Course: Artificial Intelligence and Machine Learning Code: 20CS51I

Week 8 Support vector machine

Session 2

- **Introduction to Support Vector Machine**
- **How does it work?**
- **Application of Support Vector Machine**
- **Build a machine learning model using SVM Technique**

Introduction to Support Vector Machine

Support Vector Machine is a classifier algorithm, that is, it is a classification-based technique. It is very useful if the data size is less.

It is a type of supervised machine learning algorithm. Here, Machine Learning models learn from the past input data and predict the output. Support vector machines are basically supervised learning models used for classification and regression analysis.

SVM is a model that can predict unknown data. For example, if we have a pre-labeled data of apples and strawberries, we can easily train our model to identify apples and strawberries. So, whenever we give it new data – an unknown one – it can classify it under strawberries or apples.

That's SVM in play. It analyzes the data and classifies it into one of the two categories based on the labeled data it already has. As per the previous example, it will sort the apples under the apple category and the strawberries under the strawberry category.

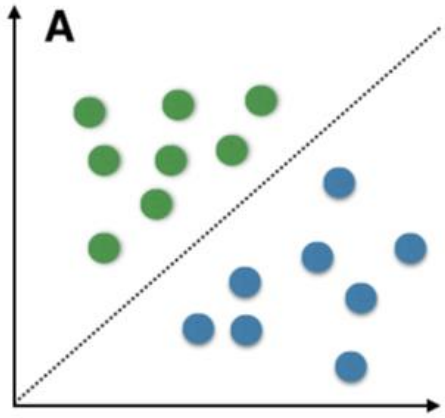
Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.SVMs are different from other classification algorithms because of the way they choose the decision boundary that maximizes the distance from the nearest

data points of all the classes. The decision boundary created by SVMs is called the maximum margin classifier or the maximum margin hyperplane.

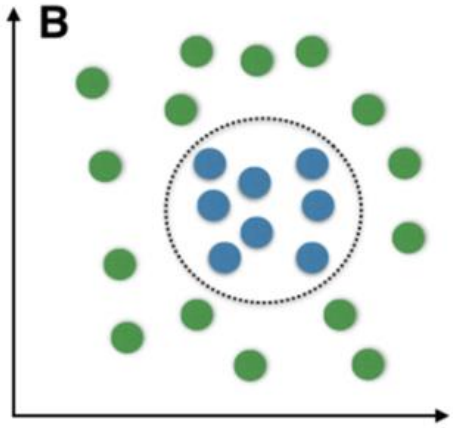
Types of SVM

There are two different types of SVM:

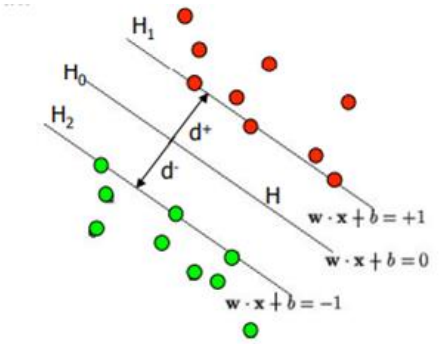
- Simple SVM: Typically used for linear regression and classification problems.



- Kernel SVM (Non-Linear SVM): Has more flexibility for non-linear data because you can add more features to fit a hyperplane instead of a two-dimensional space.



A simple SVM classifier works by making a straight line between two classes. That means all of the data points on one side (Positive or Negative) of the line will represent a category and the data points on the other side of the line will be put into a different category. This means there can be an infinite number of lines to choose from. Kernel SVM is like if we have multiple classifications in that time we user this SVM by apply Kernel Tricks.



H1 and H2 are the planes:

uH1: w.xi+b = +1

uH2: w.xi+b = –1

Define the hyperplanes H such that:

W.xi +b ≥ +1 when yi =+1

W.xi +b ≤ -1 when yi = –1

The points on the planes H1 and H2 are the tips of the Support Vectors

The plane H0 is the median in between, where w.xi+b =0

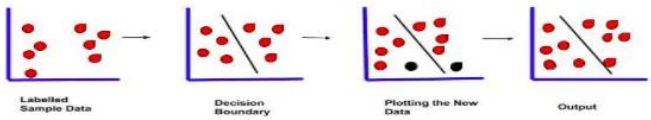
d+ = the shortest distance to the closest positive point

d- = the shortest distance to the closest negative point

The margin (gutter) of a separating hyperplane is d+ + d–.

How does it work?

1. Here, we have our Support Vector Machine where we take the labeled sample of data as seen in the first graph.



2. Further, we draw a line separating the two categories. This line is called the decision boundary. Herein one side of the decision boundary has apples and the other side has strawberries.

3. Now when new data is taken as seen in the third graph, it automatically goes into the group it belongs to – the right or the left side of the decision boundary.

4. And depending on which side of the line the unknown sample data goes, we can predict the unknown and classify it under the apple or strawberry category.

5. Let's get into the details

Application of Support Vector Machine

1. Facial Expression Classification

There are many ways to use this classification. We can use it in various life-care systems, we can use it in normal happy or sad look classification as well. We can use it in filters. If we make certain facial expressions, it would add the specific filter as per the expression. The range of expressions lies between happy and sad.

