CS 337, Fall 2023
Support Vector Machine

Scribes: Arijit Saha, Atharv Kshirsagar, Vanapalli Raja Gopal
Ashwin Goyal, Patil Vipul Sudhir, Hrishikesh Jedhe Deshmukh
(***Equal contribution by all scribes***)
Edited by: Dhiraj Kumar Sah

September 7, 2023

Recap

For linearly separable data, the perceptron finds a hyperplane which separates the given data into 2 classes. Note that the perceptron algorithm finds a hyperplane among all possible hyperplanes which separate the data. Different algorithm runs will not lead to the same hyperplane as output.
The figure below shows an example of hyperplanes that separate the linear separable data

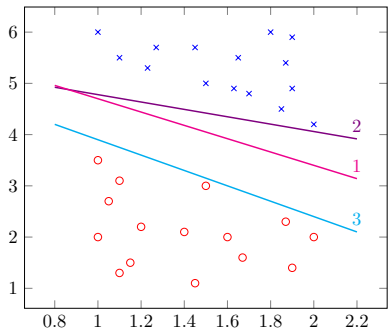

Figure 1: Hyperplanes found by perceptron algorithm

Question: Which is the **best** hyperplane? What do we mean by **best**? How do we find it?
Ans: In Figure 1, we would like our hyperplane to be 1 instead of 2 or 3. The reason is hyperplanes 2 and 3 are extremely close to either of the classes creating a possibility of mis-classification for new test samples. So we define **best** hyperplane to be the hyperplane which **maximizes** the **margin**, that is the distance of the closest point in either class from this hyperplane should be maximized while correctly classifying/predicting all the training points. We can find that hyperplane using **Support Vector Machines** (SVM).

Binary Classification using Support Vector Machine (SVM)

At its essence, an SVM functions as a linear classifier, but it distinguishes itself by going beyond merely finding a separating hyperplane for data. It takes the additional step of seeking the optimal hyperplane, which doesn't just divide the data accurately but also maximizes the **margin***.

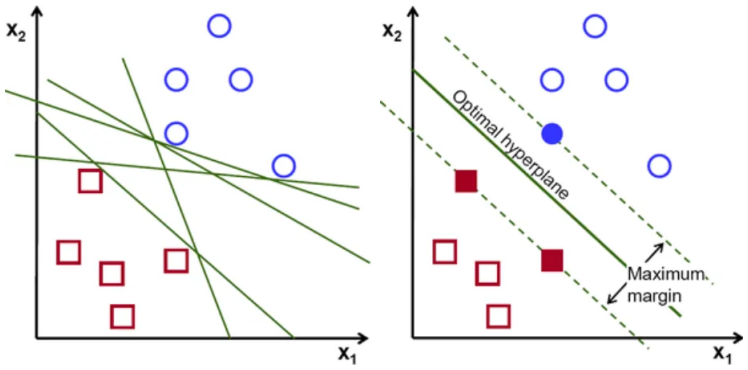

Figure 2: Different Hyperplane Seperators

In Figure 2, while all the lines in the left image successfully classify the data points, the line in the right image stands out as the optimal classifier because it boasts the widest margin, ensuring better robustness and generalization.

We know that,

$$\text{Margin } \gamma(w,b) = \min_i \frac{|w^T x_i + b|}{\|w\|}$$

So, our optimization problem becomes,

$$\max_{(w,b)} \min_i \frac{2|w^T x_i + b|}{\|w\|} \tag{1}$$

$$\text{such that, } \forall i \; y_i(w^T x_i + b) > 0$$

We can scale w and b to make,

$$\min_i |w^T X_i + b| = 1 \tag{2}$$

From equations (1) and (2), We can write,

$$\max_{(w,b)} \frac{2}{\|w\|}$$

$$\text{or, } \min_{(w,b)} \frac{\|w\|^2}{2}$$

$$\text{such that, } \forall i \; y_i(w^T x_i + b) \geq 1 \tag{3}$$

Equation (3) is our Final Optimization Problem.

Hard-Margin SVM

A Hard Margin Support Vector Machine (SVM) imposes a stringent condition where no data point is allowed to exist within the margin, and it must accurately classify all training data. This approach is most effective when dealing with linearly separable data, with a distinct and unfaltering boundary between the different classes.

You might wonder why our final optimisation hasn't included the constraint $\min_i |w^T X_i + b| = 1$. We don't need this constraint separately because it's already accounted for by the requirement $\forall i \; y_i(w^T x_i + b) \geq 1$. This latter condition ensures that the margin between the data points and the decision boundary is maximized and, as a result, incorporates the concept of $\min_i |w^T X_i + b| = 1$.

