EE 559 **Midterm Assignment (Week 9)** Posted: Tues., 3/16/2021
Jenkins 11:59 PM

Due dates
Recommended to upload by: Mon., 3/22/2021, 11:59 PM PDT
Final due date: Wed., 3/24/2021, 11:59 PM PDT

Your name__________________________ **Student ID**____________

Signature____________________________ **Date**_____________

Ground rules

For this assignment, you may use any of our EE 559 course materials, and you may use your computer for coding and accessing the assigenemnt, EE 559 materials, piazza, and D2L. You may also use the internet for information about python, python coding, and python libraries that you are allowed to use for the problem you are working on (e.g., looking up information for sklearn functions that you might want to use in a problem that allows the sklearn library).

You are not allowed to use the internet to look up answers or partial answers to any of this assignment's problems, or communicate with other people (e.g., other students, people on discussion forums, etc.) on topics related to this assignment.

For all coding in this assignment, you are allowed to use built-in python functions, numpy, matplotlib (and pandas only for reading/writing csv files). Other libraries are allowed where stated so in the problem; notably, in Problem 3 you are also allowed to use sklearn functions for regression.

For questions on this assignment, when posting on piazza, please consider whether this is a question that is appropriate for all students (e.g., clarifying the problem statement, what is allowed, suspected typos or omissions in the problem statement), or only for the instructors (e.g., something that includes your approach or solution) which should be posted as a "private" post that is only visibile to the professor, TAs and you.

Please respect your classmates and follow all of these guidelines, so that everyone can be graded fairly. Any violations that are detected will result in penalties.

Uploading your solutions. Please upload the following files before the final due date:

1. A single pdf file of all your soltions/answers.
2. A single computer-readable pdf file of all your code.
3. A csv file of each of your optimized systems' predictions on the "unknowns" dataset, as described in Problem 3. (2 csv files for Dataset1)

Problems and points. This Midterm Assignment has 3 problems, worth a total of 100 points possible. There will be partial credit on most problem parts. Good luck!

Problem 1. *Complexity of OvO, OvR, MVM.* (25 points)

Consider a *C*-class classification problem with *D* original features. We will be using perceptron and MSE learning algorithms (or similar algorithms) to train the classifiers.

We are interested in $C \geq 3$ classes.

(a) For this part, assume the classifiers are based on linear discriminant functions $g_k(\underline{x})$ or $g_{kl}(\underline{x})$.

(i) How many degrees of freedom (d.o.f.) are there if we use the one vs. rest method?

(ii) How many d.o.f. if we use the one vs. one method?

(iii) How many d.o.f. if we use the maximal value method?

For parts (b)-(d) below, suppose we use a nonlinear mapping $\underline{u} = \underline{\phi}(\underline{x})$ to an expanded feature space $\underline{u}$, then apply OvR, OvO, or MVM, in the $\underline{u}$-space. For parts (b) and (c), give all your answers in terms of *C* and *D*.

(b) Let $\underline{\phi}(\underline{x})$ be a polynomial of degree $M = 2$ (quadratic) in $\underline{x}$. Assume $D \geq 2$.

(i) What is the dimensionality of (augmented) $\underline{u}$-space?

(ii) How many degrees of freedom (d.o.f.) are there if we use the one vs. rest method?

(iii) How many d.o.f. if we use the one vs. one method?

(iv) How many d.o.f. if we use the maximal value method?

(c) Let $\underline{\phi}(\underline{x})$ be a polynomial of degree $M = 3$ in $\underline{x}$. Assume $D \geq 2$.

(i) What is the dimensionality of (augmented) $\underline{u}$-space?

(ii) How many degrees of freedom (d.o.f.) are there if we use the one vs. rest method?

(iii) How many d.o.f. if we use the one vs. one method?

(iv) How many d.o.f. if we use the maximal value method?

