MIDTERM EXAM

CMU 10-601B: MACHINE LEARNING (FALL 2016)

Oct. 10, 2016

Name: _____

Andrew ID: _____

START HERE: Instructions

- This exam has 20 pages and 5 Questions (page one is this cover page). Check to see if any pages are missing. Enter your name and Andrew ID above.
- You are allowed to use one page of notes, front and back.
- Electronic devices are not acceptable.
- Some of the questions are True/False or Multiple Choice with no explanation required. In this case, we will give partial credit if you supply an incorrect choice, but a partially correct justification.
- Note that the questions vary in difficulty. Make sure to look over the entire exam before you start and answer the easier questions first.

Question	Points	Extra Credit	Score
1	25	0	
2	15	0	
3	20	3	
4	20	4	
5	20	4	
Total	100	11	

1 Probability, Naive Bayes and MLE [25 pts]

1.1 Probability

For each question, circle the correct option.

1. **[3 pts]** Which of the following expressions is equivalent to $p(A|B,C,D)$?

   (a) $\frac{p(A,B,C,D)}{p(C|B,D)p(B|D)p(D)}$

   (b) $\frac{p(A,B,C,D)}{p(B,C)p(D)}$

   (c) $\frac{p(A,B,C,D)}{p(B,C|D)p(B)p(C)}$

   Answer is (a). $p(A|BCD) = \frac{p(A,B,C,D)}{p(B,C,D)} = \frac{p(A,B,C,D)}{p(C|B,D)p(B,D)} = \frac{p(A,B,C,D)}{p(C|B,D)p(B|D)p(D)}$

2. **[3 pts]** Let μ be the mean of some probability distribution. $p(\mu)$ is always non-zero.

   (a) True

   (b) False

   False. Think of the mean of the Bernoulli distribution, in which $X \in \{0, 1\}$.

1.2 Naïve Bayes

Consider the following data. It has 4 features $\mathbf{x} = (x_1, x_2, x_3, x_4)$ and 3 labels $(+1, 0, -1)$. Assume that the probabilities $p(x_i|y)$ is a Bernoulli distribution and $p(y)$ is a Categorical distribution. Answer the questions that follow under the Naïve Bayes assumption.

x_1	x_2	x_3	x_4	y
1	1	0	1	+1
0	1	1	0	+1
1	0	1	1	0
0	1	1	1	0
0	1	0	0	-1
1	0	0	1	-1
0	0	1	1	-1

1. **[5 pts]** Compute the Maximum Likelihood Estimate for $p(x_i = 1|y), \forall i \in [1, 4], \forall y \in \{+1, 0, -1\}$ See filled in table

2. **[5 pt]** Compute the Maximum Likelihood Estimate for the prior probabilities $p(y = +1), p(y = 0), p(y = -1)$.

   $p(y = +1) = \frac{2}{7}$, $p(y = 0) = \frac{2}{7}$ and $p(y = -1) = \frac{3}{7}$.

	$y=+1$	$y=0$	$y=-1$
$x_1=1$			
$x_2=1$			
$x_3=1$			
$x_4=1$			

	$y=+1$	$y=0$	$y=-1$
$x_1=1$	0.5	0.5	1/3
$x_2=1$	1	0.5	1/3
$x_3=1$	0.5	1	1/3
$x_4=1$	0.5	1	2/3

3. **[3 pts]** Use the values computed in the above two parts to classify the data point $(x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1)$ as either belonging to class $+1, 0$ or -1.

