs of candidates from the pool. For a pair of candidates (k, l) you can have them face off in a "Pacapalooza-fight." The result is $\text{score}(k > l)$, which tells you that a candidate k is at least $\text{score}(k > l)$ better than candidate l. Note that the score will be negative when l is a better candidate than k. Assume you collected scores for a set of pairs of candidates P, that $\text{score}(k > l) = -\text{score}(k < l)$, and that $\text{score}(k > l) \neq 0$ for any pair $(k, l) \in P$.

Q6.1

5 Points

You decide to employ a perceptron-like algorithm to determine w, where your dataset is P.

Suppose that we encounter a pair $(k, l) \in P$ for which $\text{score}(k > l) > 0$. How do we update w?

- (●) If $w^T f(x^{(k)}) \geq w^T f(x^{(l)}) + \text{score}(k > l)$, do nothing. Otherwise update $w \leftarrow w + f(x^{(k)}) - f(x^{(l)})$.
- ( ) Update $w \leftarrow w + f(x^{(k)}) - f(x^{(l)})$.
- ( ) If $w^T f(x^{(k)}) \geq w^T f(x^{(l)}) + \text{score}(k > l)$, do nothing. Otherwise update $w \leftarrow w - f(x^{(k)}) + f(x^{(l)})$.
- ( ) If $w^T f(x^{(k)}) \geq w^T f(x^{(l)}) + \text{score}(k > l)$, do nothing. Otherwise update $w \leftarrow w + w^T(f(x^{(k)}) - f(x^{(l)}))$.
- ( ) Update $w \leftarrow w - f(x^{(k)}) + f(x^{(l)})$.

Suppose that we encounter a pair $(k, l) \in P$ for which $\text{score}(k > l) < 0$. How do we update w?

- If $w^T f(x^{(k)}) \geq w^T f(x^{(l)}) + \text{score}(k > l)$, do nothing. Otherwise update $w \leftarrow w - f(x^{(l)}) - f(x^{(k)})$.
- Update $w \leftarrow w - f(x^{(k)}) + f(x^{(l)})$.
- If $w^T f(x^{(l)}) \geq w^T f(x^{(k)}) - \text{score}(k > l)$, do nothing. Otherwise update $w \leftarrow w + f(x^{(l)}) - f(x^{(k)})$.
- Update $w \leftarrow w + f(x^{(k)}) - f(x^{(l)})$.
- If $w^T f(x^{(k)}) \geq w^T f(x^{(l)}) + \text{score}(k > l)$, do nothing. Otherwise update $w \leftarrow w - w^T(f(x^{(k)}) - f(x^{(l)}))$.

EXPLANATION

The idea behind the solution is to use a margin-based perceptron. Given a pair of candidates k, l, our goal is to have $w^T f(x^{(k)}) \geq w^T f(x^{(l)}) + \text{score}(k > l)$ if $\text{score}(k > l) > 0$. Otherwise, we want the opposite inequality, that is, $w^T f(x^{(l)}) \geq w^T f(x^{(k)}) - \text{score}(k > l) = w^T f(x^{(k)}) + \text{score}(l > k)$.

So, our perceptron-like algorithm is as follows: we repeat

1. Choose a random pair $(k, l) \in P$ (you can also sequentially run through P)
2. If $\text{score}(k > l) > 0$: if $w^T f(x^{(k)}) \geq w^T f(x^{(l)}) + \text{score}(k > l)$, do nothing; else, update $w \leftarrow w + f(x^{(k)}) - f(x^{(l)})$.
3. If $\text{score}(k > l) \leq 0$: if $w^T f(x^{(l)}) \geq w^T f(x^{(k)}) - \text{score}(k > l)$, do nothing; else, update $w \leftarrow w + f(x^{(l)}) - f(x^{(k)})$.

✔ **Correct**

Submit Last submitted on **Aug 08 at 3:11 PM**

Q6.2

5 Points

Your perceptron-like algorithm is unable to reach zero errors on your training data. Which of the following techniques would help improve performance on the training data?