(d) Suppose you are given a machine learning problem with $C = 100$ classes and $D = 20$ original features. There are N_{Tr} training data points. Assume no regularizer will be used. Let's take our rule of thumb on balancing constraints with d.o.f. to be: $N_c \geq 5(\text{d.o.f.})$. For each method given below, approximately how many training data points N_{Tr} do we need to have sufficient constraints so that we are reasonably likely to avoiding overfitting?

(i) OvR with $M = 0$? $M = 1$? $M = 2$? $M = 3$?

(ii) OvO with $M = 0$? $M = 1$? $M = 2$? $M = 3$?

(iii) MVM with $M = 0$? $M = 1$? $M = 2$? $M = 3$?

(e) To get an overall perspective on the results in (a)-(c) above, put together a table with methods across the top (column labels OvR, OvO, and MVM), and polynomial degrees $M = 0, 1, 2, 3$ across the left (row labels). For the table entries, give the d.o.f. in terms of their asymptotic (big-*O*) bounds (in simplest form, tightest upper bound). Note that you are not asked to give computational time or space complexity; you are asked to give the d.o.f. in terms of asymptotic bounds.

Problem 2. *Criterion function and optimization.* (35 points)

This problem uses augmented notation.

In a 2-class classification problem, let the criterion function be:

$$J\left(\underline{w}\right)=\left(\frac{1}{N}\sum_{n=1}^{N}\left(2\left[\!\left[\underline{w}^{T}z_{n}\underline{x}_{n}\le 0\right]\!\right]-1\right)\left(\underline{w}^{T}\underline{x}_{n}\right)^{2}\right)+C\left(\left\|\underline{w}\right\|_{2}^{2}-1\right)^{2},\ C\ge 0$$

in which *C* is a parameter (not the number of classes).

(a) Interpret $J\left(\underline{w}\right)$ geometrically (using a feature-space or weight-space plot), and in words. Better interpretations get more credit.

(b) Consider the second term of $J\left(\underline{w}\right)$: $C\left(\left\|\underline{w}\right\|_{2}^{2}-1\right)^{2}$, $C\ge 0$. Is this term convex? Strictly convex? Prove your answers.

(c) Express $J\left(\underline{w}\right)$ in matrix-vector form (no summations).

Hints:

(1) Let $\hat{\underline{\underline{Z}}}=\operatorname{diag}\left\{\hat{z}_{1},\hat{z}_{2},\cdots\hat{z}_{N}\right\}$, $\hat{z}_{n}\triangleq\left(2\left[\!\left[\underline{w}^{T}z_{n}\underline{x}_{n}\le 0\right]\!\right]-1\right)$;

also $\underline{\underline{X}}=$ data matrix of augmented data points;

(2) If you get stuck, try writing out $\underline{\underline{X}}\underline{w}$ in terms of $\underline{w}$ and $\underline{x}_{n}$.

(d) Suppose we want to use batch (true) gradient descent to find $\hat{\underline{w}}$. Give an expression for the weight update $\Delta\underline{w}\left(i\right)=\underline{w}\left(i+1\right)-\underline{w}\left(i\right)$; express in simplest form.

Hint: for taking derivatives, you may treat $\hat{\underline{\underline{Z}}}$ as a constant of $\underline{w}$, for small (infinitessimal) changes in $\underline{w}$.

(e) Suppose instead we want to use stochastic gradient descent (variant 1) to find $\hat{\underline{w}}$. Answer the following:

(i) Express $J\left(\underline{w}\right)$ in the form $J\left(\underline{w}\right)=\sum_{n=1}^{N}J_{n}\left(\underline{w}\right)$, and give a (simplest form) expression for $J_{n}\left(\underline{w}\right)$.

Hint: Let $C\left(\left\|\underline{w}\right\|_{2}^{2}-1\right)^{2}=\sum_{n=1}^{N}\frac{C}{N}\left(\left\|\underline{w}\right\|_{2}^{2}-1\right)^{2}$

(ii) Derive and state (in simplest form) the weight-update formula $\Delta \underline{w}(i) = \underline{w}(i+1) - \underline{w}(i)$.

