CS 312: Artificial Intelligence Laboratory

Lab 6 report

— B Siddharth Prabhu (200010003)

— Sourabh Bhosale (200010004)

- **Introduction:**

  The objective of this task is to do Spam Email classification using Support Vector Machine. Using an SVM to classify emails into spam or non-spam categories and report the classification accuracy for various SVM parameters and kernel functions.

- **Dataset description (As given):**

  An email is represented by various features like frequency of occurrences of certain keywords, length of capitalised words etc. A data set containing about 4601 instances are available in this **link**. The data format is also described in the above link. You have to randomly pick 70% of the data set as training data and the remaining as test data.

- **Libraries and Packages:**

A) Pandas:

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis/manipulation tool available in any language. Pandas allows importing data from various file formats such as comma-separated values, JSON, SQL, Microsoft Excel.

B) Scikit-Learn:

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It uses numpy extensively for high-performance linear algebra and array operations. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to inter-operate with the Python numerical and scientific libraries NumPy and SciPy. Furthermore, some core algorithms are written in Python to improve performance. Support vector machines are implemented by a Python wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR.

- **Kernels:**

  A) Linear kernel:

  Linear Kernel is used when the data is linearly separable, that is, it can be separated using a single line. It is one of the most common kernels to be used. If there are large number of features in a particular data set, linear kernel-based separation gives better results. Linear kernel-based classification gives better results when no pre-processing has happened.

  SVM Package used: svm.SVC and kernel function is linear

  (svm.SVC(kernel = 'linear', C = c_val))

  B) Quadratic kernel:

  Here, instead of a line we use a Quadratic kernel function to classify the given data. In machine learning, the polynomial kernel is a kernel function commonly used with support vector machines (SVMs) and other kernelized models, that represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models.
  Intuitively, the polynomial kernel looks not only at the given features of input samples to determine their similarity, but also combinations of these. In this problem, the degree is assumed as 2. (By default, it is 3.)
  Quadratic kernel function is the most-used polynomial kernel because higher degree may cause over-fitting and leads to misclassification.

  SVM Package used: svm.SVC and kernel function is poly of degree 2

  (svm.SVC(kernel = 'poly', degree = 2, C = c_val))

C) RBF kernel:

RBF means Radial basis function. In machine learning, the radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification.

SVM Package used: svm.SVC and kernel function is RBF

(svm.SVC(kernel = 'RBF', C = c))

RBF is the default kernel used within the sklearn's SVM classification algorithm and can be described with the following formula:

$$K(x, x') = e^{\frac{||x - x'||^2}{n_{features} \times \sigma^2}}$$

Where, $||x - x'||^2$ is the squared euclidean distance between two points (feature vectors).

- **Experimental results and analysis:**

  Note: First row for each c-value is the training accuracy, second row is the testing accuracy.

	C	Linear	Quadratic	RBF
1	0.0001	0.711801	0.612732	0.612732
		0.696596	0.590152	0.590152
2	0.001	0.886645	0.614285	0.612732
		0.881245	0.591600	0.590152
3	0.01	0.920186	0.635714	0.692857
		0.916727	0.604634	0.674149
4	0.1	0.930745	0.731677	0.913975
		0.928312	0.716871	0.909485
5	1	0.932608	0.860869	0.949689
		0.929036	0.833454	0.936278
6	10	0.931987	0.953416	0.966770
		0.928312	0.914554	0.932657
7	100	0.934161	0.972360	0.988198
		0.931209	0.903692	0.916002
8	1000	0.933960	0.986645	0.993167
		0.931108	0.900796	0.912382

Accuracy obtained across different C parameters and different kernels

For a given C and a given kernel, the pair (a, b) is obtained. a is the training accuracy and b is the testing accuracy.

Linear: In the above mentioned dataset, for C = 100, we obtain the highest accuracy for testing (0.931209) & training (0.934161) dataset in the linear kernel mode implementation. Although Linear kernel function gives more accuracy for lower C values, it takes a while to execute and give output.

Quadratic: It takes very less time for execution but accuracy of the results is somewhat less compared to linear kernel. The best value of C for quadratic kernel is C = 10, with high testing (0.914554) and training (0.953416) accuracies.

RBF: This is also taking less time for execution and accuracy of the results is also more. It performs poorly for smaller C but with larger C, it performs as well as the linear kernel. The best value of C for RBF kernel is C = 1, with high testing (0.936278) and training (0.949689) accuracies.

By taking both computational time and accuracy of the algorithm, we would prefer to use RBF kernel function for SVM classification.

We also considered sigmoid kernel, for which the best value of C comes out to be 0.1; with corresponding testing and training accuracies as 0.895003 and 0.896583.

- **Conclusion:**

  In this assignment, we have made the spam email classifier using different kernels in Support Vector Machine. We have analysed the behaviour of the SVM by checking testing accuracy of the model with different C Values. The above-mentioned table reveals the best C values which yield best test accuracy.

  The training data is linearly separable because kernels other than linear model perform poorly with low value of C. For very tiny values of C, we should get misclassified examples, often even if our training data is linearly separable. (Linear kernel works well for small C, but takes long execution time for larger C.)