   Circle one: +1 0 -1

   Justification:

   According to Naïve Bayes assumption, samples are independent given y, thus we can write the conditional joint probability as

$$p(x_1=1,x_2=1,x_3=1,x_4=1) = p(x_1=1,x_2=1,x_3=1,x_4=1|y)p(y) \quad (1)$$

$$= p(y)\prod_{i=1}^{4} p(x_i=1|y). \quad (2)$$

   We calculate the probability given different value on y and pick the y that gives us largest probability.

$$p(y=+1)\prod_{i=1}^{4} p(x_i=1|y=+1) = \frac{1}{2}\cdot 1\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot\frac{2}{7} = \frac{1}{28} \quad (3)$$

$$p(y=0)\prod_{i=1}^{4} p(x_i=1|y=0) = \frac{1}{2}\cdot\frac{1}{2}\cdot 1\cdot 1\cdot\frac{2}{7} = \frac{1}{14} \quad (4)$$

$$p(y=-1)\prod_{i=1}^{4} p(x_i=1|y=-1) = \frac{1}{3}\cdot\frac{1}{3}\cdot\frac{1}{3}\cdot\frac{2}{3}\cdot\frac{3}{7} = \frac{2}{189} \quad (5)$$

   Since $y=0$ yields the largest value, we classify the data as $\hat{y}=0$.

1.3 MLE vs MAP

For each question state **True** or **False** and give one line justifications.

1. **[3 pts]** The value of the Maximum Likelihood Estimate (MLE) is equal to the value of the Maximum A Posteriori (MAP) Estimate with a uniform prior.

   Circle one: True False

   Justification:

   True. We know the posterior is proportional to the product of likelihood and prior, i.e.,

   $$p(\theta|x) \propto p(x|\theta)p(\theta). \tag{6}$$

   Since the uniform prior gives us a constant value on $p(\theta)$, after proper normalization, we know that the likelihood of MLE and the posterior of MAP are the same. Thus, the MLE and MAP estimators are also the same.

2. **[3 pts]** The bias of the Maximum Likelihood Estimate (MLE) is typically less than or equal to the bias of the Maximum A Posteriori (MAP) Estimate.

   Circle one: True False

   Justification:

   True. The MAP estimate injects some prior knowledge and typically adds bias.

2 To err is machine-like [15 pts]

2.1 Train and test errors

In this problem, we will see how you can debug a classifier by looking at its train and test errors. Consider a classifier trained till convergence on some training data $\mathcal{D}^{\text{train}}$, and tested on a separate test set $\mathcal{D}^{\text{test}}$. You look at the test error, and find that it is very high. You then compute the training error and find that it is close to 0.

1. **[2 pts]** What is this scenario called?

   Overfitting

2. **[5 pts]** Which of the following is expected to help? Select all that apply.

   (a) Increase the training data size.

   (b) Decrease the training data size.

   (c) Increase model complexity (For example, if your classifier is an SVM, use a more complex kernel. Or if it is a decision tree, increase the depth).

   (d) Decrease model complexity.

   (e) Train on a combination of $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$ and test on $\mathcal{D}^{\text{test}}$

   (f) Conclude that Machine Learning does not work.

   (a)(d)
   Rubric:

   - Not choosing either a or d: -1.5 each
   - Choosing b, c, e or f: -0.5 each

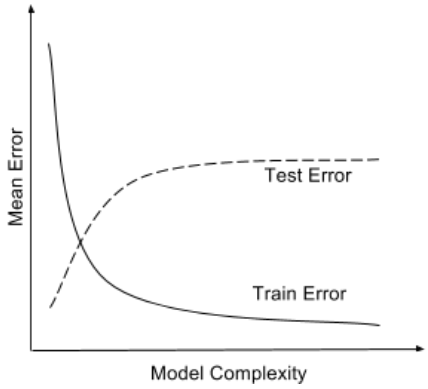
3. **[5 pts]** Explain your choices .

   In order to address the problem of overfitting, we should ensure that the model generalizes better (and thus reduce the generalization error). Increasing the data size lets the model generalize better. Decreasing the complexity reduces the sufficient sample size required to generalize well. So 2a and 2d are expected to help. Concretely, the PAC bound is inversely proportional to the training data size and directly proportional to the hypothesis space complexity. So, to reduce the bound, we need to increase training data size and/or decrease the model complexity.

