6.390 Introduction to Machine Learning

Recitation Week #5

Issued October 3, 2022

1. We work for an investment banking firm Silver Bags, and we are trying to build several predictive models about the stocks of companies.

   Companies are described to us in terms of 3 features. For each feature, describe a transformation to make a new feature vector where every element is in $\mathbb{R}$. Ultimately, we will concatenate all these new feature vectors to represent the company in a machine learning algorithm, so you should choose wisely with that goal in mind. It is totally reasonable for more than one transformation to exist, so please explain your reasoning!

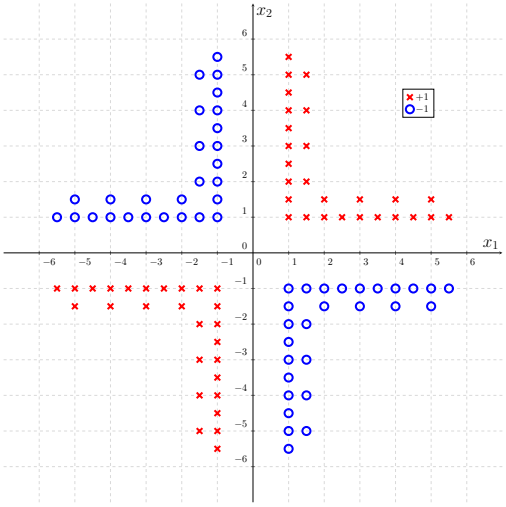
   (a) Market segment (one of "service," "natural resources," or "technology.")

   (b) Number of countries in which it operates $(1 - 50)$.

   (c) Total valuation (-1 billion to $+1$ billion).

   (d) Ava worries that having some features in a continuous range and others in a discrete range, e.g., $[0, 1]$ and $\{0, 1\}$, would mean that regularization will not work well for the regression models to be built. Ben argues that with these feature encodings, regularization will no longer be needed. Carla is most interested in analysis of the importance of different features for stock predictions, and believes that these encodings will make that difficult. You decide to proceed with these encodings and to include a ridge $\lambda||\theta||^2$ regularization term in your objective. Why, and how do you respond to Ava, Ben, and Carla?

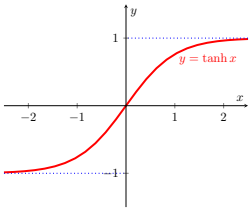
2. Consider the following dataset:



Suppose that we would like to design non-linear functions to introduce new features to create a linearly separable dataset. Out of the choices below, determine whether these choices of features would make the dataset linearly separable or not, and explain.

Hint: Consider the data points which reside in each of the four quadrants of the plot and reason what will happen to groups of points with the proposed feature transformations.

Recall that $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$:



(a) $[x_1, x_2, x_1 x_2]$

(b) $[x_1^2, x_2^2, \frac{x_1+x_2}{2}]$

(c) $[x_1 + x_2, x_1 \tanh(x_2), 1]$

(d) Come up with some other simple feature transformations that would make this data set linearly separable. What's the minimum number of (transformed) features necessary to achieve this linear separability?

3. Prof. Regu LaRisashun has just joined the 6.390 team, and they are excited to help teach students about machine learning. In particular, Prof. Regu (as they are fondly called) wants to try reducing stress by eliminating the final exam. They believe that labs and homeworks should be suffcient to predict exam performance.

   Specifically, Prof. Regu takes the homework and lab grades, $x = [x_1, x_2]^\top$, and runs a linear regression with hypothesis $\hat{y} = \theta^\top x + \theta_0$ to make predictions ($\hat{y}$) for students' midterm grades. They minimize an objective function with just mean square error between the predicted and actual midterm grades (y). Data from 70% of the students are used for training, and the remaining 30% for evaluating the model.

   The initial results do not look so good, but Prof. Regu understands that this often happens with a simple linear model, and it can help a great deal to model and encode features more thoughtfully. Prof. Regu thus writes a problem for the midterm exam, asking students to help make the final exam unnecessary, by exploring five specific ideas.

