SVM - REPORT

Linear SVM with hard margin provides a hyperplane to separate the data points. After training the hard margin linear SVM by using LIBSVM as a python interface, accuracy is computed by using prediction on the model. The model enables to separate the data points linearly and accuracy shows how much the data points are separated properly.

Training accuracy: *86.6667%*
Test accuracy: *85%*

Training and test accuracies indicates that the hyperplane in the hard margin linear SVM separate data points, but the model has some outliers.

Soft-margin SVM accepts changes in the margin by adding a term. In the lecture notes, it is seen as below.

$$\min_{b,\mathbf{w},\varepsilon} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\varepsilon_n$$

$$\text{subject to:} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 - \varepsilon_n \text{ for } n = 1,\dots,N$$

$$\varepsilon_n \geq 0, \text{ for } n = 1,\dots,N$$

C in the added part enables to change margin. That is; if C is large, then the margin will be smaller and if C is small, then the margin will be larger. Task 2 is to understand how to affect C to accuracy of the model and to get accuracies of the model with different kernels and different values of C. Different kernels refers to linear, polynomial, radial basis function and sigmoid and C values are selected as 1, 10, 100, and 1000.

Firstly, let's look at the effect of different models with different kernels on the accuracy. C is fixed as 1.

	Linear Kernel	**Polynomial Kernel**	**Radial Basis Kernel**	**Sigmoid Kernel**
Training accuracy	*86.67 %*	*86.0 %*	*86.67 %*	*82.67 %*
Test accuracy	*85.0 %*	*82.5 %*	*84.17 %*	*84.17 %*

According to accuracy results, linear kernel and radial basis function kernel have the highest training accuracy and linear kernel has the highest testing accuracy. Therefore, linear kernel provides the most accurate model.

Now, let's look at the effect of C on the accuracy in different models with different kernels.

		Linear Kernel	Polynomial Kernel	Radial Basis Kernel	Sigmoid Kernel
C = 1	*Training accuracy*	*86.67 %*	*86.0 %*	*86.67 %*	*82.67 %*
	Test accuracy	*85.0 %*	*82.5 %*	*84.17 %*	*84.17 %*
C = 10	*Training accuracy*	*88.67 %*	*94.0 %*	*95.34 %*	*78.0 %*
	Test accuracy	*81.67 %*	*80.83 %*	*77.5 %*	*80.0 %*
C = 100	*Training accuracy*	*88.67 %*	*98.67 %*	*99.33 %*	*76.67 %*
	Test accuracy	*81.67 %*	*75.0 %*	*78.33 %*	*72.5 %*
C = 1000	*Training accuracy*	*90.0 %*	*100.0 %*	*100.0 %*	*75.33 %*
	Test accuracy	*81.67 %*	*75.83 %*	*76.67 %*	*74.17 %*

As a result:

C = 1000 is the best choice for train data and C = 1 is the best choice for test data for the linear model.

C = 1000 is the best choice for train data and C = 1 is the best choice for test data for polynomial model.

C = 1000 is the best choice for train data and C = 1 is the best choice for test data for radial basis model.

C = 1 is the best choice for both train and test data for sigmoid kernel model.

Value of C is increased for different types of kernels while other parameters remain the same to see how the number of support vectors changed.

	Linear Kernel	Polynomial Kernel	Radial Basis Kernel	Sigmoid Kernel
C = 1	58	118	83	79
C = 10	51	84	74	55
C = 100	50	74	70	45
C = 1000	49	73	64	43

As it can be seen above, number of support vectors decrease as C increases.
In theory, if C increases, the size of the violations will reduce; so, the margin is narrower, and there are fewer support vectors. Therefore, our observations match the theory.

One of the support vectors and one of the data points that is not a support vector is removed and changes in the hyperplane are compared.

When a normal data point is removed from the training set, it is observed that the support vectors do not change (as well as the margin). Because only the support vector points contribute to the result of the algorithm, removal of a normal data point does not affect the hyperplane.

On the other hand, removal of a support vector changes the hyperplane because the foundation of the hyperplane (support vectors) and the margin changes.

Python and CVXOPT QP solver are used to implement hard margin SVM. Matrices that were derived in the class are used:

$$Q = \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & I_d \end{bmatrix}, \mathbf{p} = \mathbf{0}_{d+1}$$

$$A = \begin{bmatrix} y_1 & y_1\mathbf{x}_1^T \\ \vdots & \vdots \\ y_N & y_N\mathbf{x}_N^T \end{bmatrix}, c = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

As a result, the vector u is found which includes the b* and w*.