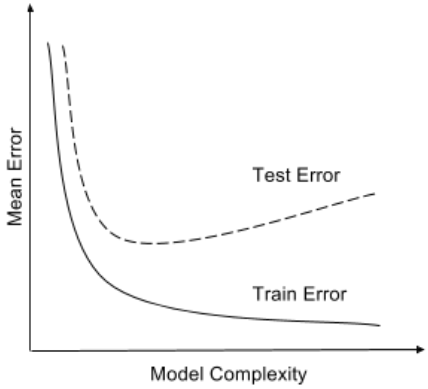
   We test on unseen data to get an estimate of the true error, and doing this on data seen during training defeats the purpose.

4. **[3 pts]** Say you plot the train and test errors as a function of the model complexity. Which of the following two plots is your plot expected to look like?

   Circle one: Plot (a) Plot (b)



(a)



(b)

(b). When model complexity increases, model can fit better, so training error will decrease. But when it overfits, testing error will increase.

3 Support Vector Machines [20+3 pts]

3.1 T/F, Multiple Choice

For true/false, circle one answer. For multiple choice, circle all answers that apply. No justifications are needed.

1. **[3 pts]** Applying the kernel trick enables features to be mapped into a higher dimensional space, at a cost of higher computational complexity to operate in the higher dimensional space.

   Circle one: True False

   False. The kernel trick occurs with no loss in computational complexity.

2. **[3 pts]** Suppose $\phi(\mathbf{x})$ is an arbitrary feature mapping from input $\mathbf{x} \in \mathcal{X}$ to $\phi(\mathbf{x}) \in \mathbb{R}^N$ and let $K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$. Then $K(\mathbf{x}, \mathbf{z})$ will always be a valid kernel function.

   Circle one: True False

   True. Kernels can be represented as high-dimensional dot products.

3. **[3 pts]** Given the same training data, in which the points are linearly separable, the margin of the decision boundary produced by SVM will always be greater than or equal to the margin of the decision boundary produced by Perceptron.

   Circle one: True False

   True. SVM solves for the maximum margin. Perceptron does not.

4. **[3 pts]** Recall that the formulation of the SVM in the case where the data is not linearly separable is as follows:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i$$
$$\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i) \geq 1 - \xi_i \quad \forall i = 1, ..., N$$
$$\xi_i \geq 0 \quad \forall i = 1, ..., N$$

   where $(\mathbf{x}_i, y_i)$ are training samples and $\mathbf{w}$ defines a linear decision boundary. If the data are not linearly separable (i.e., overlap between classes), SVM can use a tradeoff parameter C that allows for errors in the training samples. Which of the following may happen to the size of the margin if the tradeoff parameter C is increased?

   Circle all that apply: Remains the same Increases Decreases

Decrease, remain the same. $C \to \infty$ implies hard-margin. Either the margin will shrink (to account for misclassifications) or for some C, the margin will remain the same, because the number of misclassifications stays the same.

3.2 Short Answer

Give brief explanations for the following questions.

1. **[3 pts]** SVM is a discriminative classifier, whereas Naïve Bayes is a generative classifier. Describe one statistical advantage SVM has over Naïve Bayes.
   Explanation: Many statistical advantages exist:

   (a) SVM can account for correlations between features (NB assumes conditional independence)

   (b) NB makes strong modelling assumptions and fails when those assumptions are broken

   (c) SVM does not need to add smoothing as a prior (SVM is regularized by design)

   (d) SVM can be kernelized to account for nonlinear decision boundaries

   (e) SVM can ignore outliers far away from the decision boundary; NB still accounts for them

   Note that we will not accept solutions that say SVM is computationally-faster because it uses less training data. 1) This is not a statistical advantage, it would be computational. 2) This is not necessarily true. SVM still must use all of the training data to optimize its parameters.

