Understanding One-Class Support Vector Machines

Last Updated : 06 Aug, 2025

Support Vector Machine is a popular supervised machine learning algorithm. it is used for both classifications and regression. In this article, we will discuss One-Class Support Vector Machines model.

One-Class Support Vector Machines

One-Class Support Vector Machine is a special variant of Support Vector Machine that is primarily designed for outlier, anomaly, or novelty detection. The objective behind using one-class SVM is to identify instances that deviate significantly from the norm. Unlike other traditional Machine Learning models, one-class SVM is not used to perform binary or multiclass classification tasks but to detect outliers or novelties within the dataset. Some of the key working principles of one-class SVM is discussed below.

- **Outlier Boundary**: One-Class SVM operates by defining a boundary around the majority class (normal instances) in the feature space. This boundary is constructed to encapsulate the normal data points, effectively creating a region of normalcy.
- **Margin Maximization**: This algorithm strives to maximize the margin around the normal instances, allowing for a more robust separation between normal and anomalous data points. This margin is crucial for accurately identifying outliers during testing.
- **High sensitivity:** One-Class SVM has an in-build hyperparameter called "nu," representing an upper bound on the fraction of margin errors and support vectors. Fine-tuning this parameter influences the model's sensitivity to outliers.

How does One-Class SVM differ from SVM?

SVM and one-class SVMs are like twins but not identical twins, as their usage and principals are different. The three most common differences are discussed below:

Aspects	Support Vector Machine	One-Class Support Vector Machine
Single-Class Training	Requires labeled data from both classes for training	Operates with only the majority class during training
Imbalance Handling	Can't handle imbalance nature of datasets.	Inherently addresses class imbalance, prevalent in outlier detection tasks. By concentrating on the majority class during training.
Outlier Detection Focus	Only aims to find a hyperplane that best separates multiple classes.	Excels in scenarios where the goal is to uncover instances that deviate from the norm like in fraud detection or fault monitoring.

How One-Class SVM Works?

One-Class Support Vector Machines (OCSVM) operate on a fascinating principle inspired by the idea of isolating the norm from the abnormal in a dataset. Unlike traditional Support Vector Machines (SVM), which are adept at handling binary and multiclass classification problems, OCSVM specializes in the nuanced task of anomaly detection. The workflow of OCSVM is discussed below:

1. **Conceptual Foundation:** OCSVM establishes itself on the premise that the majority of real-world data is inherently normal. In most

scenarios, outliers or anomalies are rare occurrences that deviate significantly from the usual patterns. OCSVM's goal is to define a boundary encapsulating the normal instances in the feature space, thereby creating a region of familiarity.

2. **Outlier Boundary Definition:** The algorithm crafts a boundary around the normal instances, often referred to as the "normalcy region." This boundary is strategically positioned to maximize the margin around the normal data points, allowing for a clear delineation between what is considered ordinary and what may be deemed unusual. Think of it as drawing a protective circle around the typical instances to shield them from the outliers or anomalies.
3. **Margin Maximization:** The heart of OCSVM lies in its commitment to maximizing the margin between the normal instances and the boundary. A larger margin provides a robust separation, enhancing the model's ability to discern anomalies during testing. This emphasis on margin maximization is akin to creating a safety buffer around the normal instances, fortifying the model against the influence of potential outliers or anomalies.
4. **Training Process:** During the training phase, OCSVM exclusively leverages the majority class or normal instances. This unimodal focus distinguishes it from traditional SVMs, which necessitate examples from both classes for effective training. By concentrating solely on the norm, OCSVM tailors itself to scenarios where anomalies are sparse, and labeled instances of anomalies are hard to come by. It comes with a fantastic **hyperparameter called 'nu'**. This parameter acts as an upper bound on the fraction of margin errors and support vectors allowed by the model. Tuning the nu parameter enables practitioners to strike a balance between the desire for a stringent model that minimizes false positives (normal instances misclassified as anomalies) and a more lenient model that embraces a higher fraction of anomalies.
5. T**esting and Anomaly Identification:** Armed with the learned normalcy region, OCSVM can swiftly identify anomalies during testing. Instances falling outside the defined boundary are flagged as potential outliers. The model essentially acts as a vigilant guardian,

scrutinizing new data points and signaling if they exhibit behavior significantly different from the norm.

