CSE, IITB

CS 335 - AI & ML
Lab 3 - Logistic & Kernel Regression

Please read the following important instructions before getting started on the assignment

1. The assignment should be completed individually. Do not look at solutions to this assignment or related ones on the Internet.
2. Solutions to the theory questions must be submitted in a single pdf file.
3. All the hyperparameters must be specified in pdf file under **Hyperparameter** section and resources consulted must be duly listed in the **References** section of the pdf file. Do not upload multiple pdf files.
4. **Upload Guidelines** Put all the assignment related files in folder with the convention **lab3-roll_no** and .zip the folder.
5. **Not following folder guidelines will attract penalty.**
6. All source code are checked with code plagiarism checker. Strict action will be taken against the defaulters.
7. Comment out all the print statements from the source before submission. Repetition of such mistakes will attract penalty. We had relaxed this condition for previous two labs.

Problem 1 - Logistic Regression (2.5)

1. The Bank Marketing Data Set is related with direct marketing campaigns of a Portuguese banking institution. The classification goal is to predict if the client will subscribe (yes/no) a term deposit. There are 17 columns in the table that provide information about each client, such as age, marital status, and education level. There are 45,212 data points and training data consists of 30,000 points.
   In this task, you will complete the logistic_train and logistic_predict functions in task.py file. Make sure to write efficient code for this task . You are free to change the default parameter max_iter and learning_rate as you wish (as long as it passes autograder). To test your code, run the following command **python3 autograder.py 1**. You can use preprocess code from your previous submissions. (1.5)
2. What is the accuracy on the test data? Do you think accuracy is a good performance metric for the given problem (based on data)? Go through data before answering this question. You can also try your model on Admission.csv which predicts graduate admissions in dataset folder(optional) (1)

Problem 2 - Kernel Perceptron (1.5)

1. In this problem complete the fit() & project() functions in kernel_perceptron.py. To check you can use **python3 autograder.py 2**. Set bounds on alpha and see how the accuracy varies. Use different kernels and report your observations. For more info on kernel perceptron see tutorial 5 problem 5. (1.5)

Problem 3 - Kernel Logistic Regression (1 Mark)

a) Implement the kernel logistic regression using the gaussian kernel $K_\sigma(x, x') = \exp\left(\frac{\|x-x'\|^2}{2\sigma^2}\right)$. And run your program on dataset1.txt (first two columns are X, last column is Y) with $\sigma = 1$. Report the training error. Set stepsize to be 0.01 and maximum number of iterations 100 (Please use this setting and dont try alternative settings). The scatterplot of the dataset1.txt is shown in Figure 1.
Create a new file **kernel_logistic.py** and make all functions related to Problem 3 in this file.

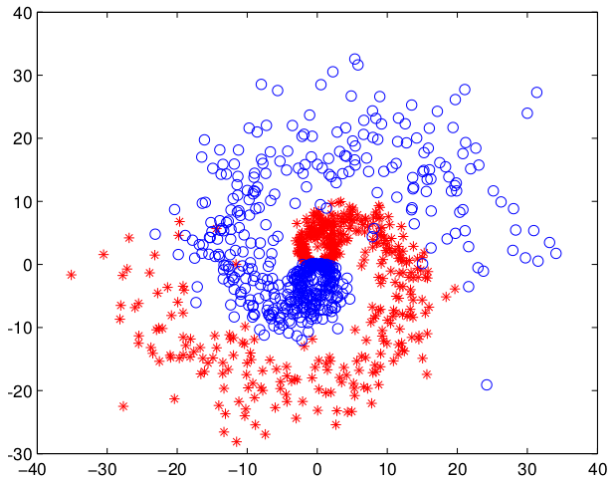


Figure 1: Scatter Plot

b) Use the 10-folds cross-validation to find the best and plot the total number of mistakes for $\sigma = 0.5, 1, 2, 3, 4, 5, 6$.

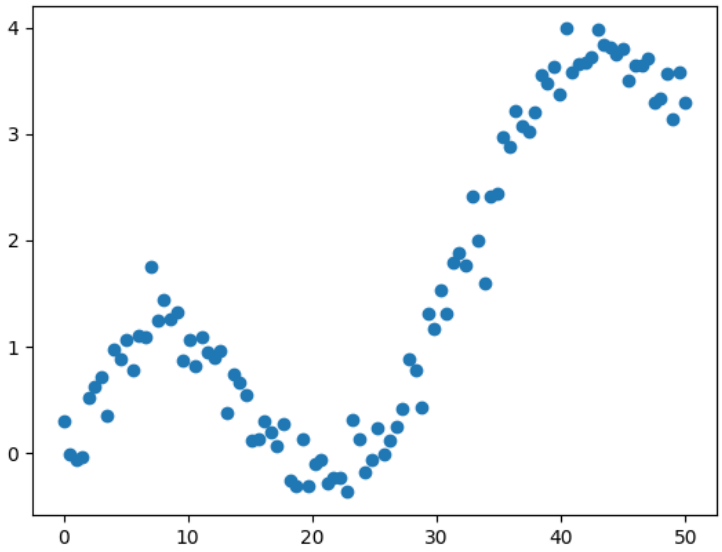
Problem 4 - Kernel Ridge Regression (5)

