CS534 — Implementation Assignment 3

General instructions.

1. Please use Python 3 (preferably version 3.6+). Unless otherwise specified, you may only use the following packages: Numpy, Pandas, and matplotlib, along with the standard library (such as 'math', 'os', or 'random' - for example).

2. You can collaborate in group of up to three people. Please do not share code with other groups, or copy program files/structure from any outside sources like Github. Each group's work should be your own.

3. Submit your report on Canvas and your code on TEACH following this link: `https://teach.engr.oregonstate.edu/teach.php?type=assignment`.

4. Your code should follow the structure and function specification of the provided skeleton code. Your code should assume that the data is in the same folder as the code.

5. Please test your code before submission on the EECS servers (i.e. babylon) to make sure that it runs correctly on the server and produce the correct results as reported in the your report.

6. Be sure to answer all the questions in your report. You will be graded based on your code as well as the report. The report should be clear and concise, with figures and tables clearly labeled with necessary legend and captions. The quality of the report and is worth 10 pts. The report should be a PDF document.

7. In your report, the **results should always be accompanied by discussions** of the results. Do the results follow your expectation? Any surprises? What kind of explanation can you provide?

Sentiment analysis with support vector machines

(total points: 90 pts + 10 report pts)

For this assignment you will use sklearn (specifically the function svm.SVC()) to experiment with support vector machines, and perform hyperparameter tuning.

Data. The data for this assignment is a natural language sentiment dataset from Twitter. The first column contains the text of the tweets in question, and the second column contains a zero for negative sentiment tweets and a one for positive sentiment tweets.

The data is provided to you in both a training set: **IA3-train.csv**, and a validation set: **IA3-dev.csv**.

Part 0: preprocessing (10 pts) Preprocessing will involve converting the text of the tweets into a bag-of-words representation that can be processed by the models you'll train.

(a) You will first use the CountVectorizer function in sklearn to generate a feature vector for each tweet where the element of the vector tracks the number of times a specific word appears in the tweet. This function internally will first build a vocabulary of features, which determines what word each dimension tracks. E.g., if the 10^{th} entry in our vocabulary is the word "good", then the 10^{th} entry in a tweet's feature vector will be equal to the number of times the word "good" appears in the tweet.

> Please report the ten most frequent words (frequency is measure by the total counts of each word are maximal) for the positive tweet and negative tweet respectively. Do you find these words to be informative of the classes?

(b) Next you will use the TfidfVectorizer function in sklearn to generate the TF-IDF(term frequency-inverse document frequency) representation for the tweets.

> Please report the ten words with the highest total TF-IDF's for the positive tweet and negative tweet respectively. How do they compare to the list in 0(a)? Which one do you think are more informative and why?

For the remainder of this assignment, you will use the TF-IDF representation to perform the experiments.

Part 1: Linear SVM (20 pts Use sklearn to train linear SVMs and tune hyperparameter c. You should start with $c = 10^i$ with $i = -4, -3, \cdots, 3, 4$. With the results of these experiments, you should then expand your search to include other possible values of c. The basic principle is that if the best c value is found on the boundary of the search range, you want to further expand that particular boundary. If it is the middle, you should refine the search grid to look further in the neighborhood of the current best. You should search for additional values of c with the goal of optimizing the validation performance.

> 1. What is the best validation performance you are able to achieve with linear SVM? What c value is used?
>
> 2. What trend do you theoretically expect to see for training and validation performance as we increase c? Plot the training and validation accuracy against different values of c. Does the trend you observe match your expectation?
>
> 3. What relationship do we theoretically expect between c and the number of support vectors? Plot the number of support vectors against the different values of c. Does the trend you see match your theoretical expectations?

Part 2. Quadratic SVM. (20 pts) For this part, you will use sklearn to trian SVMs with quadratic kernels and tune hyperparameter c following the same protocal as outlined in part 1.

1. What is the best validation performance you are able to achieve with the quadratic kernel? what c value is used?

2. What trend do you theoretically expect to see for training and validation performance as we increase c? Plot the training and validation accuracy against different values of c. Does the trend you observe match your expectation?

3. What relationship do we theoretically expect between c and the number of support vectors? Plot the number of support vectors against the different values of c. Does the trend you see match your theoretical expectations?

Part 3: SVM with RBF kernel (30 pts) Use sklearn to train RBF kernel SVMs and tune hyperparameters c and γ ($\gamma = 1/\sigma^2$ in the RBF kernel definition provided in the lecture slides). For c you should start with the same range as described above. For γ, you should start with the following range: 10^i with $i = -5, -4, \cdots, 1$. Use the heatmap function from seaborn `https://seaborn.pydata.org/generated/seaborn.heatmap.html` to plot the training accuracy and validation accuracy (separately in two different heatmaps) as a function of c and γ. Please expand your search beyond the provided values to optimize validation performance but your heatmap don't have to include these additional values beyond the specified range.

1. What is the best validation performance you are able to achieve for RBF kernel? What c and γ parameters are used?

2. What trend do you theoretically expect to see for training and validation performance as we increase c with fixed γ? Does the trend you observe match your expectation?

3. What trend do you theoretically expect to see for training and validation performance as we decrease γ with fixed c? does the trend you observe match your expectation?

4. With fixed γ What relationship do we theoretically expect between c and the number of support vectors? Plot the number of support vectors as a function of c value for $\gamma = 0.1$. does the trend you see match your theoretical expectations?

5. With fixed c, what relationship do we theoretically expect between γ and the number of support vectors? Plot the number of support vectors as a function of γ for $c = 10$. Does the trend you see match your theoretical expectations?

Part 4. Final discussion question (10 pts):

Comparing across the three different models and their performances, please discuss, for each model, the main sources for their error. Specifically, do you think for each case, the error is primarily the modeling, estimation, optimization or Bayes error? Or a combination of multiple types of errors? Please provide a clear rationale for your claims.