- [ ] Running the perceptron algorithm for a longer period of time; since the perceptron algorithm is guaranteed to keep equal or reduce the number of errors at each time step, we are guaranteed to eventually reach zero errors.

- [x] Add higher-order features to our list of six features, e.g., pairwise products, and run our perceptron algorithm with this newly constructed dataset. New features increase the dimensionality of the space, and improve the chances that the data is separable.

- [ ] Removing some of the features from the training data, and training the perceptron on this subset of data. Too many features increases the chance of overfitting on the training data, which would decrease performance on the training data.

- [ ] Collect a larger set of data, so the perceptron algorithm does a better job of fitting to the data distribution; with a small training set, the perceptron cannot fully learn w, causing it to produce errors on the training data.

EXPLANATION

Running the algorithm for a longer period of time does not necessarily help improve performance, since the perceptron algorithm doesn't necessarily decrease the number of errors at each time step.

Adding higher-order features could help lower the error rate, since the data could be linearly separable in this higher dimensional space.

Removing features doesn't help, as this reduces dimensionality. Reducing dimensionality will keep equal or increase the training error.

Training on a larger set of data will not help reduce the number of errors on the current training set, but will rather likely increase the number of errors. The introduction of more samples will cause the perceptron to have to focus on properly classifying the new samples.

✔ Correct

Submit Last submitted on **Aug 08 at 3:17 PM**

Q7 Neural Network Gradient Computation

5 Points

Q7.1

2.5 Points

Let $f_w(x) = \frac{1}{1+e^{\{-w^T x\}}}$. Which of the following expressions are equivalent to $\nabla_w f_w(x)$?

- [ ] $\frac{-e^{-w^T x}}{\left(1+e^{-w^T x}\right)^2} \cdot x$
- [ ] $\frac{e^{-w^T x}}{1+e^{-w^T x}} \cdot x$
- [x] $\frac{e^{-w^T x}}{\left(1+e^{-w^T x}\right)^2} \cdot x$
- [ ] $f_w(x)^2 \cdot (1 + f_w(x)) \cdot x$
- [x] $f_w(x) \cdot (1 - f_w(x)) \cdot x$
- [ ] $f_w(x) \cdot (1 + f_w(x)) \cdot x$

✔ Correct

Submit Last submitted on **Aug 08 at 3:28 PM**

Q7.2

2.5 Points

Let $f_w(x) = \frac{e^{w^T x} - e^{-w^T x}}{e^{w^T x} + e^{-w^T x}}$. Which of the following expressions are equivalent to $\nabla_w f_w(x)$?

- [x] $\left[\frac{2}{e^{w^T x}+e^{-w^T x}}\right]^2 \cdot x$
- [ ] $\left[\frac{4}{e^{2w^T x}-1} - \frac{4}{\left(e^{2w^T x}-1\right)^2}\right] \cdot x$
- [ ] $\left[\frac{2}{e^{2w^T x}+1} - \frac{2}{\left(e^{2w^T x}+1\right)^2}\right] \cdot x$
- [x] $\left[\frac{4}{e^{2w^T x}+1} - \frac{4}{\left(e^{2w^T x}+1\right)^2}\right] \cdot x$
- [ ] $\left[\frac{2}{e^{w^T x}-e^{-w^T x}}\right]^2 \cdot x$
- [ ] $\left[\frac{1}{e^{w^T x}+e^{-w^T x}}\right]^2 \cdot x$