2. **[5 pts]** For logistic regression, the probability of instance $\mathbf{x}$ being in class $y = 1$ is

$$P(y = 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T\mathbf{x})}$$

   We can map $\mathbf{x} \in \mathcal{X}$ to an arbitrary feature mapping $\phi(\mathbf{x}) \in \mathbb{R}^N$, and recall $K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$. Is there a kernelized version of logistic regression? If yes, show the formula for $\mathbf{w}$ in terms of $\phi(\mathbf{x}_i)$ and explain how the kernel trick can be used. If no, explain why we cannot use the kernel trick.

   Circle one: yes no
   Explanation: Yes.

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i)$$

$$\mathbf{w}^T\mathbf{x} = \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i)^T \phi(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, x)$$

3. **Extra Credit: [3 pts]** One formulation of soft-margin SVM optimization problem is:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i$$
$$\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i) \geq 1 - \xi_i \quad \forall i = 1, ..., N$$
$$\xi_i \geq 0 \quad \forall i = 1, ..., N$$

where $(\mathbf{x}_i, y_i)$ are training samples and $\mathbf{w}$ defines a linear decision boundary.

Derive a formula for ξ_i when the objective function achieves its minimum (No steps necessary). Note it is a function of $y_i\mathbf{w}^\top\mathbf{x}_i$. Sketch a plot of ξ_i with $y_i\mathbf{w}^\top\mathbf{x}_i$ on the x-axis and value of ξ_i on the y-axis. What is the name of this function?

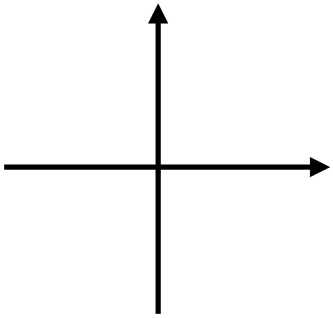
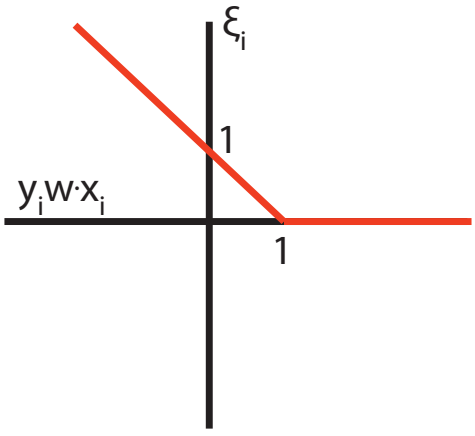


Figure 2: Plot here

$\xi_i = \max(0, 1 - y_i\mathbf{w}^T\mathbf{x}_i)$
hinge loss function



4 Learning Theory [20+4 pts]

4.1 PAC Learning

Let X be the feature space and let D be the underlying distribution over X. We have training samples

$$S : \{(x_1, c^\star(x_1)), \cdots, (x_m, c^\star(x_m))\},$$

x_i i.i.d from D. We assume that the labels are binary, $c^*(x_i) \in \{+1, -1\}$.
Let $\mathcal{H}$ be a hypothesis space and let $h \in \mathcal{H}$ be a hypothesis. We use

$$err_S(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}(h(x_i) \neq c^\star(x_i))$$

to denote the training error and

$$err_D(h) = \mathbf{Pr}_{x\sim D}(h(x) \neq c^\star(x))$$

to denote the true error. Recall the theorem from class:

Theorem 1. If the hypothesis space is finite, in the realizable case

$$m \geq \frac{1}{\epsilon}\left[\ln(|\mathcal{H}|) + \ln\left(\frac{1}{\delta}\right)\right]$$

labeled examples are sufficient so that with probability at least $1-\delta$, all $h \in \mathcal{H}$ with $err_D(h) \geq \epsilon$ have $err_S(h) > 0$ or equivalently if $err_S(h) = 0$, $err_D(h) \leq \epsilon$.

1. **[3 pts]** What does PAC learning stand for? What is the correspondence between ϵ, δ and the full name?

   "Probably approximately correct." The hypotheses we find with m examples are *probably* (with probability $p \geq 1-\delta$) *approximately* correct, with $err_D(h) \leq \epsilon$

2. **[3 pts]** Briefly explain what is the realizable case and what is the agnostic case. Why is Theorem 1 not meaningful in the agnostic case?

   Realizable- the true classifier, $c*$, is in $\mathcal{H}$
   Agnostic- $c* \notin H$, but "close"
   In the agnostic case, since $c* \notin \mathcal{H}$, we will not be able to find a hypothesis $h \epsilon \mathcal{H}$ for which $err_S(h) = 0$

3. **[2 pts] True or False:** The true error, $err_D(h)$, of any hypothesis h is an upper bound on its training error, $err_S(h)$ on the sample S.

   Circle one: True False

   Explanation:

   False. We said true error is close to training error, but it might be smaller than training error, so it is not an upper bound.

4.2 VC Dimension and Generalization

1. **[6 pts]** Briefly explain in **2-3 sentences** the importance of sample complexity and VC dimension for machine learning.

   Sample complexity and VC dimension are used to determine how many samples we need in our training data in order to be confident that the model we find will generalize to unseen data. We can also use them to help choose a hypothesis space $\mathcal{H}$ that will reduce the risk of overfitting. VC dimension is helpful when we are dealing with infinite hyptohesis space.

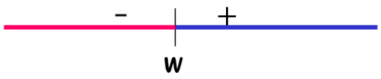
2. **[2 pts]** (True or False) VC dimension of linear separators in $\mathbb{R}^d$ is infinity. You do not need to justify your answer.

   Circle one: True False. The VC dimension is $d + 1$.

   True.

3. **[4 pts]** Let $\mathcal{H}$ be the set of thresholds in $\mathbb{R}$, i.e., each classifier corresponds to a real number w where examples $x \in \mathbb{R}$ are labeled $+1$ if $x \geq w$ and are labeled -1 otherwise. See Figure 3. What is the VC dimension of $\mathcal{H}$? Please justify your answer.

Figure 3: Thresholds in $\mathbb{R}$.



$VC(\mathcal{H}) = 1$

To prove this, we need to show that there is a set of 1 point that can be shattered and that no set of 2 points can be shattered.

Any set of 1 point x can be shattered since we can find a classifier in $\mathcal{H}$ such that $w < x$ and another such that $w > x$

No set of 2 points (x_1, x_2) can be shattered. WLOG, assume $x_1 < x_2$. If x_1 is labeled as positive and x_2 as negative, we cannot find a threshold that will correctly classify these points. Since there is an assignment for any two points for which we cannot find a threshold, they cannot be shattered in H.

4.3 Extra Credit

1. **[Extra Credit: 2 pts]** Let $\mathcal{H}$ be the set of linear separators in $\mathbb{R}^2$, i.e., each classifier corresponds to a (w, b) pair where $w \in \mathbb{R}^2$ and $b \in \mathbb{R}$ where examples $x \in \mathbb{R}^2$ are labeled $+1$ if $x^\top w \geq b$ are labeled -1 otherwise. See Figure 4. What is VC dimension of $\mathcal{H}$? Please justify your answer.

Figure 4: Linear Separators in $\mathbb{R}^2$.

$VC(\mathcal{H}) = 3$

First, we need to show that there is a set of three points that can be shattered. If we draw three points in $\mathbb{R}^2$ we'll see that no matter how we label them, we can always draw a linear separator such that the +1 points are on one side of it and the -1 points are on the other.

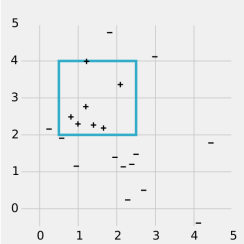
Next, we need to show that there is not a set of four points that can be shattered. If we have four points, they must either create the corners of a rectangle or one must be within the convex hull of the other three. If they form a rectangle, we cannot find a linear separator for the labeling where two diagonal corners are +1 and the other two are -1. If one is within the convex hull of the others, we cannot find a linear separator for the labelling where the inner one is the opposite of all three of the others (e.g. if three are +1 and the one in the convex hull is -1).

2. **[Extra Credit: 2 pts]** Let $\mathcal{H}$ be the set of axis-aligned rectangles in $\mathbb{R}^2$, where examples inside the rectangles are labeled $+1$ and outside the rectangles are -1. See figure 5. What is VC dimension of $\mathcal{H}$? Please justify your answer.

$VC(\mathcal{H}) = 4$

If we draw four points in $\mathbb{R}^2$ we can always draw a rectangle such that any 0, 1, 2, 3, or 4 are bounded by the rectangle.

Figure 5: Rectangles in $\mathbb{R}^2$.



If we have 5 points, they will either all be on the edge of a rectangle or one will be within the convex hull of the others. If they are all on the edge of a rectangle and three are +1 and the other is -1, we cannot find a rectangle that will bound the +1 and not the -1. Similarly, if one is in the convex hull of the others, if that one is -1 and the others are +1, we cannot bound the +1s without boudning the -1.

5 Linear and Logistic Regression [20+4 pts]

5.1 Linear Regression

Please circle **True** or **False** for the following questions, providing brief explanations to support your answer.

1. **[4 pts]** Consider n data points, each with one feature x_i and its corresponding output y_i. In linear regression, we assume $y_i \sim \mathcal{N}(wx_i, \sigma^2)$ and compute $\hat{w}$ through MLE.

   Suppose $y_i \sim \mathcal{N}(\log(wx_i), 1)$ instead. Then the maximum likelihood estimate $\hat{w}$ is the solution to the following equality:

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

   See question 1.3 for a definition of the Gaussian pdf.

   Circle one: **True** **False**
   Brief explanation:

   False. The likelihood function can be written as

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

   Differentiating wrt w and setting to zero gives us

   $$\sum_{i=1}^{n} 2(y_i - \log(wx_i))\frac{x_i}{wx_i} = 0$$
   $$\therefore \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \log(wx_i)$$

2. **[3 pts]** Consider a linear regression model with only one parameter, the bias, ie., $y = \beta_0$. Then given n data points (x_i, y_i) (where x_i is the feature and y_i is the output), minimizing the sum of squared errors results in β_0 being the median of the y_i values.

   Circle one: **True** **False**
   Brief explanation:

   False. $\sum_{i=1}^{n}(y_i - \beta_0)^2$ is the training cost, which when differentiated and set to zero gives $\beta_0 = \frac{\sum_{i=1}^{n} y_i}{n}$, the mean of the y_i values.

3. **[3 pts]** Given data $D = \{(x_1, y_1), ..., (x_n, y_n)\}$, we obtain $\hat{w}$, the parameters that minimize the training error cost for the linear regression model $y = w^T\mathbf{x}$ we learn from D.

   Consider a new dataset D_{new} generated by duplicating the points in D and adding 10 points that lie along $y = \hat{w}^T\mathbf{x}$. Then the $\hat{w}_{\text{new}}$ that we learn for $y = w^T\mathbf{x}$ from D_{new} is equal to $\hat{w}$.

   Circle one: True False
   Brief explanation:

   True. The new squared error can be written as $2k + m$, where k is the old squared error. $m = 0$ for the 10 points that lie along the line, the lowest possible value for m. And $2k$ is least when k is least, which is when the parameters don't change.

4. **Extra Credit: [2 pts]** For linear regression, when solving for the $\hat{w}$ that minimizes the training cost function, we would prefer to use the closed form solution rather than an iterative technique like gradient descent, even when the number of parameters are high.

   Circle one: True False
   Brief explanation:

   False. Using the closed form expression to solve for $\hat{w}$ requires $O(d^3)$ operations (inverse computation), while performing gradient descent requires only $O(d^2)$ (matrix multiplication cost) in every step. If the number of parameters are high, inverting a matrix may not be computationally feasible. Also, if the precision required of $\hat{w}$ isn't high, we can get by with gradient descent.

5.2 Logistic Regression

Answer the following questions with brief explanations where necessary.

1. **[2 pts]** A generalization of logistic regression to a multiclass settings involves expressing the per-class probabilities $P(y = c|\mathbf{x})$ as the softmax function $\frac{\exp(\mathbf{w}_c^T\mathbf{x})}{\sum_{d\in C}\exp(\mathbf{w}_d^T\mathbf{x})}$, where c is some class from the set of all classes C.

   Consider a 2-class problem (labels 0 or 1). Rewrite the above expression for this situation, to end up with expressions for $P(Y = 1|\mathbf{x})$ and $P(Y = 0|\mathbf{x})$ that we have already come across in class for binary logistic regression.

   $P(y = 1|x) = \frac{e^{w_1^T x}}{e^{w_0^T x} + e^{w_1^T x}} = \frac{e^{(w_1 - w_0)^T x}}{1 + e^{(w_1 - w_0)^T x}} = \frac{e^{w^T x}}{1 + e^{w^T x}} = p$ Therefore, $1 - p = \frac{1}{1 + e^{w^T x}}$.

2. **[3 pts]** Given 3 data points $(1,1),(1,0),(0,0)$ with labels $0,1,0$ respectively. Consider 2 models that compute $p(y=1|\mathbf{x})$: **Model 1:** $\sigma(w_1x_1+w_2x_2)$, **Model 2:** $\sigma(w_0+w_1x_1+w_2x_2)$ ($\sigma(z)$ is the sigmoid function $\frac{1}{1+e^{-z}}$). Using the given data, we can learn parameters $\hat{w}$ by maximizing the conditional log-likelihood.

   Suppose we switched $(0,0)$ to label 1 instead.

   Do the parameters learnt for Model 1 change?

   Circle one: True False
   One-line explanation:

   False. The parameters learnt for Model 1 don't change because $w_1x_1 + w_2x_2 = 0$ for $(0,0)$. Hence $p = 0.5$ irrespective of the labels or the values of w.

What about Model 2?

Circle one: **True** **False**
One-line explanation:

True. This model has a bias term which remains non-zero for $(0, 0)$, and can thus change the model depending on the label assigned.

3. **[2 pts]** For logistic regression, we need to resort to iterative methods such as gradient descent to compute the $\hat{w}$ that maximizes the conditional log likelihood. Why?

   There is no closed-form solution.

4. **[3 pts]** Considering a Gaussian prior, write out the MAP objective function $J_{\text{MAP}}(\mathbf{w})$ in terms of the MLE objective $J_{\text{MLE}}(\mathbf{w})$. Name the variant of logistic regression this results in.

   $J_{MAP}(\mathbf{w}) = J_{MLE}(\mathbf{w}) - \lambda||\mathbf{w}||_2^2$ This is L2 regularized logistic regression.

5. **Extra Credit: [2 pts]** For a binary logistic regression model, we predict $y = 1$, when $p(y = 1|\mathbf{x}) \geq 0.5$. Show that this is a linear classifier.

   We predict $y = 1$ when $p \geq 0.5 \leq 1 - p$. That is, when $\frac{p}{1-p} \geq 1 \implies \log\left(\frac{p}{1-p}\right) \geq 0$. The LHS is known as log-odds and this is nothing but $\log\left(e^{w^T x}\right) = w^T x$. Therefore, we predict $y = 1$ when $w^T x \geq 0$ and $y = 0$ otherwise. Therefore, logistic regression is a linear classifier.

Use this page for scratch work