One-Class SVM in Anomaly Detection

In the domain of anomaly detection, One-Class Support Vector Machines (OCSVM) serve as a robust and versatile tool designed to discern normal patterns from irregular occurrences. Notably, OCSVM takes a distinctive approach by training exclusively on the majority class, which represents normal instances, eliminating the need for labeled anomalous data during training. This is particularly advantageous in real-world scenarios where anomalies are rare, making it challenging to obtain sufficient labeled samples. The core principle of OCSVM involves defining a boundary around the normal instances in the feature space, achieved through a specified kernel function and a nuanced parameter termed "nu." This parameter acts as an upper limit on the fraction of margin errors and support vectors, enabling users to fine-tune the model's sensitivity to outliers. During the testing phase, instances falling outside this learned boundary are flagged as potential outliers, facilitating efficient anomaly identification. OCSVM's adaptability extends to various applications, including fraud detection in financial transactions, fault monitoring in industrial systems, and network intrusion detection. Its innate ability to capture complex, non-linear relationships and its focus on the majority class make it a valuable asset in safeguarding systems against unexpected events and ensuring robust anomaly detection across diverse domains. In this article, we will implement credit card anomaly detect using OCSVM further.

Use Cases of One-Class SVM

There are several real-world use-cases of One-Class SVM which are listed below-->

1. **Detecting fraud in financial transactions:** OCSVM excels in rare cases that are not uncommon associated with fraudulent activity in financial transactions. With specialized training in general practices, it becomes adept at distinguishing specific patterns. During testing,

deviations from these scholarly models are immediately flagged, indicating that the fraud can be treated as abnormalities.

2. **Fault detection in commercial systems:** Companies that rely on complex devices can benefit from OCSVM's real-time monitoring of defects or anomalies. When applied to sensor data, OCSVM identifies abnormal behavior, and identifies potential errors. Early detection through OCSVM prevents maintenance, reduces downtime and increases operational efficiency.
3. **Network Intrusion Detection:** OCSVM can play an important role in continuously monitoring computer networks to protect against malicious activity. It helps identify unusual network behaviors that may indicate a possible attack. OCSVM works well in situations where most network traffic is normal and anomalies are very rare.
4. **Quality Control in Manufacturing:** Strict quality control is needed in manufacturing to ensure fault-free products. OCSVM is applied on sensor data or product characteristics to detect deviations from the perfect product. It helps to detect defects early during production.

One-Class SVM Kernel Trick

One-Class SVM supports various kernel options like SVM for optimized performance which are discussed below:

- **Linear Kernel:** The linear kernel is the simplest form of a kernel and is equivalent to performing a linear transformation. It is suitable when the relationship between the features is approximately linear. The decision boundary in the higher-dimensional space is a hyperplane.
- **Polynomial Kernel:** The polynomial kernel introduces non-linearity by considering not just the dot product but also higher-order interactions between features. It is characterized by a user-defined degree parameter (`degree`). A higher degree allows the model to capture more complex relationships but in the same time it may increase the risk of overfitting.
- **Sigmoid Kernel:** The sigmoid kernel is particularly suitable for scenarios where the data distribution is not well defined or exhibits sigmoidal patterns. It is often used in neural network-inspired SVMs.

The `gamma` and `coef0` parameters govern the shape and position of the decision boundary.