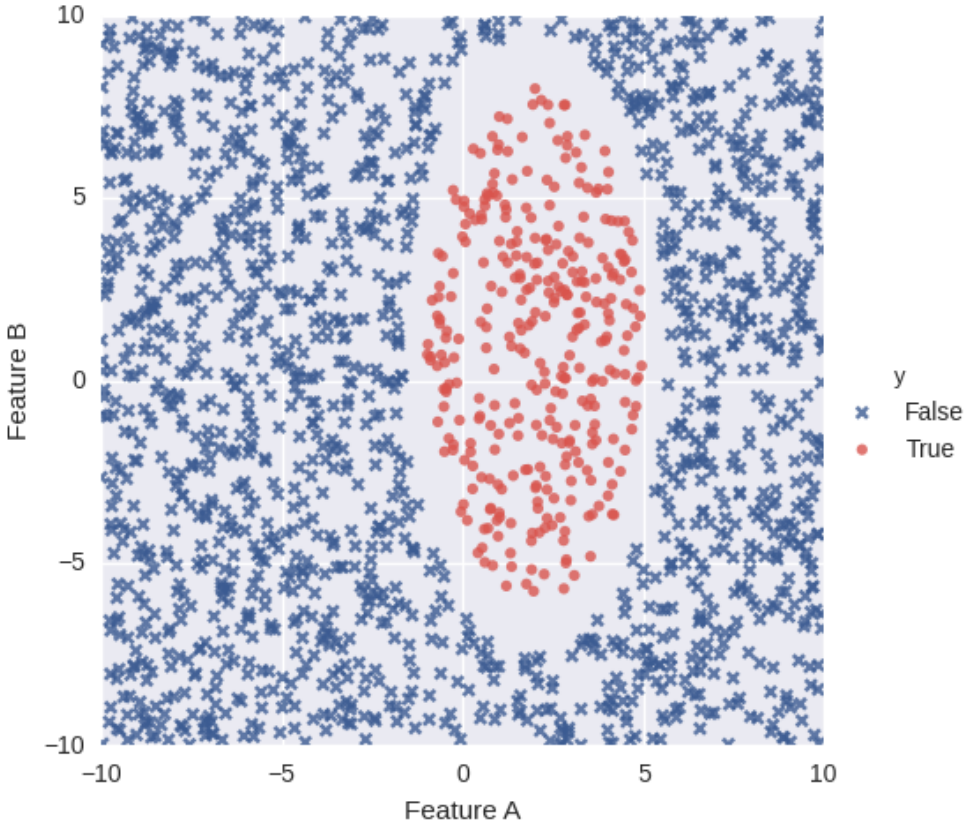
✔ **Correct**

Submit Last submitted on **Aug 08 at 3:33 PM**

Q8 Neural Network Separability

5 Points

It is well known that Pactronic LLC is the premier manufacturer of Pacmen. At Pactronic LLC, quality control is currently done manually -- a group of scientists decide whether a Pacman is ready to be released into the wild based on (Feature A) a Pacman's intelligence score and (Feature B) a Pacman's empathy score. Here are many examples of Pacmen that have been released and withheld in the past. Each dot corresponds to a Pacman, and responds to the following question as true or false: this Pacman is ready to be released.



Q8.1

2.5 Points

As the Vice President of Science, you would like to automate the decision making process, and decide to use the perceptron algorithm. Which of the following subsets of features would allow you to perfectly classify whether or not a Pacman can be released in the wild?

- [ ] A, B
- [x] A^2, AB, B^2, A, B
- [ ] $A, B, X,$ where $X = (A \geq C_1) \wedge (B \geq C_2)$ for some fixed C_1, C_2 th
- [ ] A
- [ ] B

