CSCI 5525: Machine Learning (Fall 2020)
Homework 2
Due 10/15/2020 11:59 PM CDT

1. **(20 points)** Recall that a function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a valid kernel function if it is symmetric and positive semi-definite function. For the current problem, we assume that the domain $\mathcal{X} = \mathbb{R}$.

   (a) **(10 points)** Let $K_1, \ldots, K_m$ be a set of valid kernel functions. Show that for any $w_j \geq 0, j = 1, \ldots, m$, $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^p$ that the function $K(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{m} w_j K_j(\mathbf{x}, \mathbf{x}')$ is a valid kernel function.

   (b) **(10 points)** Consider the function $K(\mathbf{x}, \mathbf{x}') = K_1(\mathbf{x}, \mathbf{x}') + K_2(\mathbf{x}, \mathbf{x}')$ where K_1 and K_2 are valid kernel functions. Show that K is a valid kernel function.

2. **(20 points)** The SVM classifier can be implemented in either primal or dual form. In this problem, implement a linear SVM in dual form using slack variables. Note, this will be a quadratic program with linear constraints. For this you need an optimizer. Use the optimizer `cvxopt` which can be installed in your environment either through pip or conda. Refer to the `cvxopt` document for more details about quadratic programming: `https://cvxopt.org/userguide/coneprog.html#quadratic-programming`.

   Apply your SVM to the dataset "hw2_data_2020.csv". This dataset consists of samples from 2 classes making this a 2-class classification problem. The rows are the samples and the first p columns are the features and the last column is the label. Split this dataset into 80% train data and 20% test data. Apply $k = 10$ fold cross validation on the train data to choose the optimal value of C (see (a) below).

   Please submit (a) **summary of methods and results** report and (b) **code**:

   (a) **Summary of methods and results**: Briefly describe the approaches used above, along with relevant equations. Also, calculate the train and validation error rates over the 10 folds for each value of $C = \{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10, 100, 1000\}$. Report the average train error rate and its associated standard deviation (over the 10 train error rates) and the average validation error rate and its associated standard deviation (over the 10 validation error rates). After running cross validation, choose the value of C which gives the lowest average validation error. Apply the learned model with that value of C to the held out test set and report the error rate on the test set (1 number). Make sure to explain why you chose that value of C beyond that it has the lowest validation error rate.

   (b) **Code**: Submit the file `SVM_dual.py` which contains the function `def SVM_dual(dataset: str) -> None:` where `dataset` is a string consisting of the name of the dataset and the function does not return anything but must print out to the terminal (stdout) the average train and validation error rates and standard deviations, the optimal value of C, and the test set error rate for the model with the lowest validation error rate.

3. **(30 points)** In this problem, we consider Kernel SVM. Implement a Kernel SVM for a generic kernel. Apply your Kernel SVM to the dataset "hw2_data_2020.csv". Split the dataset in the same way as in Problem 2 (80% train, 20% test) and apply $k = 10$ fold cross validation on the train data to choose to optimal hyperparameters (you must decide on reasonable hyperparameter ranges) for the following kernels:

   (i) Linear kernel,

   (ii) RBF kernel.

   Please submit (a) **summary of methods and results** report and (b) **code**:

   (a) **Summary of methods and results**: Briefly describe the approaches used above, along with relevant equations. Also, for both (i) and (ii), report the average train and validation error rates and standard deviations (over the 10 folds) for each combination of the hyperparameter values (you choose the values to experiment with - they must be reasonable and you must be able to explain why they are reasonable). After running cross validation, choose the optimal hyperparameter values and apply the learned model with those values to the held out test set and report the error rate on the test set. Make sure to explain why you chose those hyperparameter values.

   (b) **Code**: Submit the file `kernel_SVM.py` which contains the function `def kernel_SVM(dataset: str) -> None:` where `dataset` is a string consisting of the name of the dataset and the function does not return anything but must print out to the terminal (stdout) the average train and validation error rates and standard deviations, the optimal hyperparameter values, and the test set error rate for the best model.

4. **(30 points)** In this problem, we consider multi-class classification using SVM. Implement a multi-class SVM using the one vs all strategy. Apply your SVM to the "mfeat" dataset[1] which contains descriptors from MNIST for reducing the data dimensionality.

   Split the dataset in the same way as in Problem 2 (80% train, 20% test) and apply $k = 10$ fold cross validation on the train data to choose to optimal hyperparameters (you must decide on reasonable hyperparameter ranges) for the following kernels:

   (i) Linear kernel,

   (ii) RBF kernel.

   Please submit (a) **summary of methods and results** report and (b) **code**:

   (a) **Summary of methods and results**: Briefly describe the approaches used above, along with relevant equations. Also, for both (i) and (ii), report the average train and validation error rates and standard deviations (over the 10 folds) for each combination of the hyperparameter values (you choose the values to experiment with - they must be reasonable and you must be able to explain why they are reasonable). After running cross

[1] Download the dataset here: `https://archive.ics.uci.edu/ml/datasets/Multiple+Features`

validation, choose the optimal hyperparameter values and apply the learned model with those values to the held out test set and report the error rate on the test set. Make sure to explain why you chose those hyperparameter values.

(b) **Code**: Submit the file `multi_SVM.py` which contains the function `def multi_SVM(dataset: str) -> None:` where dataset is a string consisting of the name of the dataset and the function does not return anything but must print out to the terminal (stdout) the average train and validation error rates and standard deviations, the optimal hyperparameter values, and the test set error rate for the best model.

Additional instructions: Code can only be written in Python 3.6+; no other programming languages will be accepted. One should be able to execute all programs from the Python command prompt or terminal. Please specify instructions on how to run your program in the README file.

Each function must take the inputs in the order specified in the problem and display the textual output via the terminal and plots/figures should be included in the report.

For each part, you can submit additional files/functions (as needed) which will be used by the main file. In your code, you **cannot** use machine learning libraries such as those available from scikit-learn for learning the models. However, you may now use scikit-learn for cross validation - consider the function `sklearn.model_selection.KFold` and see details here: `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html`. You may also use libraries for basic matrix computations and plotting such as numpy, pandas, and matplotlib. Put comments in your code so that one can follow the key parts and steps in your code.

Your code must be runnable on a CSE lab machine (e.g., csel-kh1260-01.cselabs.umn.edu). One option is to SSH into a machine. Learn about SSH at these links: `https://cseit.umn.edu/knowledge-help/learn-about-ssh`, `https://cseit.umn.edu/knowledge-help/choose-ssh-tool`, and `https://cseit.umn.edu/knowledge-help/remote-linux-applications-over-ssh`.

Instructions

Follow the rules strictly. If we cannot run your code, you will not get any credit.

- **Things to submit**

  1. hw2.pdf: The report that contains the solutions to Problems 1, 2, 3, and 4 including the summary of methods and results.
  2. `dual_SVM.py`: Code for Problem 2.
  3. `kernel_SVM.py`: Code for Problem 3.
  4. `multi_SVM.py`: Code for Problem 4.
  5. README.txt: README file that contains your name, student ID, email, instructions on how to run your code, any assumptions you are making, and any other necessary details.
  6. Any other files, except the data, which are necessary for your code.

Homework Policy. (1) You are encouraged to collaborate with your classmates on homework problems, but each person must write up the final solutions individually. You need to list in the README.txt which problems were a collaborative effort and with whom. (2) Regarding online resources, you should **not**:

- Google around for solutions to homework problems,
- Ask for help on online,
- Look up things/post on sites like Quora, StackExchange, etc.