

Using SVM Parameters with Scikit-learn's SVC Class

Support Vector Machines (SVM) is a powerful machine learning algorithm used for classification and regression tasks. In Scikit-learn, the `SVC` class is used to implement SVM for classification tasks.

When using SVM, there are several parameters that can be tuned to achieve better performance. In this handout, we will cover some of the most commonly used parameters when using the `SVC` class from Scikit-learn.

1. C parameter

The `C` parameter is the penalty parameter of the error term. It controls the tradeoff between the smoothness of the decision boundary and the classification accuracy. A small `C` value results in a smoother decision boundary but may lead to underfitting, while a larger `C` value can lead to overfitting.

To set the value of `C` in Scikit-learn's `SVC` class, use the `C` parameter. For example:

```
from sklearn.svm import SVC

# Instantiate the SVC class with C=1.0
svc = SVC(C=1.0)
```

2. Kernel

The `kernel` parameter specifies the kernel type used for the SVM. The kernel is a function that transforms the input data into a higher-dimensional space to make it easier to separate the classes. Scikit-learn supports several kernel functions, including linear, polynomial, radial basis function (RBF), and sigmoid.

To set the kernel type in Scikit-learn's `SVC` class, use the `kernel` parameter. For example:

```
# Instantiate the SVC class with RBF kernel
svc = SVC(kernel='rbf')
```

3. Gamma

The `gamma` parameter defines the width of the Gaussian kernel used in the RBF kernel. A small `gamma` value results in a wider kernel and smoother decision boundary, while a larger `gamma` value leads to a narrower kernel and more complex decision boundary.

To set the value of `gamma` in Scikit-learn's `SVC` class, use the `gamma` parameter. For example:

```
# Instantiate the SVC class with RBF kernel and gamma=0.1
svc = SVC(kernel='rbf', gamma=0.1)
```

4. Degree

The `degree` parameter is used when using a polynomial kernel. It specifies the degree of the polynomial kernel function. A higher degree results in a more complex decision boundary.

To set the degree of the polynomial kernel in Scikit-learn's `SVC` class, use the `degree` parameter. For example:

```
# Instantiate the SVC class with polynomial kernel of degree 3
svc = SVC(kernel='poly', degree=3)
```

5. Class weight

The `class_weight` parameter is used to balance the weight of the classes in the training data. If the classes are imbalanced, the classifier may be biased towards the majority class.

The `class_weight` parameter assigns a weight to each class, where a higher weight is assigned to the minority class to balance the class distribution.

To set the class weight in Scikit-learn's `SVC` class, use the `class_weight` parameter. For example:

```
# Instantiate the SVC class with class_weight={0:1, 1:10}
svc = SVC(class_weight={0:1, 1:10})
```

In the above example, the weight of class 0 is set to 1 and the weight of class 1 is set to 10, indicating that class 1 is 10 times more important than class 0.

6. Decision function shape

The `decision_function_shape` parameter is used to specify the shape of the decision function. The decision function is a function that maps the input data to a confidence score. By default, Scikit-learn's `SVC` class uses the "ovr" (one-vs-rest) decision function shape, which fits one classifier per class and returns the confidence scores for each class.

To change the decision function shape in Scikit-learn's `SVC` class, use the `decision_function_shape` parameter. For example:

```
# Instantiate the SVC class with decision_function_shape='ovo'
svc = SVC(decision_function_shape='ovo')
```

In the above example, the "ovo" (one-vs-one) decision function shape is used, which fits one classifier for each pair of classes and returns the class that received the most votes.

7. Other parameters

Scikit-learn's `SVC` class also provides several other parameters that can be used to fine-tune the SVM algorithm, such as:

- `tol`: the tolerance for stopping criterion
- `shrinking`: whether to use the shrinking heuristic
- `max_iter`: the maximum number of iterations for the solver

These parameters are less commonly used and are usually set to their default values.

Conclusion

In this handout, we covered some of the most commonly used parameters when using the `SVC` class from Scikit-learn to implement SVM for classification tasks. By tuning these parameters, we can achieve better performance and improve the accuracy of our SVM classifier.

Siolabs Inc.