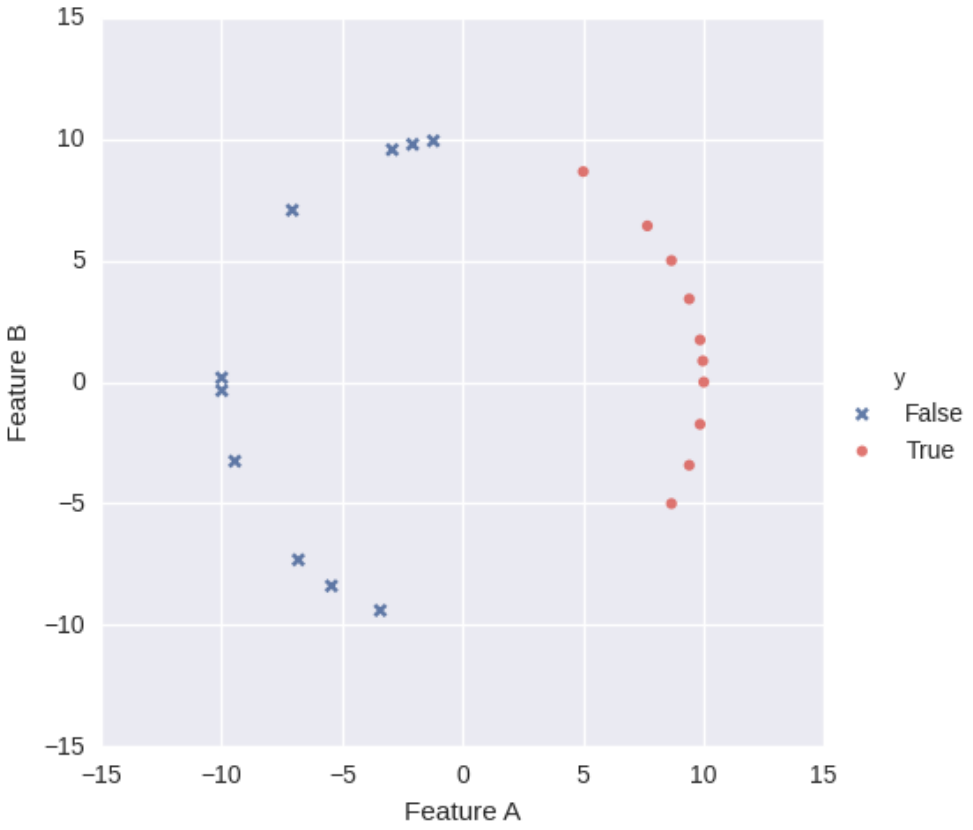
✔ Correct

Submit Last submitted on **Aug 08 at 3:35 PM**

Q8.2

2.5 Points

The CEO of Pactronic soon decides that the company will be focusing on creating fewer, but much better Pacmen. This calls for an entire re-design of the Pacman. Accordingly, the scientists come up with the latest and greatest generation of Pacmen, and once again seek your advice in quality control. Here are the newest Pacmen, and their respective features:



Which of the following subsets of features would allow you to perfectly classify whether or not a Pacman can be released in the wild?

- ✔ A, B
- ✔ A^2, AB, B^2, A, B
- ✔ $A, B, X,$ where $X = (A \geq C_1) \wedge (B \geq C_2)$ for some fixed C_1, C_2 tl
- ✔ A
- ☐ B

✔ **Correct**

Submit Last submitted on **Aug 08 at 3:37 PM**

Q9 Local Optima and Gradient Descent

15 Points

After a busy year of chasing ghosts, Pacman and Paclady are planning to visit the Kakslauttanen Arctic Resort for their winter vacation. Paclady who is particularly fond of skiing, excitedly begins planning ahead. Pacman, who is apprehensive of skiing (when asked why, he rambles on about the Aspen Red Ghost Chase of 2012, but we won't get into that), reluctantly agreed to go skiing, but under one condition: Paclady must tell Pacman how steep the slopes are at several points of interest.

Paclady asks the resort for terrain details, and receives the following graph. The resort says at any given location x, $f(x)$ models the terrain height.

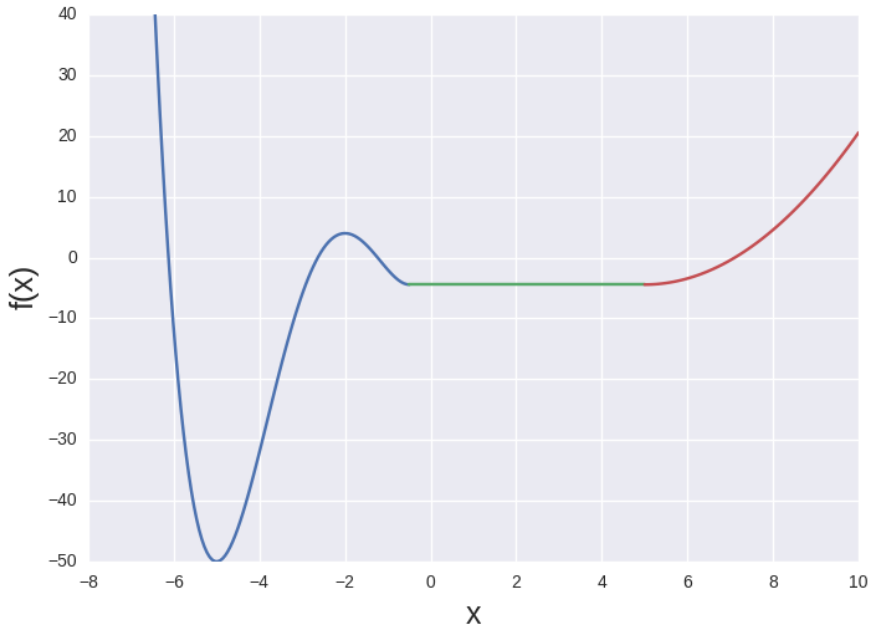
Specifically:

(1) when $x \leq -\frac{1}{2}, f(x) = \frac{1}{2}x^4 + 5x^3 + \frac{27}{2}x^2 + 10x$

(2) when $-\frac{1}{2} \leq x \leq 5, f(x) = -\frac{71}{16}$

(3) and when $x \geq 5 f(x) = x^2 - 10x + \frac{329}{16}$.

See below for a plot:



The local optima for f lie at $x = -5$ and $x = -2$, with a plateau in the region $-1/2 \leq x \leq 5$

Q9.1

2 Points

Paclady decides to compute derivatives to measure how steep slopes are. Evaluate $f'(-6)$

-44

✔ Correct

Submit Last submitted on **Aug 08 at 3:40 PM**

Q9.2

1 Point

Evaluate $f'(0)$

0

✔ Correct

Submit Last submitted on **Aug 08 at 3:41 PM**

Q9.3

2 Points

Evaluate $f'(8)$

6

✔ **Correct**

Submit Last submitted on **Aug 08 at 3:41 PM**

Q9.4

5 Points

Pacman and Paclady get to the resort, and have a fantastic time skiing, but get lost. Unfortunately, a blizzard kicks in right then, reducing visibility. As Pacman panics and brings up the Aspen Red Ghost Chase of 2012, Paclady remembers that their glass igloo cabin is located at the global minimum elevation point of the resort $(x = -5)$. The blizzard complicates things, since they can't ski due to the reduced visibility for safety. After thinking for a minute, Pacman says, "Aha! We can get home in that case by following gradient descent, as long as we employ a small step size -- once we hit a gradient of 0, we know we're home!" Paclady pauses and says, "Your algorithm almost works, but it depends on where in the resort we currently are." Check all regions where Pacman and Paclady can be, and still find their igloo, assuming that they employ gradient descent with a small step size and stop walking when they encounter a gradient of 0.

- ✔ $x < -5$
- ✔ $-5 < x < -2$
- ☐ $2 < x < -1/2$
- ☐ $1/2 < x < 3$
- ☐ $3 < x < 5$
- ☐ $x > 5$

✔ **Correct**

Submit Last submitted on **Aug 08 at 3:43 PM**

Q9.5

5 Points

While slowly trudging to their igloo via gradient descent, Pacman and Paclady get into an argument. Pacman complains that trudging down a hill is tiresome, and that they instead should have gotten an igloo closer to $x = 3$. Paclady says that Pacman's previous gradient descent algorithm wouldn't lead them to the igloo in this case, unless they were already at the igloo. Why is this the case?

- Gradient descent would cause Pacman and Paclady to reach $x = -2$ rather than $x = 3$, since it is at a local maximum.
- Gradient descent terminates when it reaches a gradient of 0, and neighboring regions around $x = 3$ all have a gradient of 0, so Pacman and Paclady would stop searching outside of $x = 3$, within the plateau.
- When gradient descent is stuck in a plateau, it searches for regions with negative rather than zero gradient.
- When gradient descent is stuck in a plateau, it searches for regions with positive rather than zero gradient.
- Gradient descent seeks to maximize a function, which would lead Pacman and Paclady either to $-\infty$ or to ∞

✔ **Correct**

Submit Last submitted on **Aug 08 at 3:45 PM**

Submit Assignment