- **Radial Basis Function (RBF) or Gaussian Kernel:** The RBF kernel is versatile for handling complex, non-linear relationships. It transforms data into a space where intricate decision boundaries can be drawn. Well-suited when the exact form of relationships is unknown or intricate.
- **Precomputed Kernel**: This kernel allows users to provide a precomputed kernel matrix instead of the actual data. Useful when the kernel matrix is computed using a custom kernel function or when using pairwise similarities between instances.

From these above kernels the RBF kernel is the default kernel for One-Class SVM. In our implementation we will use default kernel for credit card anomaly detection.

Step-by-Step implementation of One-Class Support Vector Machines in Python

Importing required modules

At first, we will import all required Python libraries like Pandas, NumPy, Matplotlib and SKlearn etc.

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import OneClassSVM
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler
```

Dataset loading and preprocessing

Now we will load the famous credit card dataset. For faster implementation we will use first 50k rows of the dataset. Then we will use Standard Scaler to scale the target column. Then we will separate the features and target variable for further usages.

```
credit_data = pd.read_csv('creditcard.csv', nrows=50000) # https://www.kaggle.com/mlg-ulb/creditcardfraud
standardized_data_without_class = StandardScaler().fit_transform(credit_data.loc[:,credit_data.columns!='Class'])
data_50k_new = standardized_data_without_class[0:50000]
data_50k_df = pd.DataFrame(data=data_50k_new)
# Separate features and target variable
X = credit_data.drop(columns=['Class'])
y = credit_data['Class']
```

Model training

Now we will train the One-class SVM on various hyperparameters which are discussed below:

- **kernel**: The choice of the kernel determines the transformation applied to the input data in a higher-dimensional space. Here we have set to default "rbf" which stands for Radial Basis Function, commonly known as the Gaussian kernel. This kernel is suitable for capturing complex, non-linear relationships in the data.
- **degree**: We have set it to default value 3. It defines the degree of the polynomial function and is particularly applicable when the kernel is set to "poly." However, if the patterns of dataset act as polynomial then this parameter automatically handles the kernel as required.
- **gamma**: is a crucial parameter that influences the shape of the decision boundary. A smaller gamma value results in a broader decision boundary which makes the model less sensitive to individual data points. Conversely, a larger gamma value leads to a more complex decision boundary, potentially capturing intricate patterns in the data. Fine-tuning gamma is essential for achieving optimal model performance.
- **nu**: It represents an upper bound on the fraction of margin errors and support vectors. It allows users to control the balance between precision and recall in the model. A smaller nu value makes the algorithm more lenient, permitting a higher fraction of margin errors and support vectors, which can be useful in scenarios with a considerable number of anomalies.

```
clf_svm = OneClassSVM(kernel="rbf", degree=3, gamma=0.1, nu=0.01)
y_predict = clf_svm.fit_predict(data_50k_df)
```

Model evaluation

Here model evaluation is little different from other traditional ML models. This model can be evaluated by accuracy by not for performance of classification rather for outlier or anomalies detection or logically separation.

```
svm_predict = pd.Series(y_predict).replace([-1,1],[1,0])
svm_anomalies = data_50k_df[svm_predict==1]
# Calculate accuracy
accuracy = accuracy_score(y, svm_predict)
print("Accuracy in separating Outlier:", accuracy)
```

Output:

```
Accuracy in separating Outlier: 0.9641
```

Visualizing detected outliers(anomalies)

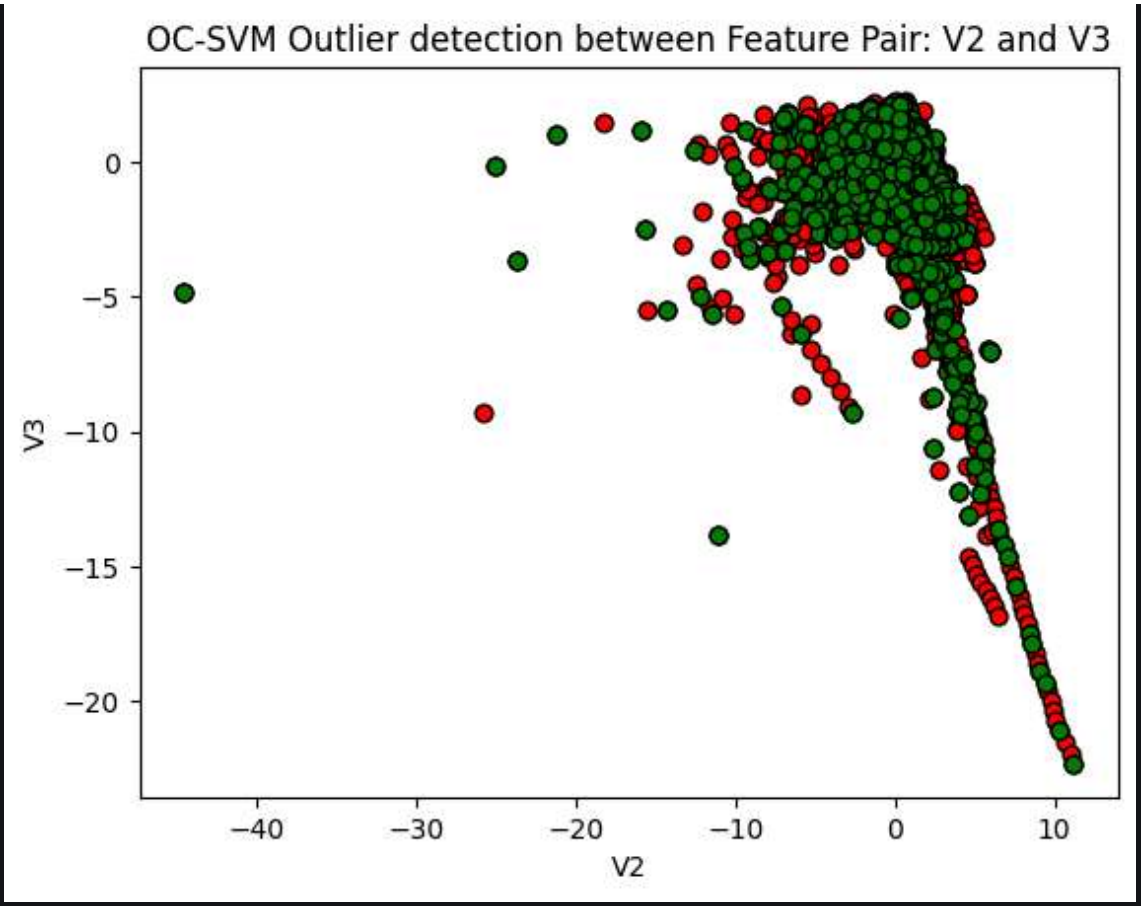
Now we will plot the inlier and outlier plots between any two features. To do this we will define a small function(plot_OCSVM) which can plot any features with outliers any per our choice. By just changing the integer value during function calling we can visualize them.

```
def plot_OCSVM(i):

plt.scatter(data_50k_df.iloc[:,i],data_50k_df.iloc[:,i+1],c='red',s=40,
edgecolor="k")

plt.scatter(svm_anomalies.iloc[:,i],svm_anomalies.iloc[:,i+1],c='green',
s=40, edgecolor="k")
    plt.title("OC-SVM Outlier detection between Feature Pair: V{} and
V{}".format(i,i+1))
    plt.xlabel("V{}".format(i))
    plt.ylabel("V{}".format(i+1))
#plot_OCSVM(1)  # chnage the integer value to visualize different pairs of
features
plot_OCSVM(2)
#plot_OCSVM(3)
```

Output:



Detected Outliers or anomalies

So, from the above plot we can clearly see the One-class SVM has sharply separated the normal occurrences with anomalies(potentially outliers) for both the features. We can also visualize other features by calling the function with different values.

GeeksforGeeks
Sanchhaya Education Private Limited
Corporate & Communications Address:
A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)