   (a) *Majors*

   Prof. Regu notices that some students find the homework questions harder than other students, and believes this could be due to what students have studied in their other classes. Specifically, Prof. Regu notices that EECS majors seem to do better on homeworks than Enology majors. Fortunately, at MIT students' majors are conveniently coded up as a number, so Prof. Regu enters this number for each student as a new feature for the model.

   Is this a good idea? Explain why or why not. If not, what better way might you encode students' majors for the model?

   (b) *Programming Experience*

   Looking more deeply, Prof. Regu notices that the Python coding homework questions seem to be strong predictors of exam grades. Prof. Regu obtains data from an initial survey students filled out at the start of the semester, where students were asked to check one box on this question:

   What is your level of Python programming experience?

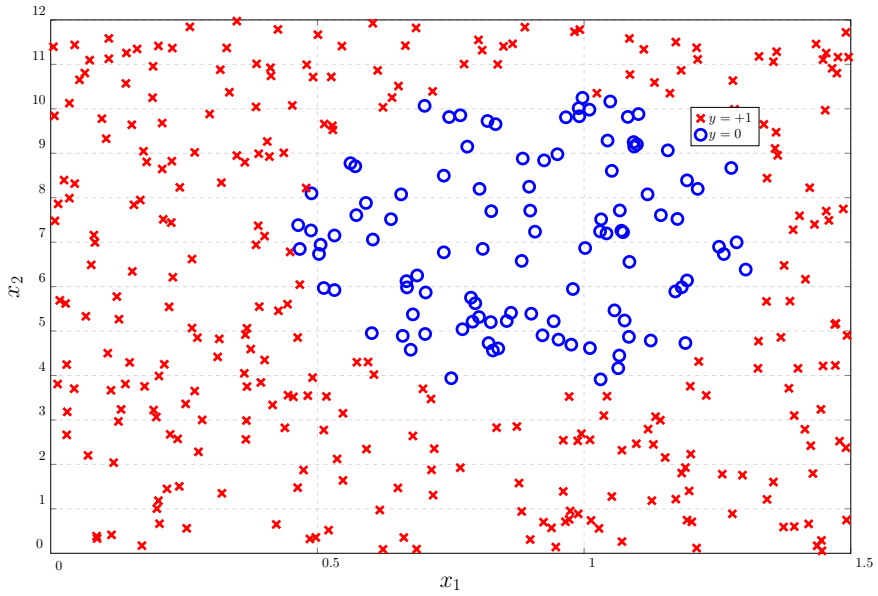
   ( ) None ( ) Beginner ( ) Experienced ( ) Expert

   How should data from this question best be encoded for Prof. Regu's model?

(c) *Time on Task*

A kind colleague at Harvard tells Prof. Regu about an interesting experiment: apparently, the Educational Testing Service is looking at the amount of time students take to answer questions, as a measure for students' understanding of the material. The idea is that a more skilled student should be able to answer questions faster than a less skilled one. Inspired by this idea, Prof. Regu mines data about how long students are taking to complete 6.390 labs (x_1) and homeworks (x_2). Prof. Regu also changes their approach: instead of predicting exam grades, Prof. Regu just tries to predict whether the student passes ($y = 1$) or fails ($y = 0$) the midterm exam based on just these x_1 and x_2 data. They employ linear logistic regression, with hypothesis $\hat{y} = \sigma(\theta^\top x + \theta_0)$.

However, this model performs poorly! Prof. Regu plots the data to try and understand why, and sees this ("×" indicates $y = 1$, and "o" indicates $y = 0$):



Apparently, while it is the case that students who take a long time on labs and homeworks indeed tend not to pass the exam, students who take a very short amount of time also tend not to pass! Prof. Regu decides to try to fix the model to accommodate this peculiar behavior, by employing a feature transform $\phi(x)$, and using the hypothesis $\hat{y} = \sigma(\theta^\top \phi(x) + \theta_0)$. Specify a mathematical function $\phi(x)$ which substantially improves the training error for these data: