Take Home Final – 4210, Fall 2024

Each problem is worth 10 points

1 Part I – theoretical problems

1. Recall standard logistic regression. The typical method of fitting the coefficient parameters β is via minimizing the negative log likelihood of the observed data (equivalent to maximizing the log likelihood); see lecture notes on this. For this problem, consider binary (two classes) logistic regression but with a regularizer/penalty term that is the same as that used in ridge regression (again, see lecture notes or the book). Recall that the ridge regression penalty term does not include the constant offset parameter β_0. Please do the following:

   (a) Write down the minimization problem described above whose solution would yield the optimal values for β. The coefficient of the penalty term in your minimization problem should be λ.

   (b) Derive a system of equations that one would need to solve in order to find the minimizer for part (a). You do not need to solve this system of equations in any way - just derive them and write them down.

   (c) For this part of the problem, consider the case where $\lambda \to \infty$. What will be the optimal values for all the β_j with $j > 0$? Find an explicit formula for the optimal value of β_0 in this case.

2. This problem will essentially mirror problem 7.9 Exercise 1 from the textbook, so please refer to it when answering. The only difference is that we will consider a quadratic spline, rather than a cubic one. So, the function we will consider is $f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 (x - \xi)_+^2$, and in parts (a) and (b) you can drop the cubic term from $f_1(x)$ and $f_2(x)$. Then simply follow 7.9 Exercise 1 with this change, answering parts (a)-(d).

3. You are given $n = 11$ data points in $p = 2$ dimensions, where each point is classified into one of $K = 3$ classes. The data is in the table below.

Point	X_1	X_2	Class
1	3	2	Red
2	2	4	Red
3	4	4	Red
4	5	4	Red
5	3	0	Blue
6	3	1	Blue
7	5	1	Blue
8	5	3	Blue
9	1	1	Green
10	1	2	Green
11	0	4	Green

This problem will explore the usage of a support vector classifier on this dataset. Specifically, consider the maximal margin classifier for this problem.

(a) Make a neat, scaled plot of the data, using different colors and/or markers for each of the three class types. You can use a computer to construct this plot.

(b) Now suppose you are going to use one-versus-one classification on this problem, as described in lecture and the textbook. Find the optimal separating hyperplane (line) for each pair of classes. Do this graphically. That is, for any given pair of classes, look at the plot and consider the family of lines that would perfectly separate the two classes. Upon some thought, I believe you will find that the unique line that also maximizes the margins will be pretty clear, due to some symmetries in the datasets. Once you determine how this line must lie graphically, you will be able to find its exact equation by noting some key points it must pass through. Be sure to explain this reasoning when you give your final answer. Draw these on your plot from (a), and give the formula for each in the form $\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$ for the appropriate values of β in each case. You must show your work and reasoning here to explain how you arrived at these formulas - simply reporting the results of a computer implementation will not suffice.

(c) Following up from part (b), indicate on your plot (via shading, for example) the regions of feature space where test points would be attributed to each of the three classes according to standard

one-versus-one classification. If you've done everything correctly, you will find a region where standard one-vs-one classification will fail to assign a class; indicate this region on your plot as well. We will refer to this region as Z below.

(d) Suppose that within Z, we offer a modified rule to assign a class to a test point. For the test point in question, compute how far it is from the closest point (not necessarily a data point, just whatever point (X_1, X_2) is closest) within each of the regions of known classification (Red, Green, and Blue), then assign it to the class whose region is closest to the test point. Using this rule, update your plot to graphically indicate within Z which subregions will be classified as each of the given classes. Be sure to explain your reasoning, but you do not have to provide any specific equations of lines, etc.

(e) Find the coordinates of the only point within Z that still could be attributed to any of the three classes. Be sure to show your work.

2 Part II – programming, 1 problem

In this problem, we will implement the matrix completion algorithm based on PCA that was discussed in lecture 21. We will use a small sample of Netflix data to explore this method. The data is included within the zip file with this exam: Netflix_Ratings.csv and Netflix_Movies.csv. The Ratings file is the main file containing many thousands of rows and three columns. Each row corresponds to a single entry in our data matrix X, with the first column being a reference to the ID number of the user (the row number in the X matrix), the second column being a reference to the ID number of the movie that was rated (the column number in the X matrix), and the third column being the actual rating that user gave to that movie (an integer from 1 to 5, 1 being a bad rating and 5 being the best). The Movies files just gives the year and title of the movies to go along with the IDs.

Your first task will be to read the ratings data into a matrix X where rows correspond to users and columns correspond to movies. This matrix will have many empty entries to begin with. All of these should be interpreted as blanks, and not as any numerical value. This leads to the first part of the problem that you must write an answer for:

(a) What movie has the lowest average rating in the raw Ratings data, and what is that rating? Which has the highest, and what is it? Which movie was rated by the largest number of users, and how many? Which was rated by the lowest number, and how many? Which user ID rated the most movies, and how many did they rate? Which user ID rated the fewest movies, and how many did they rate?

Now, implement the matrix completion algorithm as described in lecture 21. There are a few special points that you will need to do in this case that aren't mentioned in that lecture. These are:

- Before performing PCA on the augmented matrix $\tilde{X}$, you should find the mean value for each column in $\tilde{X}$; refer to the mean value for column j of $\tilde{X}$ as $\overline{x}_j$, and note that this value will be changing from one iteration to the next. Then in step 2ii of the algorithm, you will need to add $\overline{x}_j$ to what is given there (this is because PCA operates on a centered version of the data where each column has been shifted so that its mean is 0, so we need to add the mean back in).

- After finding a value for $\tilde{x}_{ij}$ via step 2ii (modified as described above), you will need to do another check on this value to see if it makes sense, and change it if it does not. Specifically, if $\tilde{x}_{ij} > 5$, then set its value to 5; and if $\tilde{x}_{ij} < 1$, then set its value to 1.

Run this algorithm using $M = 10$ for 500 full iterations. For each iteration, keep track of the value of the objective after that iteration was completed. Then:

(b) Make a plot of the value of the objective as a function of iteration number and display it. What value of the objective does the algorithm seem to settle down to?

(c) Given your final obtained $\tilde{X}$, what movie has the highest average rating, and what is the rating? What movie has the lowest average rating, and what is the rating?

(d) Make a plot that displays the average rating of each movie based off of $\tilde{X}$ versus its average rating based off of X. Considering this, which movie is the most over-rated: that movie whose difference in average rating between X and $\tilde{X}$ is largest? Which movie is the most under-rated: that movie whose difference in average rating from X to $\tilde{X}$ is smallest (most negative)?

Now run the algorithm again for 500 iterations, but this time use $M = 2$. At the end of the algorithm, consider your final 139×2 matrix $\hat{B}$, defined

such that the two columns of $\hat{B}$ are the top two principal directions for $\tilde{X}$. As stated in the notes, one can then interpret row i of $\hat{B}$ as being indicative of where movie i lies in the 2-D space we have restricted everything to. Run K-means clustering using $\hat{B}$ as your data matrix, creating 4 clusters. Then:

(e) Make a scatter plot of the data in $\hat{B}$, indicating the cluster membership of each point graphically (different colors or plot markers for the different clusters).

(f) Consider the cluster you found with the fewest members. List the movies in this cluster. Then do a little online searching, and speculate as to what these movies might have in common.