EE 559 | **Homework 5** | Posted: Thur., 2/22/2024
Jenkins | v1.2 update in yellow | Due: Fri. 3/1/2024, 11:59 PM PST

Problems created by Keith Jenkins and Ziageng Zhu.

Note: because of the upcoming midterm, this homework will only be accepted up to 24 hours late (must be uploaded by Sat., 3/2/2024, 11:59 PM PST, at the cost of 1 late day). Solutions will be posted Sunday 3/3.

Reminder: for this and all remaining homework assignments in this class, you are required to use Python for computer problems.

1. Code a Polynomial Feature Mapping and Classification problem. For this problem, some of the code you will write yourself and you are allowed to use sk-learn package where stated.

   Code outline is provided in file `HW5_for_students_2.ipynb` which also includes preprocessing code. Use of the code outline is recommended but not required.

   Use `dataset3_train` and `dataset3_test` from HW1 for your train and test sets.

   Standardize the data (**tip**: use `sklearn.preprocessing.StandardScaler`) before doing the polynomial mapping.

   (a) For each dataset, do the following.

   (i) Implement polynomial feature mappings for those two datasets. You will expand the original 2D feature space ($\underline{x}$) into higher-dimensional space ($\underline{u}$) using the following polynomial orders:

   - **Order 1-3**: **Manually code** these mappings. Ensure the data is augmented to include bias term.
   - **Order 4-7, 10, 11, 15**: Use **sk-learn function** `PolynomialFeatures` for these mappings.

   (ii) Use **sk-learn** `Perceptron` classifier to train in expanded feature space ($\underline{u}$).

   (iii) Evaluate and report the **training and testing classification accuracy** for each polynomial order.

   (iv) Plot the resulting **decision regions and training data points** in the original 2D space ($\underline{x}$) for each polynomial order.

   Tip: modify `plotDecisionBoundary()` function used in HW1.

   (b) Compare the train and test accuracy, and decision regions, for each polynomial order listed above. Comment on your observations.

   (c) Report a 21 x 3 table which shows columns for the polynomial degree, degrees-of-freedom (d.o.f), and number of constraints (number of data points), for each polynomial degree (from 1 to 20). By considering the table and your results from (a)(b), for each dataset, discuss where you think overfitting happened and why.

2. This problem is to be done by hand; however you may use a computer for (d) if you would prefer to. For a machine learning problem with *D* features, you will perform a nonlinear degree-*p* polynomial transformation of the data before the classification or regression step. Derive how many degrees of freedom (d.o.f.) provided by the weight variables, there will be for:

   (a) p = 1 (no derivation needed; just a brief justification)

   (b) p = 2 (no derivation needed; you can cite the derivation in lecture notes)

   (c) p = 3 (derivation required)

   Tip for (a)-(c): you can check your answers for $D = 2$ by comparing with Pr. 1 above.

   (d) Using your formulas from (a)-(c), produce a table showing the exact numerical value of d.o.f. for each of $p = 1, 2, 3$, for the following number of (original) features *D* (by hand (e.g., calculator) or by computer using basic python): $\mathrm{D} = 1,\ 10,\ 100,\ 1000$.

3. This problem is to be done by hand except where stated otherwise. You are given the following data in a 2-class, 2-feature problem, in unaugmented feature space:

$$S_1\text{: } (-2,-1),\ (0,3),\ (2,1)$$

$$S_2\text{: } (3,-2),\ (3,-4),\ (6,-1)$$

   (a) Plot the data in 2D unaugmented feature space.

   (b) Augment the data. Then calculate the optimal weight vector $\hat{\underline{w}}$ using the pseudoinverse solution, and calculate its norm $\left\|\hat{\underline{w}}\right\|_2$, by hand or by computer. You will need to invert a 3×3 matrix as part of the pseudoinverse calculation. (You may use `numpy.linalg.pinv()` to calculate the pseudoinverse, and then other numpy or python functions to calculate $\hat{\underline{w}}$ and $\left\|\hat{\underline{w}}\right\|_2$.)

   For the target values use $b_n = 1\ \forall n$ if using reflected data; or use $b_n = y_n$ if using unreflected data, in which $y_n \in \{-1, +1\}$ is the given class label.

   Report numerical values for $\hat{\underline{w}}$ and its norm $\left\|\hat{\underline{w}}\right\|_2$.

   (c) Plot the target lines H_T on your plot (by hand). Use 2D unaugmented feature space with unreflected data for your plot. Note that because the data is unreflected, there will be two target lines H_T, each defined as:

$$H_T^{(1)} \text{ (for class } S_1 \text{ data points): } g(\underline{x}) = b^{(1)} = 1$$

$$H_T^{(2)} \text{ (for class } S_2 \text{ data points): } \mathrm{g}(\underline{x}) = b^{(2)} = -1$$

   Hint: First write the equation for each target line, using $g(\underline{x})$ and the optimal weight vector $\hat{\underline{w}}$ from (b).

   (d) Would either target line H_T make a good decision boundary? Briefly justify.

   (e) Plot the actual decision boundary H_B determined by $\hat{\underline{w}}$ of part (b). Draw a small arrow indicating the positive (Γ_1 side) of H_B.

   Hint: This is a linear 2-class problem, so decision boundary is defined by $\mathrm{g}(\underline{x}) = 0$.

   (f) Is H_B a reasonable decision boundary? Briefly justify.