Hint: you may treat $[\![\underline{w}^T z_n \underline{x}_n \leq 0]\!]$ as a constant of $\underline{w}$, for small (infinitessimal) changes in $\underline{w}$.

(iii) Give a full statement of the stochastic GD (variant 1) algorithm, showing a sequence of steps, written in words and algebraic equations (pseudocode, not actual code). Omit any halting conditions.

(f) Consider a problem that has $D = 1$ feature. You are given the following training dataset consisting of 4 augmented unreflected data points:

$$S_1 : \ \underline{x}_1 = (1, -2)^T, \ \underline{x}_2 = (1, 0)^T$$

$$S_2 : \ \underline{x}_3 = (1, 3)^T, \ \underline{x}_4 = (1, 4)^T$$

(i) Plot the data on an (augmented) feature-space plot, and on a weight-space plot (as lines on the weight-space plot). Include small arrows to indicate positive sides of the lines on the weight-space plot.

From your result of (e), applied to this problem and dataset (again, assume there are no halting conditions):

(ii) Suppose at iteration *i* we have:

$$\underline{w}(i) = (2, -1)^T$$

and the data point for this iteration is $\underline{x}_3$. Give the weight-update $\Delta \underline{w}(i)$ numerically, and the resulting weight vector $\underline{w}(i+1)$ numerically.

On your plot of the data (in either feature space or weight space, your choice; best to be consistent with your choice in part (a)), show the following:

$\underline{w}(i)$ and if feature space then also the decision boundary $H_{\underline{w}(i)}$;

the weight-update vector $\Delta \underline{w}(i)$ and the resulting weight vector $\underline{w}(i+1)$;

the new decision boundary $H_{\underline{w}(i+1)}$ if using a feature-space plot. Decision boundaries should include a small arrow showing the positive (g > 0) side.

(iii) Separately, repeat part (ii) for the case of:

$$\underline{w}(i) = (-1, -1)^T$$

and the data point for this iteration is $\underline{x}_2$. Use a new plot to avoid clutter.

(g) Are the weight updates of part (f) consistent with your interpretation of $J(\underline{w})$ in part (a)? Please explain.

Problem 3. *MSE and Ridge Regression.* (40 points)

In this problem you will apply mean-squared error (MSE) and ridge regression to a given synthetic dataset. The dataset has 2 features.

Dataset1 has been divided up as follows:

dataset1_train.csv

dataset1_val.csv

dataset1_test.csv

dataset1_unknowns.csv

In each of the train, val, and test dataset files, there are 3 columns which give x_1, x_2, y. In the unknowns dataset file, there are only 2 columns, which give x_1, x_2.

There may be another dataaset that will also be posted; if so, you will repeat the problem on that dataset, which will be similar (the only difference will be the data-point values and the number of data points).

This problem uses augmented notation. (In the dataset files, the data is not augmented.)

You may use regression functions from sklearn in this problem.

Tips: (1) This problem uses Validation and Model Selection as described in Lecture 15. (No need to use cross-validation.)

(2) Use the test set only where instructed to.

(a) As a baseline for comparison, compute the mean-squared error (MSE) of a trivial system that always outputs the mean value $\overline{y}$ (calculated from the training data), in which:

$$\overline{y} = \frac{1}{N_{Tr}} \sum_{n=1}^{N_{Tr}} y_n$$

Report on the MSE of this trivial system on the training set, validation set, and test set.

For parts (b)-(e) below, use MSE (least-squares) regression without regularization.

For weight values you report, please limit the number of digits reported for each weight to 3 decimal places; e.g., yy.xxx.

Also, please make sure that on all plots, both axes are clearly labeled with tick marks and their values.

(b) Run MSE regression on the training data, and report on the resulting MSE on the training set and on the validation set. Also give the final (optimal) weight values.

