Assignment 2

UMC 203: Artificial Intelligence and Machine Learning

March 2024

No copying is allowed. **Thorough plagiarism check will be run.** You are given two questions, each of which requires Python programming. **You may use jupyter notebook to write your Python programs, although not necessary. If you are using a notebook, please convert it to a Python file before you submit it.**

SUBMISSION INSTRUCTIONS

1. You should submit **two files** (NOT a zip file) with the following naming convention.

   ▷ `AIML_2024_A2_LastFiveDigitsOfSRNumber.pdf` → Answers to all the problems.

   ▷ `AIML_2024_A2_LastFiveDigitsOfSRNumber.py` → Code for the all the problems.

   For example, if the last five digits of your SR Number is 20000, then you should submit four files: `AIML_2024_A1_20000.pdf`, `AIML_2024_A1_20000.py`.

2. **Any deviation from the above rule will incur serious penalty!**

3. For the coding questions, you are asked to report some values, e.g., the number of iterations. These values should be reported in the `.pdf` file you submit.

4. At the top of the `.pdf` file you submit, write your name and SR Number.

5. You will get a bonus of 10% if your reports are typed neatly in LaTeX

6. You will be using the `numpy` and `sklearn` libraries for this assignment.

7. All datasets are available under `Files>Assignment 2`

1 Support Vector Machine (25 marks)

For this question, you will be implementing the slack SVM and the kernel SVM. You will be using the `cvxopt` library to solve the quadratic programs involved.

- Download the Fashion-MNIST train and test set from the Assignment 2 folder on MS Teams.
- Load this data using numpy. For each row in the CSV, the first column corresponds to the data label, and the next 784 columns represent pixel values. It is **recommended** that you append/concat a column of `1`s to the CSV. This will act as a feature corresponding to the bias of the SVM.
- Run the following algorithms on this data set.
- Do not use the test for training.

The binary case

Train the following SVMs using the train set.

1.a ($2.5 \times 4 = 10$ marks) Report the accuracy of the test set for the four experiments below in a table.

Run the following bit of code to generate two labels.

```
srn = ... # Please fill in the last 5 digits of your SRN
c0, c1 = np.random.default_rng(srn).choice(range(0,10),size=2,replace=False)
```

- Solve the primal slack linear SVM optimization problem using `cvxopt`.
- Solve the dual slack linear SVM optimization problem using `cvxopt`.
- Solve the dual of the kernelized SVM optimization problem using the RBF kernel for $\sigma = 1$ using `cvxopt`.

$$K_{\text{RBF}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right)$$

- Solve the SVM using the scikit-learn linear SVM package.

Note: For all slack SVMs, experiment with C = 1, 10, 100 and record all observations.

Multiclass SVM

Reference: Koby Crammer and Yoram Singer. 2002. On the algorithmic implementation of multiclass kernel-based vector machines. J. Mach. Learn. Res. 2 (3/1/2002), 265–292. (Link)

We can extend the support vector machine optimization problem into a multiclass SVM. Such a generalization yields the following primal optimization problem:

$$\begin{aligned} \min_{w_1,\ldots,w_K,\xi} \quad & \frac{1}{2}\sum_k ||w_k||^2 + C \sum_{(x_i,y_i)\in\mathcal{D}} \xi_i \\ s.t. \quad & w_{y_i}^T x_i - w_k^T x_i \geq 1 - \xi_i \quad \forall (x_i, y_i) \in \mathcal{D} \text{ and } k \in [K] - \{y_i\} \\ & \xi_i \geq 0 \quad \forall\, i \in [|\mathcal{D}|] \end{aligned}$$

Solving the optimization problem gives you vectors w_k and slack constants ξ_i so that $\tilde{w}^T x_i$ for the correct class $\tilde{w}$ is at least $1 - \xi_i$ greater than $w^T x_i$ for any other class. This allows you to write a classifier.

1.b (1 mark) Formally write down the classifier in terms of w_is.

1.c (5 marks) Derive the lagrangian dual of the above optimization problem

Run the following bit of code to generate five labels.

```
srn = ... # Please fill in the last 5 digits of your SRN
c0, c1, c2, c3, c4 = np.random.default_rng(srn).choice(range(0,10),size=5,replace=False)
```

1.d (4 marks) Implement the multi-class SVM classifier for the five classes above in the fashion-MNIST dataset and compute the test accuracy. Experiment with $C = 1, 10, 100$. Report accuracy on the test set.

1.e (5 marks) Implement a kernelized multiclass classifier for the five classes above in the fashion-MNIST dataset using the dual formulation with the following kernels:

- Linear Kernel,
- RBF kernel with $\sigma = 1$, and
- RBF kernel with $\sigma = 10$.

Report accuracy on the test set.

2 Regression (15 marks)

2.1 Linear Regression (7 Marks)

- (1 Mark) Load **house_price_prediction.csv** dataset using numpy. Drop non-real datatype features. Split into train & test dataset with 80:20 ratio.
- (4 Marks) Apply **Linear Regression & Ridge Regression** on data where target varible is **Price**.
- (2 Marks) Plot regression line with **sqft-living** feature & report MSE on test dataset for both linear regression and ridge regressions.

2.2 Gaussain Process Regression (8 Marks)

- Load **weather_data.csv** dataset using numpy. (0.5 Marks)
- Create X for training and y for testing timestamps. (0.5 Marks)

```
X = np.atleast_2d([float(i) for i in range(1,201)]).T
y =  np.atleast_2d(np.linspace(1,200,10000)).T
```

- (4 Marks) Apply Gaussian Process Regression on data using **sklearn.gaussian_process** with these 4 kernel functions. Here CK = ConstantKernel() , ESS = ExpSineSquared() & RQ = RationalQuadratic().
  - CK() * ESS(length_scale=24 , periodicity = 1)
  - CK() * RQ(length_scale=24 , alpha = 1)
  - CK() * (ESS(length_scale=24 , periodicity = 1) + RQ(length_scale=24 , alpha = 0.5))

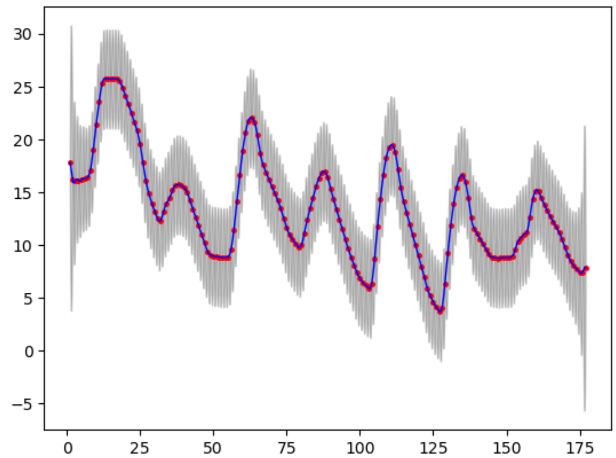

Figure 1: Example plot to submit

  - CK() * ESS(length_scale=24 , periodicity = 1) * RQ(length_scale=24 , alpha = 0.5)

- (3 Marks) Plot of real and predicted data along with 95% confidence intervals for each kernel on **Outdoor Drybulb Temperature [C]** feauture(example plot for kernel 1).