In this task, you will be implementing kernel ridge regression (for details, refer to problems 2 and 3 of tutorial 5). Complete the function `kernel_ridge_regression(K, X, Y, lmbda)` in `krr.py` that takes as input

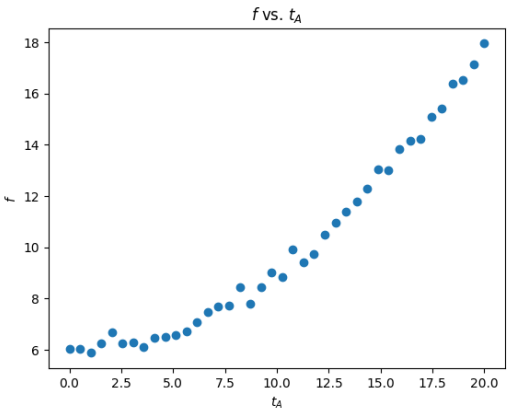
- a kernel function $K(x_i, x_j)$
- X, values of the independent variable x (array of shape $N \times D$, where N is the number of training examples and D is the dimensionality of x)
- Y, corresponding values of the dependent variable y (array of shape $N \times 1$)
- the regularisation parameter λ

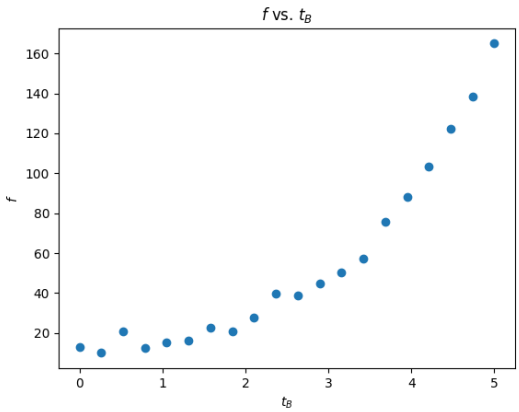
and and returns the regression function F. For evaluation, we will compare $F(X_{test})$ with Y_{test}. Now, you will implement two kernels to be used with this function. (1.5)

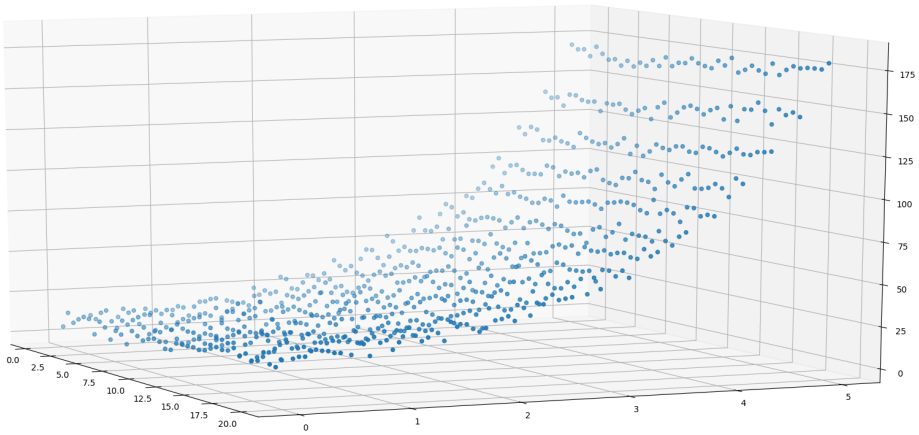
1. Complete the function `gaussian_kernel` in `krr.py` to model the data provided in `problem4_1.csv`. Plot $y = F(x)$ (here F is the regressing function) alongside the training examples (i.e. in the same graph) keeping $\lambda = 0.01$ and $\sigma \in \{1, 10, 100\}$; in a similar second graph, plot $y = F(x)$ keeping $\sigma = 10$ and $\lambda \in \{0.01, 1, 100\}$. Explain the variation of $y = F(x)$ with changes in the values of σ and λ. Also, what would happen as (i) $\sigma \to \infty$, and (ii) $\sigma \to 0$? (1.5)



2. In the future, humans successfully build a space pod capable of travelling in two modes, the slower but efficient mode A, and the faster but inefficient mode B. The variation in fuel consumption with time spent in both modes is shown in the plots below (value of the second variable is kept constant in plots involving only one variable). You are provided with `problem4_2.csv` consisting of noisy data of the form (t_A, t_B, f) where t_i denotes time spent in mode i and f denotes the total fuel consumed.







- *On the basis of the plots above*, suggest and implement a kernel for estimating the total fuel consumption given t_A and t_B by completing the function `my_kernel` in `krr.py`. (1.5)
- Is is possible to obtain a closed form expression for the dependence of f on t_A and t_B? Why or why not? (0.5)

Note that both the functions `gaussian_kernel` and `my_kernel` should be compatible with `kernel_ridge_regression` above. For generating plots, you will have to change the values of σ and λ (feel free to modify the plotting code) in `krr.py` and execute **python3 krr.py**. Refer Kernel-Ridge tutorial for more details.