The blue line is the decision boundary in Figure 3. The orange and green lines pass through support vectors. Here, we are trying to maximize the margin, i.e. the distance between the orange and green lines.

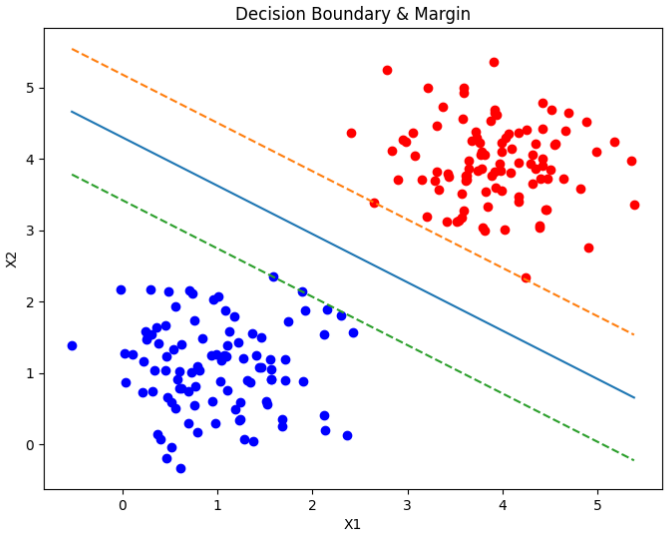

Figure 3: Example of Hard Margin SVM

Limitations

- Hard Margin SVMs exhibit high sensitivity to outliers or incorrectly labelled data points. Even a solitary outlier situated on the incorrect side of the decision boundary or within the margin can severely disrupt the model's ability to identify a valid hyperplane.
- Furthermore, Hard Margin SVMs face limitations when the data is not linearly separable, rendering them incapable of determining appropriate decision boundaries.
- Gradient descent is used to find the optimal value as there is no closed-form solution for (w, b).

Soft-Margin SVM

A Soft Margin Support Vector Machine (SVM) represents an advancement over the conventional "hard margin" SVM. Its purpose is to accommodate datasets lacking perfect linear separability and may include

some noise or data point overlap. In scenarios like these, imposing a stringent separation criterion, as in the case of the hard margin SVM, can result in overfitting or a failure to identify a suitable hyperplane.

Let there be some points which are misclassified.

$$\text{i.e. } \exists i \; y_i(w^T x_i + b) < 1$$

$$\text{or, } \exists i \; (1 - y_i(w^T x_i + b) > 0)$$

So, we will use $\max(1 - y_i(w^T x_i + b), 0)$ as penalty to track ***how badly this point is mislabelled***. This is also known as **Hinge Loss**.

So, our new optimization problem is to minimize

$$\min_{(w,b)} \frac{1}{2}\|w\|^2 + C\sum_i \max(1 - y_i(w^T x_i + b), 0)$$

Here C is the regularization factor and $\sum_i \max(1 - y_i(w^T x_i + b), 0)$ is regularization loss. C helps strike a balance between large margins and small training errors. To make this more formal, we introduce new variables $\mathcal{E}_i$ (referred to as **slack variables**), which will denote the penalty corresponding to each training instance.

So, Our Final Optimization Problem for Soft-Margin SVM is,

$$\min_{(w,b)} \left(\frac{\|w\|^2}{2} + C\sum_i \mathcal{E}_i\right)$$

$$\text{such that, } \forall i \; y_i(w^T x_i + b) \geq 1 - \mathcal{E}_i \text{ and } \forall i \; \mathcal{E}_i \geq 0$$

SVM Characteristics

- Soft margin SVMs offer increased robustness to outliers and noisy data, making them suitable for datasets with imperfect linear separability or class overlap. Consequently, they tend to provide improved generalization to new, unseen data compared to Hard-Margin SVMs.
- SVM can be used to classify non-linear data using a kernel. This will be elaborated further in the next lecture.
- A smaller value of C encourages a wider margin, allowing for more classification errors but potentially better generalization to unseen data.
- A larger C places a higher penalty on misclassifications, resulting in a narrower margin and potentially overfitting the training data.
- All the points for which $y_i(w^T x_i + b) = 1$ are called **support vectors** because these points determine the decision boundary in SVMs. Removing any of these points will alter the learned SVM decision boundary.