EE 559 | **Midterm Assignment** | Posted: Mon., 3/7/2022
Jenkins | **(Week 9)** | Due: Fri., 3/11/2022, 11:59 PM

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 assignment, 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.

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 visible only 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 substantial penalties.

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

1. A single pdf file of all your solutions/answers.
2. A single computer-readable pdf file of all your code.

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!

1. **Kernel nearest means.**

In this problem you will write code and run a nonlinear version of a nearest means classifier, on data that we provide. This problem has $D = 2$ features.

Coding: you must use Python, and code the functions yourself. sklearn and other supplementary libraries are not allowed for this problem. Similar to Homework 4, you may use Python built-in functions, NumPy, and matplotlib. You may also use plotting functions provided in our previous homework assignments, and modify them as you like. Pandas may be used only for reading and writing csv files.

Datasets: there are 2 synthetic datasets provided. Pr1_dataset1 is the synthetic dataset from HW1, but divided up to include a validation set. Pr1_dataset2 is also synthetic but is new. Each of these 2 datasets is given as 3 sets: training set, validation set, and test set.

Comment: This problem uses model selection with a validation set (covered in Lecture 13); the procedure is straightforward, and we will answer any questions you have regarding the model selection procedure.

(a) Write an expression for the discriminant function $g(\underline{x})$ for a 2-class nearest-means classifier. No need to derive it; but do express it in simplest form.

(b) From (a), develop and give the dual representation of the discriminant function; call it $g_D(\underline{x})$.

Hint: Lecture 12 for the multiclass (MVM) version.

(c) Use kernel substitution to give an expression for the discriminant function $g_{RBF}(\underline{x})$ for a nearest-means classifier using an RBF kernel, with parameter γ (as defined in Lecture 12).

(d) Code a 2-class kernel nearest means classifier that can use RBF kernel or linear kernel. Use model selection with the validation set to find an optimal value of γ (call it γ^*) in the case of RBF kernel, separately for each dataset.

Tip: not knowing what range is best for γ a priori, you might first try a range covering a few orders of magnitude, and use a log scale for the increment: e.g., to cover $0.01 \le \gamma \le 100$, one could use a loop over the index k in the range $-2 \le k \le 2$ with some increment (step size) specified for k. Then the value of γ for each k can be computed from $\gamma = 10^k$.

(e) Plot the validation-set classification error as a function of γ, for each dataset, for RBF kernel. (If you used a log scale for your increments as described in the tip of (d), then use a log scale for γ in your plot; or equivalently, plot vs. k instead of vs. γ.) Pick the optimal value γ^* for each dataset based on validation-set classification error.

(f) Compare test-set error using the linear kernel with test-set error using the RBF kernel, for each dataset. Comment on the results.

(g) For the linear kernel, plot the training data, decision regions and boundary, in the feature space, for each dataset.

(h) For the RBF kernel with optimal γ, plot the training data and decision regions in the original feature space, for each dataset.

Tip: any point in the original feature space can be classified (using nearest-means with RBF kernel) from your code.

(i) For the RBF kernel, repeat part (h) except for $\gamma = 0.01\gamma^*,\ 0.1\gamma^*,\ 0.3\gamma^*,\ 3\gamma^*,\ 10\gamma^*,\ 100\gamma^*$. Comment on the results.

2. **Learning algorithm from criterion function**.

In this problem you will derive and interpret a learning algorithm based on the following criterion function in a 2-class problem, using augmented notation:

$$J(\underline{w}) = -\sum_{n=1}^{N} \left[\left[g(z_n \underline{x}_n) \le 0 \right]\right] \frac{\underline{w}^T z_n \underline{x}_n}{\| \underline{w} \|_2}$$

in which $[[\ldots]]$ denotes the indicator function.

(a) Interpret the meaning of the criterion function; use a figure if it helps explain it.

Tip: just stating what is different from Perceptron criterion function is not sufficient.

(b) Derive a gradient descent (GD) algorithm for each GD method given below. Simplify your answers algebraically. Give a statement of the entire algorithm.

Give both:

(i) Batch GD

(ii) Stochastic GD (Variant 1)

(c) In this part use nonaugmented notation. Consider a problem with $D = 2$ features. Draw a plot in 2D weight space (w_2 *vs.* w_1), for the case $w_0 = 0$, showing items (i)-(v) below. Label your axes with numbers and tick marks, so the lines and vectors drawn are unambiguous.

Let $\eta(i) = \frac{1}{1+i}$. For the first epoch, the data has already been shuffled so take the data points in the order given ($\underline{x}_1,\ \underline{x}_2,\ \underline{x}_3,\ \underline{x}_4$).

Tip: This plot can be done by hand. If you prefer to do it by computer, that is also OK.

(i) The following data points (as lines $g_{z_n \underline{x}_n}(\underline{w}) = 0$), with an arrow pointing to the correctly classified side.

$$S_1:\ \underline{x}_1 = [-4,1]^T,\ \underline{x}_2 = [-3,-2]^T$$
$$S_2:\ \underline{x}_3 = [-1,3]^T,\ \underline{x}_4 = [3,-2]^T$$

(ii) The weight vector at the second iteration ($i = 1$) :

$$\underline{w}(1) = [-1,4]^T.$$

(iii) The vector showing the weight update at iteration $i = 1$ (based on data point $\underline{x}_2$). Also state the vector's value in numbers.

(iv) The updated weight vector $\underline{w}(2)$. Also give the vector's value in numbers.

(v) The solution region; if there is no solution region, state so.

(d) This part also uses nonaugmented notation. For the scalar case, $\underline{w} = w$, $\underline{x}_n = x_n$, with $w_0 = 0$: is the criterion function convex? Prove your answer.

3. **Nonlinear MSE regression – 2 approaches.**

In this problem you will compare two approaches, to using a degree 3 polynomial nonlinearity in a regression problem. Assume there is some reason to believe that a degree 3 polynomial is a good model for the data. This problem has $D = 3$ (original) features, and the training dataset has N data points.

Please use **augmented notation** throughout this problem.

Comment: many (but not all) of the parts below have short answers, so the problem is shorter than it probably looks.

Approach A1.

We use a degree 3 polynomial mapping from the original feature space ($\underline{x}$ space) to the expanded feature space ($\underline{u}$ space); then use a linear MSE regressor in $\underline{u}$ space. Use $\underline{w}'$ to denote the weight vector in $\underline{u}$ space (as in lecture). The nonlinear mapping uses the most general form of a third degree polynomial of the vector $\underline{x}$, for the case of $D = 3$ features (that is, it includes all terms of the third degree polynomial).

Approach A2.

This is different than what we have done in class. We use a linear MSE regressor in the original feature space to obtain the regression function $\hat{f}(\underline{x}) = \underline{w}^T\underline{x}$, and then we apply a degree 3 polynomial to its output value $\hat{f}(\underline{x})$, to obtain the final output of the nonlinear regressor, $\hat{y}(\underline{x})$, such that:

$$\hat{y}(\underline{x}) = w_1'\left(\hat{f}(\underline{x})\right) + w_2'\left(\hat{f}(\underline{x})\right)^2 + w_3'\left(\hat{f}(\underline{x})\right)^3$$

Note that here $\underline{w}'$ is defined differently than in approach A.1, and that there is no w_0'. In this approach all the weights ($\underline{w}$ and $\underline{w}'$) are learned from the data during the learning procedure.

Answer the following questions.

(a) Give the MSE criterion function $J_{A.1}(\underline{w}')$ for approach A.1. You may write this in terms of $\underline{u}$, if you also define the vector $\underline{u}$ (e.g., by writing out its components in terms of the components of $\underline{x}$). State how many components there are in the vector $\underline{w}'$.

(b) How many d.o.f. are there during learning using Approach A.1? Briefly justify your answer.

(c) Give a linear-algebra solution to the optimal weight vector $\hat{\underline{w}}'$ for approach A.1. No need to derive it. Be sure to define all variables and quantities in your solution, that aren't already defined in this problem.

(d) Give the MSE criterion function $J_{A.2}(\underline{w}, \underline{w}')$ for approach A.2. You may write it in terms of $\hat{y}(\underline{x})$ if you also define all terms used in the expression for $\hat{y}(\underline{x})$, including writing out your expression for $\hat{f}(\underline{x})$.

(e) How many d.o.f. are there during learning using Approach A.2? Briefly justify your answer.

(f) In Approach A.2, is $\hat{y}(\underline{x})$ a linear or nonlinear function of $\underline{x}$? Is $\hat{y}(\underline{x})$ a linear or nonlinear function of $\underline{w}$? Is $\hat{y}(\underline{x})$ a linear or nonlinear function of $\underline{w}'$? (No need to prove or justify your answers.)

(g) Is the criterion function of (d) a continuous, differentiable function of $\underline{w}$? Is it a continuous, differentiable function of $\underline{w}'$? (No need to prove or justify your answers.)

(h) Derive the weight-update formulas for $\underline{w}$ and $\underline{w}'$ for Approach A.2, for sequential GD, if weight update formulas exist. Use $\eta(i)$ for the learning rate parameter for $\underline{w}$ update, and $\rho(i)$ for the learning rate parameter for $\underline{w}'$ update.

If the weight update formulas don't exist, justify why not.

(i) Give the criterion function for Approach A.1 that also includes l_2 regularization of all the weights.

(j) Give the criterion function for Approach A.2 that also includes l_2 regularization of all the weights. Use regularization parameter λ for $\underline{w}$, and regularization parameter λ' for $\underline{w}'$. Be sure to define all quantities you use.

(k) Which approach is likely to give lower error on unknowns if there is plenty of data? Briefly justify.

(l) Which approach is likely to give lower error on unknowns if there is a very limited amount of data? Briefly justify.