TRINITY COLLEGE DUBLIN

School of Computer Science and Statistics

Week 6 Assignment CS7CS4/CSU44061 Machine Learning

Rules of the game:

- **Its ok to discuss with others, but we ask that you do not show any code you write to others. You must write answers in your own words and write code entirely yourself. All submissions will be checked for plagiarism.**
- **Reports must be typed (no handwritten answers please) and submitted as a separate pdf on Blackboard (not as part of a zip file please).**
- **Important: For each problem, your primary aim is to show clearly and unambiguously that you understand what you're doing - not just running a program and quoting numbers it outputs. So if you write code to carry out a calculation you need to discuss/explain what that code does, and if you present numerical results you need to discuss their interpretation. Generally most of the credit is given for the explanation/analysis as opposed to the code/numerical answer. Saying "see code" is not good enough, even if code contains comments. Similarly, standalone numbers or plots without further comment is not good enough.**
- **When your answer includes a plot be sure to (i) label the axes, (ii) make sure all the text (including axes labels/ticks) is large enough to be clearly legible and (iii) explain in text what the plot shows.**
- **Include the source of code written for the assignment as an appendix in your submitted pdf report.**
- **Also include a separate zip file containing the executable code and any data files needed. Programs should be running code written in Python, and should load data etc when run so that we can unzip your submission and just directly run it to check that it works.**
- **Keep code brief and clean with meaningful variable names etc.**
- **When selecting a hyperparameter value show cross-validation analysis to justify your choice.**
- **When evaluating an ML algorithm on data, compare against a baseline model.**

Downloading Dataset

- Download the assignment dataset from `https://www.scss.tcd.ie/Doug.Leith/CSU44061/week5.php`. Important: You must fetch your own copy of the dataset, do not use the dataset downloaded by someone else.
- Please cut and paste the first line of the data file (which begins with a #) and include in your submission as it identifies your dataset.
- The data file consists of two columns of data (plus the first header line). The first column is the input feature and the second column is the real-valued target value.

.

Assignment

In this assignment you'll take a closer look at use of kernels with an SVM classifier and with Ridge Regression. Using kernels changes these methods to be quite like the kNN method.

(i) To help get more insight into kNN you'll start by using the dummy training data $(-1, 0)$, $(0, 1)$, $(1, 0)$. That is, there is one input feature and when it equals -1 or 1 the output is 0 and when it equals 0 the output is 1. Using kNN and kernalised ridge regression you'll generate predictions on a grid of feature values that range from -3 to 3 (i.e. extending beyond the training data).

(a) Use the sklearn KNeighborsRegressor function to create kNN predictions for this data. Use a Gaussian kernel to weight neighbours based on their distance from the input feature (you do this by setting the weights parameter of the KNeighborsRegressor function - see the lecture notes for details). Using $k = 3$ (the number of training data points) plot the predictions as the parameter γ of the Gaussian kernel takes values 0, 1, 5, 10, 25.

(b) With reference to how the kNN model generates its predictions explain why the predictions plotted in (a) change the way they do as you vary γ. When $\gamma = 25$ the predictions are close to 1 for all input feature values between -0.5 and 0.5, explain why.

(c) Now use the sklearn KernelRidge function to train a kernalised ridge regression model on this data. Again, use a Gaussian kernel (by setting the kernel parameter of KernelRidge to 'rbf'). The KernelRidge uses a cost function that includes an L_2 penalty and so its necessary to choose the weight $\alpha = 1/(2C)$ given to this penalty. Plot predictions for the kernel parameter γ equal to 0, 1, 5, 10, 25 and for C values that span a wide range e.g. $C = 0.1$, $C = 1$, $C = 1000$. Also report the dual_coef_ parameters from the KernelRidge, these correspond to the trained parameters θ in the lecture notes on kernel methods.

(d) With reference to how the kernalised ridge regression model generates its predictions explain why the predictions plotted in (c) change the way they do as you vary γ and C. Discuss how the value of the parameters θ (which you reported in (c)) affect the predictions. Compare the behaviour of the kernalised ridge regression predictions with that of the kNN predictions.

(ii) Now repeat the analysis in (i) with the data that you downloaded. Generate predictions on a grid of feature values that extends beyond the range of values in the dataset e.g. if the feature in the dataset has values from -1 to 1 generate predictions for values from -3 to 3 or thereabouts.

(a) For a kNN model with Gaussian kernel weights plot the predictions for $k =$ *#points in dataset* as the parameter γ of the Gaussian kernel is varied. Describe how the predictions change as you vary γ, and explain why. The predictions for feature values outside the range of the training data give some insight into the generalisation behaviour of the kNN model, explain the behaviour you observe in your plots.

(b) Now use the sklearn KernelRidge function to train a kernalised ridge regression model on this data. Again, use a Gaussian kernel. Plot the predictions for a range of values of the kernel parameter γ. Discuss how the predictions change as you vary γ.

(c) Use cross-validation to choose a reasonable value for hyperparameter γ for the kNN model. Now use cross-validation to choose γ and α hyperparameter for the kernalised ridge regression model. Generate predictions for both models using these "optimised" hyperparameter values. How do the predictions of the kNN and kernalised ridge regression models compare?