Midterm 1, CS-UY 4563

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

2. You use gradient descent to find parameters $\vec{\beta}_{GD}$ for a multiple linear regression problem under ℓ_2 loss: $L(\vec{\beta}) = \|X\vec{\beta} - \vec{y}\|_2^2$. You are short on time, so you only run gradient descent for 10 iterations. Your friend finds parameters $\vec{\beta}_M$ using the equation $\vec{\beta}_M(X^T = X)^{-1}X^T y$. Is $L(\vec{\beta}_m) \leq L(\vec{\beta}_{GD})$?

   ALWAYS SOMETIMES NEVER

3. Does $\vec{\beta}_m$ acheive better population risk than $\vec{\beta}_{GD}$?

   ALWAYS SOMETIMES NEVER

4. To evaluate machine learning models, you should use a train-test split instead of k-fold cross validation.

   ALWAYS SOMETIMES NEVER

Short Answer

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

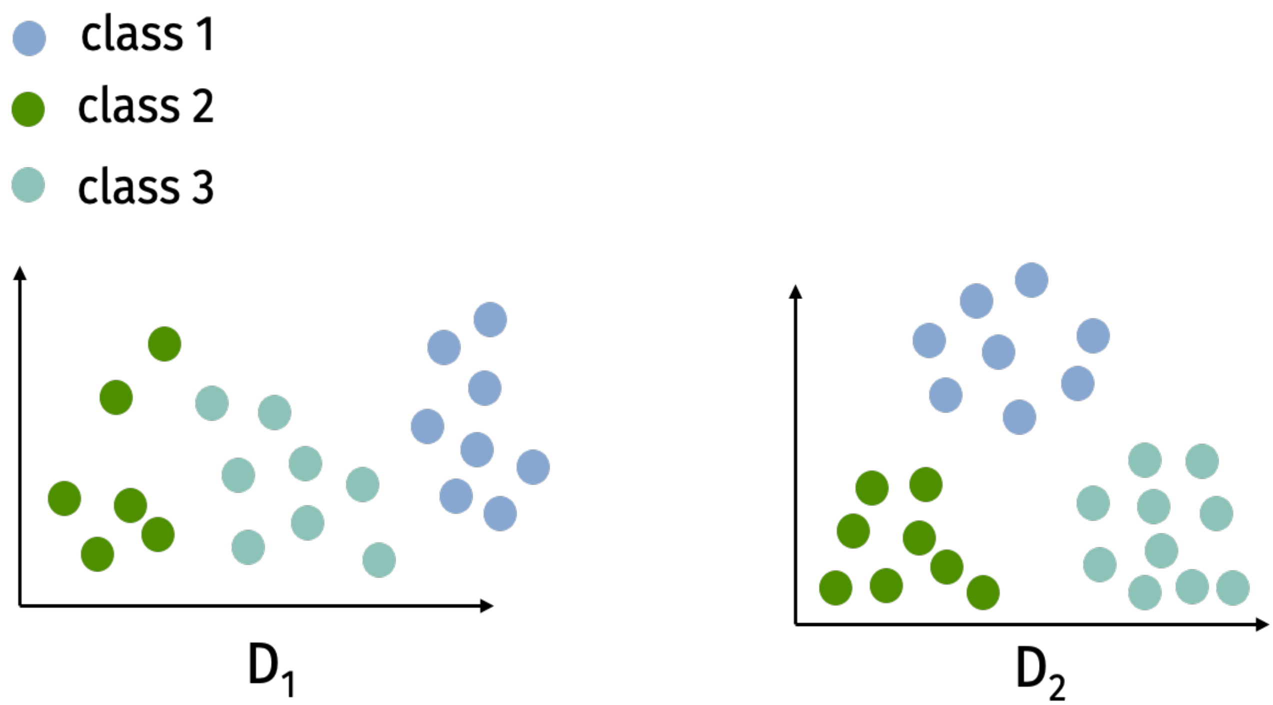
   (a) You know you can find a good linear classifier for *binary* classification (dividing into $q = 2$ classes) using logistic regression. You are considering using either the **one-vs-all** or **one-vs-one** approach to adapting this approach to the multiclass problem. In a few sort sentences describe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

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

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

(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.

(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.



2. 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:

   $X_1 = \begin{bmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix}$, $X_2 = \begin{bmatrix} 1 & x_1^2 - x_1 & x_1^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n^2 - x_n & x_n^2 \end{bmatrix}$, 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 $y_1, \dots, y_n$. Justify your answer in words, or with equations.

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

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

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

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

4. For data with one predictor variable and one target: $(x_1, y_1), \dots, (x_n, y_n)$, consider a simple linear regression model:

$$f_{\beta_0, \beta_1}(x) = \beta_0 + \beta_1 x \tag{2}$$

with a *logarithmically transformed* ℓ_2 loss:

$$L(\beta_0, \beta_1) = \sum_{i=1}^{n} (\log(y_i) - \log(f_{\beta_0, \beta_1}(x_i)))^2 \tag{3}$$

This sort of model makes sense when trying to predict a value that is more naturally expressed on a logarithmic scale (e.g. pH level, volume in decimals, etc.)

(a) Write down an expression for the gradient of the loss L.

(b) Using your expression and the gradient descent update rule, write pseudocode for finding parameters β_0^*, β_1^* which approximately minimize L.

(c) What other method could have been used to find nearly optimal β_0^*, β_1^*?