Question 1

How is Soft Margin Classifier different from Maximum Margin Classifier?

Answer

A maximum margin classifier maintains the largest possible distance from the nearest points of both the classes. The margin is like a band that the hyperplane has on both its sides. It tries to separate the classes perfectly. This is a striking contrast to the soft margin classifier, since for a soft margin classifier, the constraint of maximizing the margin of the line that separates the classes is a little bit relaxed. This allows for some points in the classes to get misclassified.

The maximal margin classifier performs perfectly on the training data but may perform poorly on unseen data. This happens because the maximal margin line (hyperplane) is very sensitive to the training data. It can be extremely sensitive to individual observations. In other words, the model can drastically change if a few points are changed. This is different for soft margin classifiers. Since there is a little scope of misclassification, the model is less sensitive to individual observations or outliers and performs better for unseen data.

There are cases where a maximum margin classifier is not possible at all. These are the cases where the classes cannot be perfectly separated by a straight line (happens when the data points are partially intermingled). This case is handled better by the soft margin classifier. A few outliers, which might make perfect classification impossible, get misclassified and a classifier can be built.

Question 2

What does the slack variable Epsilon (ε) represent?

Answer

The support vector classifier allows certain points to be deliberately misclassified. Epsilon(ε) stands for the permissible error in the SVM and is used to control this misclassification.

The maximal margin classifier equation can be formulated as $(l_i X(W_i.Y_i)) \geqslant M$. To control the misclassifications that a soft margin classifier allows, a slack variable is added to this equation. The equation is changed as: $(l_i X(W_i.Y_i)) \geqslant M(1- \varepsilon_i)$.

The value of slack can range between 0 and positive infinity.

For a value of $\varepsilon = 0$, this becomes equivalent to the maximal margin classifier. It represents the case where the point is correctly classified and is beyond the maximum margin.

A value of $\varepsilon > 1$ represents the case when a point is incorrectly classified (i.e. it violates the hyperplane).

if a data point is correctly classified but falls inside the margin (or violates the margin), then the value of its slack ϵ is between 0 and 1.

Question 3

How do you measure the cost function in SVM? What does the value of C signify?

Answer

For every point i, we must satisfy the following criteria:

$$(l_i \mathbf{X} (W_i . Y_i)) \geqslant M$$

Here, l_i is the ith label, W_i is the ith coefficient and Y_i is the ith data point. **X** denotes dot product. M here is the margin and we try to maximize this margin.

We include a slack variable to this equation to allow the distance of the data-point to be less than M, or in other words, allowing the data-point to come closer to the hyper-plane. The equation is modified as follows:

$$(l_i \mathbf{X} (W_i . Y_i)) \geqslant M(1 - \varepsilon_i)$$

The value of variable ε_i determines how close to the hyper-plane a data-point can get. The effect of different values of ε_i has been explained in the previous answer.

We put a constraint on the error term ε_i by putting a constraint on the total error that can be accepted. The constraint is as follows:

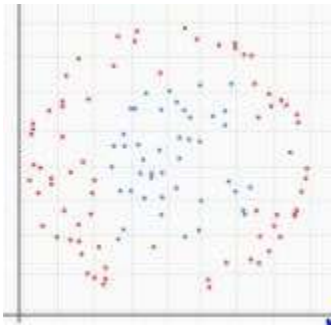
$$\sum \epsilon_i \leq C$$

C is the cost function here which is the summation of all the epsilons of each data point.

When C is large, the slack variables can be large, i.e. a larger number of data points can be misclassified or to violate the margin. The resultant hyperplane has a wide margin and misclassifications are allowed. In this case, the model is flexible, more generalisable, and less likely to overfit. In other words, it has a high bias.

On the other hand, when C is small, the individual slack variables are forced to be small, i.e. not many data points can fall on the wrong side of the margin or the hyperplane. So, the margin is narrow and there are few misclassifications. In this case, the model is less flexible, less generalisable, and more likely to overfit. In other words, it has a high variance.

Question 4



Given the above dataset where red and blue points represent the two classes, how will you use SVM to classify the data?

Answer

Looking at the distribution of the data points, it is evident that it is not possible to separate the red and blue points using a linear hyperplane (a line), a maximum or a soft margin classifier is not possible in this case directly. the data will have to be first transformed from the 2-D attribute space to a linear data in 3-D feature space. We can apply the Maximum Margin Classification or the Soft Margin Classification as applicable once the data is linear.

The linear SVM model will have to be tweaked to incorporate non-linearity. This can be done using kernels. Kernels enable a linear SVM model to separate data-points that are nonlinearly separable.

It can be clearly seen that that the possible boundary between the two classes, looks like a circle, or an ellipse. The equation of a circle/ellipse is as follows:

$$X^2/a + Y^2/b = C$$

This equation represents a circle when a = b, otherwise it represents an ellipse.

These data-points can be plotted on a plane where the axis are (x^2,y^2) instead of being (x,y).

We can consider the new axis to be (x',y'). By applying such a transformation, the equation $X^2/a + Y^2/b = c$ gets transformed to $x'/a + y'/b = c$. This is a linear equation, or an equation of a straight line.

This has enabled us to get a linear separator for a data-set which otherwise had a quadratic separator.

After this transformation, any linear method like SVM or logistic regression can be used. If SVM is used, it'll give us the linear separator that we desire to separate these two classes.

The non-linearity was in the transformation. The method that we used to formulate the separator is still a linear method.

Kernels are the functions in python which help us transform non-linear data. The data given above can be transformed using kernels like polynomial kernel or RBF kernel.

Question 5

What do you mean by feature transformation?

Answer

Feature transformation is the process of transforming the original attributes to a new feature space.

In most cases, although it might be possible to sense that the separator is not going to be linear, the functional form or the shape of the separator may not be obvious at all. For example, it the separator is visualized to be of quadratic form, we can proceed with writing the quadratic equation in its most general form. Each variable in the general equation can then be considered as a dimension in a new feature space.

For example, the most general quadratic form is $ax^2 + bx^2 + cxy + dx + ey + fc = 0$. This can be represented in a new 6-dimensional feature space where X^2, y^2, xy, x, y and c are all features themselves. This is known as feature transformation. When the coefficients from the original quadratic equation are mapped to the new feature space, the original equation can be written as a linear equation.