Support Vector Machine

quantra

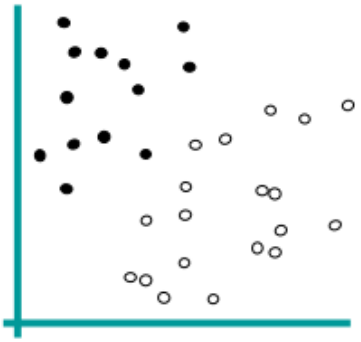
Table of Contents

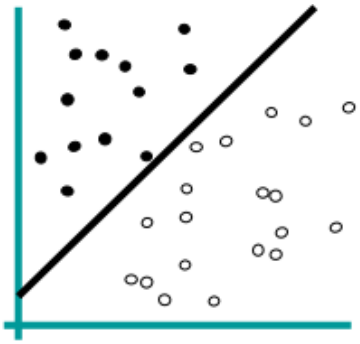
Support Vector Machine 2
SVM Algorithm 4
Output of the SVM model 5

Support Vector Machine

A support vector machine, or SVM, is a machine learning technique widely used in the fields of data analysis and pattern recognition. Classifying data has been one of the major tasks in machine learning problems. The idea of SVM is to build a hyperplane between datasets to classify the data under different labels.

SVMs are supervised learning models that analyse data used for classification and regression analysis. A set of training examples is provided to the SVM algorithm, each belonging to one or the other categories. The algorithm builds a model that assigns new data into one of the categories that it has learned about in the training phase. Consider the plots shown below, the graph on the left indicates the training data in space, the black coloured ones belong to one class and the hollow rings belong to the second class. The SVM algorithm creates a hyperplane which divides this dataset into two parts such that when new data appears in space, depending on the side on which it appears it will be classified into one of the classes. Points lying above the hyperplane will be classified as black dots and the ones lying below as hollow rings.





There could be more than one hyperplane dividing the data space for making the classification. Given a particular hyperplane, we can compute the distance between the hyperplane and the closest data point. Intuitively, a large distance between the data point and hyperplane (also called as margin) is considered optimal. The margin in an SVM model is double this distance. The optimal hyperplane will have the largest margin and that is the problem which the SVM solves.

We will not get into the mathematics of SVM here, but for now to understand the purpose and working of an SVM model would suffice to develop your own SVM algorithm. On the next page of this document is the Python code for developing a SVM model.

SVM Algorithm:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm

# we create 40 separable points
np.random.seed(0)
X = np.r_[np.random.randn(20, 2) - [2, 2], np.random.randn(20, 2) + [2, 2]]
Y = [0] * 20 + [1] * 20

# fit the model
clf = svm.SVC(kernel='linear')
clf.fit(X, Y)

# get the separating hyperplane
w = clf.coef_[0]
a = -w[0] / w[1]
xx = np.linspace(-5, 5)
yy = a * xx - (clf.intercept_[0]) / w[1]

# plot the parallels to the separating hyperplane that pass through the
# support vectors
b = clf.support_vectors_[0]
yy_down = a * xx + (b[1] - a * b[0])
b = clf.support_vectors_[-1]
yy_up = a * xx + (b[1] - a * b[0])

# plot the line, the points, and the nearest vectors to the plane
plt.plot(xx, yy, 'k-')
plt.plot(xx, yy_down, 'k--')
plt.plot(xx, yy_up, 'k--')

plt.scatter(clf.support_vectors_[:, 0], clf.support_vectors_[:, 1],
            s=80, facecolors='none')
plt.scatter(X[:, 0], X[:, 1], c=Y, cmap=plt.cm.Paired)

plt.axis('tight')
plt.show()
```

Output of the SVM model: