Midterm, CS-GY 6923

Sample Questions

Show all of your work to receive full (and partial) credit.

Always, Sometimes, Never

Indicate whether each of the following statements is **always** true, **sometimes** true, or **never** true. Provide a one or two short justification or example to explain your choice.

1. For random events $p(x \mid y) < p(x, y)$.

   ALWAYS SOMETIMES **NEVER**

   Justification: $p(x, y) = p(x \mid y)p(y)$. Since $p(y) \leq 1$, we have $p(x, y) \leq p(x \mid y)$.

2. Consider a loss function L. If $\nabla L(\beta) = 0$, then β is a minimizer of L.

   ALWAYS **SOMETIMES** NEVER

   It could also be a maximizer, or local minimum, or saddle point. $\nabla L(\beta) = 0$ is necessarily but not sufficient for β to be a minimizer of L.

3. The empirical risk of a model is lower than the population risk.

ALWAYS **SOMETIMES** NEVER

Justification: The two are equal in expectation, but due to randomness, sometimes the empirical risk will be higher, sometimes lower.

4. The linear classifier found by logistic regression minimizes error rate (0-1 loss) on the training data.

ALWAYS SOMETIMES **NEVER**

Justification: The linear classified returned minimizes the *logisitc loss*, not the 0-1 loss.

Note: This isn't a great question because it's slighty ambigious. Maybe we get lucky and the classifier found by logisitic regression also happens to minimize error rate (this might happen e.g. when your data is perfectly linearly seperable. For this reason, I would have also accepted a solution of **SOMETIMES**)

5. Suppose we learn a linear classifier. I.e., we learn parameters β and classify an input vector x in class 1 if $\mathbb{1}[\langle x, b\rangle > \lambda]$. Increasing λ increases the recall of our classifier.

ALWAYS SOMETIMES **NEVER**

Increasing λ means that some data point classified as 1 will now be classified at 0. On the other hand, no new data points will be classified as 1. So recall must decrease.

Short Answer

4. You are trying to develop a machine learning algorithm for classifying data $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ into catagories $1, \ldots, q$. You have decided to use linear classification for the problem.

   (a) You know you can find a good linear classifier for *binary* classification (dividing into $q = 2$ classes) using logistic regression. You are considering using either the **one-vs-all** or **one-vs-one** approach to adapting this approach to the multiclass problem. In a few sort sentences describe why you might use one over the other.

   One-vs-all might be used for speed if we have a large number of classes. One-vs-one I migth use if we have a difficult dataset, and enough computation to run all $O(q^2)$ training operations.

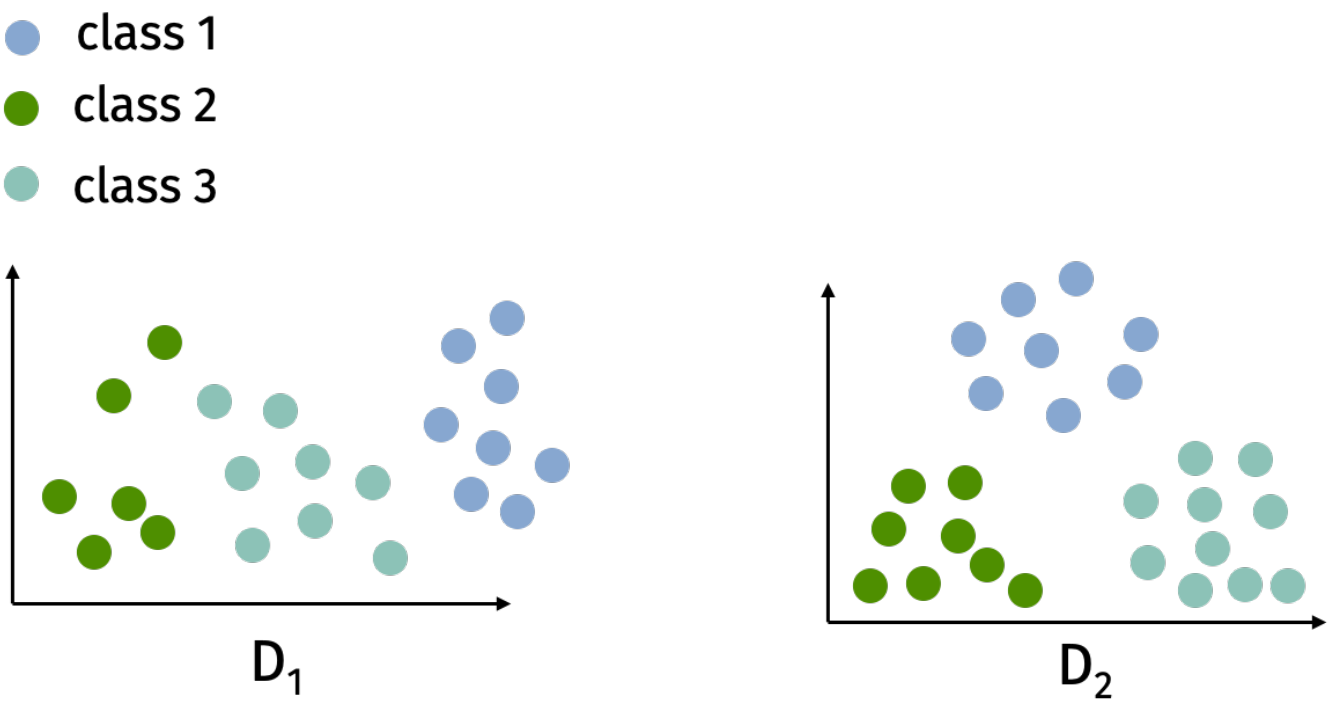
   (b) Your coworker suggests the following alternative approach: let's try to learn a parameter vector $\boldsymbol{\beta} \in \mathbb{R}^d$ and classify using the following model:

$$f_\beta(\mathbf{x}) = \begin{cases} 1 \text{ if } & \langle \boldsymbol{\beta}, \mathbf{x} \rangle \leq 1 \\ 2 \text{ if } & 2 < \langle \boldsymbol{\beta}, \mathbf{x} \rangle \leq 3 \\ 3 \text{ if } & 3 < \langle \boldsymbol{\beta}, \mathbf{x} \rangle \leq 4 \\ \vdots & \\ q-1 \text{ if } & q-2 < \langle \boldsymbol{\beta}, \mathbf{x} \rangle \leq q-1 \\ q \text{ if } & q-1 < \langle \boldsymbol{\beta}, \mathbf{x} \rangle \end{cases} \tag{1}$$

   (c) Describe **one potential issue** and **one potental benefit** of your coworker's method over the approaches mentioned in (a). There is no one "right" answer here.

Issue: *Like using a linear instead of a one hot encoding, this approach basically bakes in some assumptions about your data. E.g. that class 2 lives "between" class 1 adn 3 in a linear way*. **Benefit:** It should be very fast, both to train (depending on what loss is used) and to classify since it only uses one single parameter vector (instead of q).

(d) For the two datasets D_1 and D_2 below, indicate which of the three approaches (**one-vs-one**, **one-vs-all**, or your **coworkers approach**) would lead to an accurate solution to the multiclass classification problem. No explanation is required, but having one might help you earn partial credit.



One-vs-one: Could do both of them.

One-vs-all: Can do the second only.

Co-workers can do neither. The first because the classes aren't in the right order, and the second because you can't seperate the classes using parallel lines.

5. We are given data with just one predictor variable and one target: $(x_1, y_1), \ldots, (x_n, y_n)$, with the goal of fitting a degree two polynomial model using unregularized multiple linear regression with data transformation. The goal is to find the best coefficients $\beta_0, \beta_1, \beta_2$ for predicting y as $\beta_0 + \beta_1 x + \beta_2 x^2$.

Consider the following three transformed data matrices:

$$X_1 = \begin{bmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix}, X_2 = \begin{bmatrix} 1 & x_1^2 - x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n^2 - x_n & x_n^2 \end{bmatrix}, \text{ and } X_3 = \begin{bmatrix} 1 & 2x_1^2 - x_1 & 2x_1 - 4x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & 2x_n^2 - x_n & 2x_n - 4x_n^2 \end{bmatrix}$$

Which of the above matrices can be used to solve this problem? In other words, if we train a multiple linear regression problem with X_i can we obtain an optimal degree two polynomial fit for $y_1, \ldots, y_n$. Justify your answer in words, or with equations.

First: Could be used. Any polynomial $\beta_0 + \beta_1 x + \beta_2 x^2$ can be expressed by just choosing the parameter vector to have entries $[\beta_0, \beta_1, \beta_2]$.

Second: Could we used. Any polynomial $\beta_0 + \beta_1 x + \beta_2 x^2$ can be expressed by choising the parameter vector $[\beta_0, -\beta_1, \beta_1 + \beta_2]$.

Third: Would not work. Notice that the third column is a scaling of the second. So in particular, linear combinations of these columns can *only* express polynomials where $\beta_2 = -2\beta_1$. We could not e.g. write the polynomial $x^2 + x$ as a linear combination of the columns in this matrix.

6. Write each of the following models as transformed linear models. That is, find a parameter vector $\boldsymbol{\beta}$ in terms of the given parameters a_i and data transformation $\phi(\mathbf{x})$ such that $y = \langle \boldsymbol{\beta}, \phi(\mathbf{x}) \rangle$. Also, show how to recover the original parameters a_i from the parameters β_j:

   (a) **Example:** $y = a_1 x_1^2 + a_2 \log(a_3 x_2)$.

   Solution: Notice that $y = a_1 x_1^2 + a_2 \log(x_2) + a_2 \log(a_3)$. Let $\phi([x_1, x_2]) = [x_1^2, \log(x_2), 1]$. Set $a_1 = \beta_1$, $a_2 = \beta_2$, $a_3 = e^{\beta_3 / a_2}$.

   (b) $y = \begin{cases} a_1 + a_2 x & \text{if } x < 1 \\ a_3 + a_4 x & \text{if } x \geq 1 \end{cases}$

   Solution: Let $\phi(x) = [1, x, 0, 0,]$ if $x < 1$ and $\phi(x) = [0, 0, 1, x]$ if $x \geq 1$. Set $\alpha_1 = \beta_1, \alpha_2 = \beta_2, \alpha_3 = \beta_3, \alpha_4 = \beta_4$.

   (c) $y = (1 + a_1 x_1) e^{-x_2 + a_2}$.

   Solution: Notice that $y = e^{a_2} e^{-x_2} + e^{a_2} a_1 x_1 e^{-x_2}$. Let $\phi([x_1, x_2]) = [e^{-x_2}, x_1 e^{-x_2}]$. Set $a_2 = -\log(\beta_1)$, $a_1 = \beta_2 / a_2$\$.

   (d) $y = (a_1 x_1 + a_2 x_2) e^{-x_1 - x_2}$.

Solution: Let $\phi([x_1, x_2]) = [e^{-x_1 - x_2} x_1, e^{-x_1 - x_2} x_2]$. Set $\alpha_1 = \beta_1, \alpha_2 = \beta_2$.