(c) Try nonlinear MSE regression, using a transformation of feature space $\underline{u} = \underline{\phi}(\underline{x})$ that is a polynomial of degree *M*. Use model selection to optimize *M*, with the provided training and validation sets.

Report on your optimal value of *M*, the MSE on training and validation sets, and the final weight values. For the final weight values, be sure to show which weights correspond to which terms of the polynomial.

(d) For visualization, plot the following, on a y vs. x_j plot, for your optimal linear model from (b), and for your optimal nonlinear model chosen in (c):

(i) For $j=1$, plot all the training data for which $-0.1 \le x_2 \le 0.1$, on a y vs. x_1 plot. Also plot your prediction curve, $\hat{y}(\underline{x}) = \hat{y}(x_1, x_2)$ vs. x_1, over the domain $-1 \le x_1 \le 1$, with $x_2 = 0$. Do this for your result of (b) and your result of (c).

(ii) For $j=2$, plot all the training data for which $-0.1 \le x_1 \le 0.1$, on a y vs. x_2 plot. Also plot your prediction curve, $\hat{y}(\underline{x}) = \hat{y}(x_1, x_2)$ vs. x_2, over the domain $-1 \le x_2 \le 1$, with $x_1 = 0$. Do this for your result of (b) and your result of (c).

Tips: 1. You may find this link helpful:
https://matplotlib.org/stable/gallery/subplots_axes_and_figures/subplot.html

2. These plots are not used in parts below; so it's not required to do them before solving parts (e)-(h). If you have any issues in creating the plots, do not hesitate to consult the internet on python or matpotlib plotting functions and examples, or to use coures help (Piazza, office hours, etc.) to address your issue (always respecting the guidelines laid out at the beginning of the assignment).

(e) For your best system (using your optimal *M*):

(i) Run the system on the test set; report on the MSE.

(ii) Run the system to predict output values $y(\underline{x}_m)$ on the unknown dataset with its given inputs $\underline{x}_m$, $m = 1, 2, \cdots N_{un}$. Output a csv file with one column that gives the predicted values $y(\underline{x}_m)$, $m = 1, 2, \cdots, N_{un}$, in the same order as given in the unknown dataset.

For parts (f)-(h) below, use Ridge Regression.

(f) Run Ridge Regression on the training data, and use model selection to find the best values M^*, λ^* of the parameters *M* and λ.

Tips:

1. Use two nested loops for varying *M* and λ.

2. While we don't know which values of λ are best to try, it is common to try values spanning a few orders of magnitude, from much less than 1, to somewhat greater than 1 (e.g., spacing values of λ on a log scale rather than linear).

(i) Report on the MSE on the training set and on the validation set, for baseline values of $M=1$ and $\lambda=0$, as well as for your optimal values M^* and λ^*. For your system that uses the optimal values M^* and λ^*, also report the final weight values, and show which weights correspond to which terms of the polynomial.

(ii) Show on a validation-set-MSE vs. λ plot, a curve for each value of *M* that you tried, each curve labeled accroding to its value of *M*. Show your choice of best values M^* and λ^*. If it's too crowded to see clearly, or if you have trouble plotting it this way, then please use separate plots for different values of *M*.

Tip: consider using matplotlib.pyplot.plot, .xticks, and .show.

(g) If your optinal system from part (f)(i) is significantly different than your final system of part (c), then repeat the plots of (d)(i)-(ii), except for your optimal sysem from part (f)(i). (The systems are not significantly different if they use the same M^*, have weight values that are close to the same, and have MSE that are very close to the same.)

(h) For your best system (using M^*, λ^*):

(i) Run the system on the test set; report on the MSE.

(ii) Run the system to predict output values $y\left(\underline{x}_m\right)$ on the unknown dataset with its given inputs $\underline{x}_m$, $m=1,2,\cdots N_{un}$. Output a csv file with one column that gives the predicted values $y\left(\underline{x}_m\right)$, $m=1,2,\cdots,N_{un}$, in the same order as given in the unknown dataset.