SVM can be of great use here.

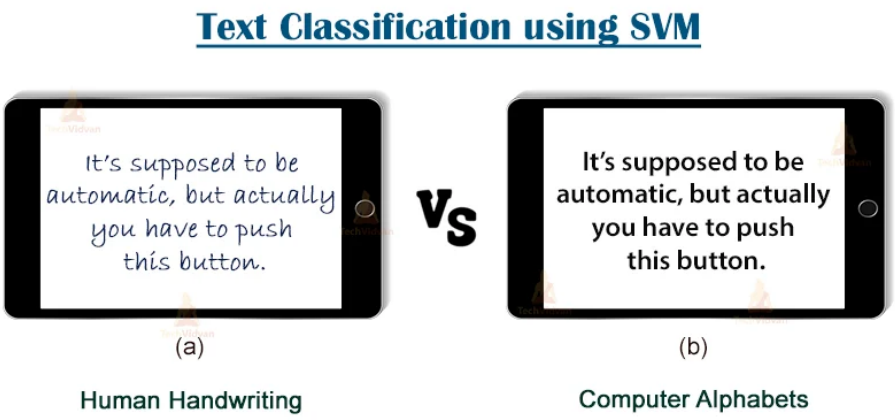
Formatted: Font: Bold

2.Texture Classification using SVM

In this SVM application, we use the images of certain textures and use that data to classify whether the surface is smooth or not. This SVM application will be of great use. If we use a sensitive camera to take pictures and use that data in our model, we could train a really powerful model.

Also, if we take images of surfaces, we could determine the surface as smooth or gritty. SVM here classifies the surface as smooth or gritty.

3.Text Classification



Text Classification is an automatic process of classification of text into predefined categories. We will classify Emails into spam or non-spam, news articles into different categories like Politics, stock exchange , Sports, etc. We could use SVM in differentiating between the handwritings of two different people.

The dataset would contain images of various alphabets and sentences written in both handwriting. The SVM classifier would be trained using the sample data. We can also use SVM to differentiate between human writing and computer alphabets.

4.Speech Recognition

Speech recognition is employed for segregating individual words out of the continual speeches. Features for every isolated word are extracted and models were trained successfully Using speech

recognition, we could make an interface for communication with deaf people. Here, the acoustics would be the data.

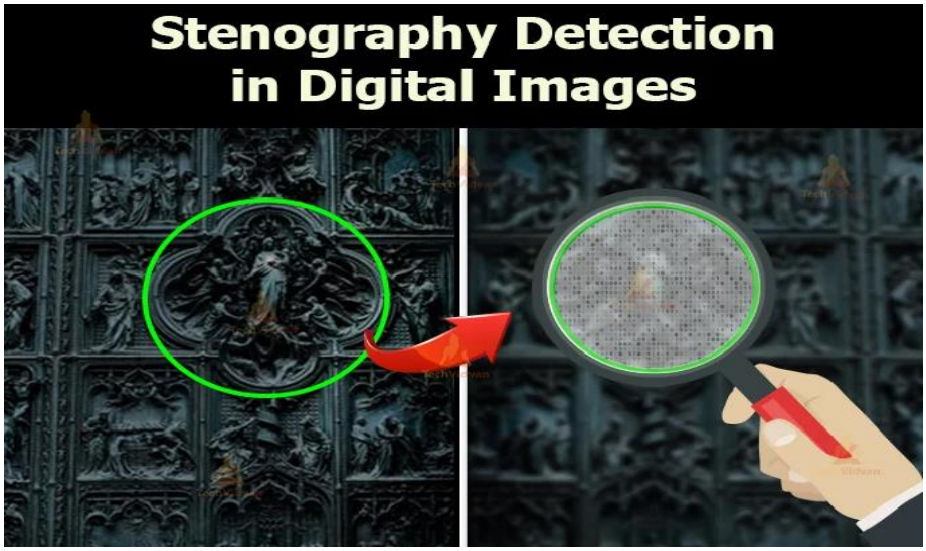
There are many functions like LPC, LPCC, and MFCC which collect the acoustic data. The SVM uses the acoustic data to train its models. We use the data to train many models and use them in the system. The results obtained using SVM are generally accurate.

5.Cancer Diagnosis and Prognosis

Cancer detection is one of the top research fields in the world right now. There are various types of ML methods using which the researches are being conducted. Google is using Image classification methods to find cancer.

There are many such examples. SVM is one of the algorithms that are in use for this. SVM helps in diagnosis and prognosis by running numerous models. There is a lot of data in the form of an image that is analyzed. We have thousands of datasets.

6.Stenography Detection in Digital Images



Using SVM, we can find out if an image is pure or adulterated. This could be used in security-based organizations to uncover secret messages. Yes, we can encrypt messages in high-resolution images.

In high-resolution images, there are more pixels, hence, the message is more hard to find. We can segregate the pixels and store in data in various datasets. We can analyze those datasets using SVM.

Build a machine learning model using SVM Technique

Importing the dataset

```
import pandas as pd
data = pd.read_csv("apples_and_oranges.csv")
```

```
Download dataset from below link
```

https://www.kaggle.com/datasets/raykleptzo/classification-data-apples-oranges

Here is what it looks like :

Index	Weight	Size	Class
0	69	4.39	orange
1	69	4.21	orange
2	65	4.09	orange
3	72	5.85	apple
4	67	4.7	orange
5	73	5.68	apple
6	70	5.56	apple
7	75	5.11	apple
8	74	5.36	apple
9	65	4.27	orange
10	73	5.79	apple
11	70	5.47	apple
12	74	5.53	apple
13	68	4.47	orange
14	74	5.22	apple

Splitting the dataset into training and test samples

```
from sklearn.model_selection import train_test_split
training_set, test_set = train_test_split(data, test_size = 0.2, random_state = 1)
```

Classifying the predictors and target

```
X_train = training_set.iloc[:,0:2].values
Y_train = training_set.iloc[:,2].values
X_test = test_set.iloc[:,0:2].values
Y_test = test_set.iloc[:,2].values
```

Initializing Support Vector Machine and fitting the training data

```
from sklearn.svm import SVC
classifier = SVC(kernel='rbf', random_state = 1)
classifier.fit(X_train,Y_train)
```

Predicting the classes for test set

```
Y_pred = classifier.predict(X_test)
```

Attaching the predictions to test set for comparing

```
test_set["Predictions"] = Y_pred
```

Comparing the actual classes and predictions

Let's have a look at the test_set:

Weight	Size	Class	Predictions
65	4.09	orange	orange
66	4.68	orange	orange
72	5.85	apple	apple
70	4.83	orange	apple
70	4.22	orange	orange
71	5.26	apple	apple
69	4.61	orange	orange
73	5.03	apple	apple

Comparing the 'Class' and 'Predictions' column we find that only one of the 8 predictions has gone wrong.

Calculating the accuracy of the predictions

We will calculate the accuracy using the confusion matrix as follows :

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_test,Y_pred)
accuracy = float(cm.diagonal().sum())/len(Y_test)
print("\nAccuracy Of SVM For The Given Dataset : ", accuracy)
```

Output:

Accuracy Of SVM For The Given Dataset : 0.875

Visualizing the classifier

Before we visualize we might need to encode the classes ‘apple’ and ‘orange’ into numericals.We can achieve that using the label encoder.

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
Y_train = le.fit_transform(Y_train)
```

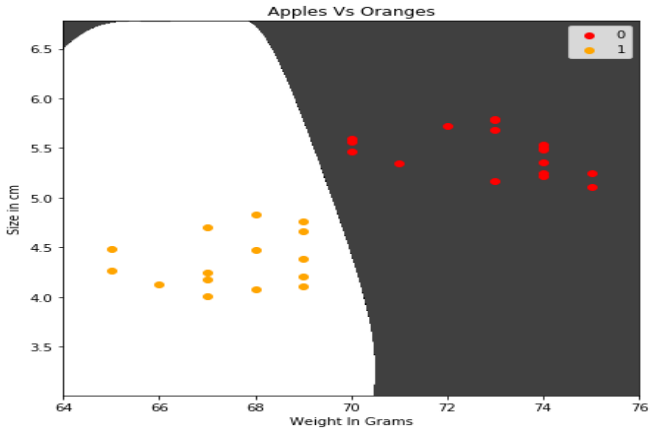
After encoding , fit the encoded data to the SVM

```
from sklearn.svm import SVC
classifier = SVC(kernel='rbf', random_state = 1)
classifier.fit(X_train,Y_train)
```

Let’s Visualize!

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
plt.figure(figsize = (7,7))
X_set, y_set = X_train, Y_train
X1, X2 = np.meshgrid(np.arange(start = X_set[:, 0].min() - 1, stop = X_set[:,
0].max() + 1, step = 0.01), np.arange(start = X_set[:, 1].min() - 1, stop =
X_set[:, 1].max() + 1, step = 0.01))
plt.contourf(X1, X2, classifier.predict(np.array([X1.ravel(),
X2.ravel()]).T).reshape(X1.shape), alpha = 0.75, cmap = ListedColormap(('black',
'white')))
plt.xlim(X1.min(), X1.max())
plt.ylim(X2.min(), X2.max())
for i, j in enumerate(np.unique(y_set)):
plt.scatter(X_set[y_set == j, 0], X_set[y_set == j, 1], c = ListedColormap(('red',
'orange'))(i), label = j)
plt.title('Apples Vs Oranges')
plt.xlabel('Weight In Grams')
plt.ylabel('Size in cm')
plt.legend()
plt.show()
```

Output :



The above image shows the plotting of the training set after fitting the training data to the classifier.The border that separates both the white and black colours represent the Maximum Margin Hyperplane or Line in this case.

According to the SVM classifier, any new data point that falls within the white region is classified as oranges(denoted in orange colour) and any data point that falls on black region is classified as apples(denoted in red colour).

Practice:

Build a SVM Model in python for Fish dataset from Kaggle by using the reference above