TRINITY COLLEGE DUBLIN

School of Computer Science and Statistics

Week 2 Assignment CS7CS4/CSU44061 Machine Learning

Rules of the game:

- **Its ok to discuss with others, but 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 articulate that you understand what you're doing - not just running a program and quoting numbers it outputs. Long rambling answers and "brain dumps" are not the way to achieve this. 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.**
- **Reports should typically be about 5 pages, with 10 pages the upper limit (excluding appendix with code). If you go over 10 pages then the extra pages will not be marked.**

Downloading Dataset

- Download the assignment dataset from `https://www.scss.tcd.ie/Doug.Leith/CSU44061/week2.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 three columns of data (plus the first header line). The first two columns are input features and the third column is the ± 1 valued target value.

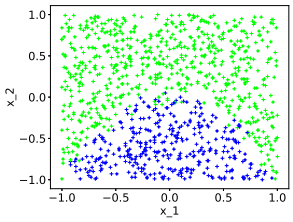
.

Assignment

Use sklearn for this assignment, there's no need to implement any of the models used here yourself. Include the code you write in an appendix to your submission. To read in the data file you can, for example, use pandas:

```
import numpy as np
import pandas as pd
df = pd.read_csv("week2.csv")
print(df.head())
X1=df.iloc[:,0]
X2=df.iloc[:,1]
X=np.column_stack((X1,X2))
y=df.iloc[:,2]
```

(a) (i) Visualise the data you downloaded by placing a marker on a 2D plot for each pair of feature values i.e. for each row in the data. On the plot the x-axis should be the value of the first feature, the y-axis the value of the second feature and the marker should be, for example, a + marker when the target value is +1 and a o when the target is −1. Your plot should look similar in style to this (with different data points of course!):



Be sure to include a legend explaining what markers/colours are used for the +1 and -1 points.

(ii) Use sklearn to train a logistic regression classifier on the data. Give the logistic regression model for predictions and report the parameter values of the trained model. Discuss e.g. which feature has most influence on the prediction, which features cause the prediction to increase and which to decrease.

(iii) Now use the trained logistic regression classifier to predict the target values in the training data. Add these predictions to the 2D plot you generated in part (i), using a different marker and colour so that the training data and the predictions can be distinguished. Show the decision boundary of the logistic regression classifier as a line on the plot (you'll need to use the parameter values of the trained model to figure out what line this should be - explain how you obtain it).

(iv) Briefly comment on how the predictions and the training data compare.

(b) Use sklearn to train linear SVM classifiers on your data (nb: be sure to use the LinearSVC function in sklearn, *not* the SVC function).

(i) Train linear SVM classifiers for a wide range of values of the penalty parameter C e.g. $C = 0.001$, $C = 1$, $C = 100$. Give the SVM model for predictions and report the parameter values of each trained model.

(ii) Use each of these trained classifiers to predict the target values in the training data. Plot these predictions and the actual target values from the data, together with the classifier decision boundary.

(iii) What is the impact on the model parameters of changing C, and why ? What is the impact on the SVM predictions?

(iv) How do the SVM model parameters and predictions compare to those of the logistic regression model in part (a)?

(c) (i) Now create two additional features by adding the square of each feature (i.e. giving four features in total). Train a logistic regression classifier. Give the model and the trained parameter values.

(ii) Use the trained classifier to predict the target values in the training data. Plot these predictions and the actual target values from the data using the same style of plot as before i.e. using just the two original features as x and y axes. Compare and discuss. How do the predictions compare with those in parts (a) and (b) above?

(iii) Compare the performance of the classifier against a reasonable baseline predictor, e.g. one that always predicts the most common class.

(iv) For a bonus try to plot the classifier decision boundary (nb: it can be derived from the model parameters but its not a straight line anymore and you'll need to solve a quadratic equation). Don't worry if you can't do this though.