2.2.1.2 Support Vector Machines

The question now is whether and how one can select an optimal hyperplane. Our goal will be to find a hyperplane that has the maximum possible distance from the nearest points of the two classes. For this, we first derive an expression for the distance of a point to a hyperplane.

Proposition 2.2.1

Let $x_0 \in \mathbb{R}^d$ and $\mathcal{H} := \{x \in \mathbb{R}^d : \langle w, x \rangle + b = 0\}$ ($w \in \mathbb{R}^d$ is the normal vector (perpendicular to the hyperplane), $b \in \mathbb{R}$ is the bias term (offset from the margin)): Then it holds that

$$\operatorname{dist}(x_0, \mathcal{H}) := \inf\{||x_0 - x|| : x \in \mathcal{H}\} = \frac{|\langle w, x_0 \rangle + b|}{||w||}$$

(The shortest distance from a point x_0 to a hyperplane is the length of the perpendicular line from x_0 to $\mathcal{H}$ $\frac{|\langle w, x_0 \rangle + b|}{||w||}$. This arises from projecting the vector $x_0 - x$ (where $x \in \mathcal{H}$) onto the normal vector w
Proof. The proof will be provided in the exercise

We assume that the data are linearly separable and now consider the following optimization problem

2.2.5

$$\max\left\{M : w \in \mathbb{R}^d, b \in \mathbb{R}, y_i(\langle w, x_i \rangle + b \geq M||w||, \forall i = 1, \ldots, N\right\}$$

(The constraint ensures all points are classified correctly and are at least M away from the hyperplane).
The value M is the distance of the data points to the hyperplane that we want to maximize. The constraint $y_i(\langle w, x_i \rangle + b) \geq M||w||$ is, according to Proposition 2.2.1 equivalent to all points being correctly classified and having a distance of at least M from the hyperplane. Since we assume that the data are linearly separable, the constraints are at least satisfiable with $M = 0$. Since the constraints are scale-invariant, we can, without loss of generality, set $||w|| = \frac{1}{M}$ and obtain the problem

$$\max\left\{\frac{1}{||w||} : w \in \mathbb{R}^d, y_i(\langle w, x_i \rangle + b) \geq 1, \forall i = 1, \ldots, N\right\}$$

which is usually given in the following equivalent standard form (maximizing $\frac{1}{||w||}$, is equivalent to minimizing $||w||^2$)

2.2.6

$$\min\left\{\frac{1}{2}||w||^2 : w \in \mathbb{R}^d, b \in \mathbb{R}, y_i(\langle w, x_i\rangle + b) \geq 1, \forall i = 1, \ldots, N\right\}$$

(objective function $\frac{1}{2}||w||^2$ minimizes the norm of w, which maximizes the margin, the constraints ensure that every data point is correctly classified and at least distance 1 from the hyperplane)
This is a quadratic optimization problem with linear constraints and can be solved using classical numerical optimization methods. A classifier of the form $\hat{f}(x) = \text{sign}(\langle w, x\rangle + b)$, where w and b are solutions to 2.2.6, is called a **support vector machine** (SVM). The reason for this is that the classifier essentially depends only on the points that are closest to the hyperplane $\langle w, x\rangle + b$ and thus satisfy $y_i(\langle w, x_i\rangle + b) = 1$ (points that lie **exactly on the margin**), the so-called support vectors (points nearest to the hyperplane). If the points **that are further away**, i.e. $y_i(\langle w, x_i\rangle + b) > 1$, are slightly altered, the SVM remains unchanged.
So far, we have assumed that the data are linearly separable, which ensures that the constraints in 2.2.5 or 2.2.6 are satisfiable.
In most realistic applications, this assumption does not hold (there is usually overlap between classes, making constraints like above impossible to satisfy for all points)!
Nevertheless, we would like to find a hyperplane in these situations that violates the constraints for as few points as possible. To do this, we introduce non-negative slack variables $\xi_i \geq 0$ for $i = 1, \ldots, N$ which measure the deviation from $y_i(\langle w, x_i\rangle + b) \geq 1$ by replacing this constraint with $y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i$.

- If $\xi_i = 0$: The point is correctly classified outside the margin
- If $0 < \xi_i < 1$: The point is correctly classified inside the margin
- If $\xi_i \geq 1$: The point is misclassified
  Those slack variables are introduced one for each data point and relax the constraint to

At the same time, we want the slack variables to be as small as possible and therefore consider the following optimization problem

2.2.7

$$\min\left\{\frac{1}{N}\sum_{i=1}^{N}\xi_i + \frac{\lambda}{2}||w||^2 : w \in \mathbb{R}^d, b \in \mathbb{R}, \xi_i \in \mathbb{R},\right.$$

$$\left. y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i = 1, \ldots, N\right\}$$

- the term $\frac{1}{N}\sum_{i=1}^{N}\xi_i$ penalizes constraint violations - minimizing this keeps most points correctly classified

- the term $\frac{\lambda}{2}||w||^2$ maintains a large margin - we still want to keep the margin as wide as possible
- The parameter λ controls trade-off between margin-size and misclassification allowance.
  - Large λ prioritizes a large margin, allowing fewer misclassifications
  - Small λ allows more misclassifications to better fit noisy data

where $\lambda > 0$ is a selectable parameter. Formally, we see that in the linearly separable case 2.2.7 reduces to 2.2.6 in the limit as $\lambda \to \infty$
In practice 2.2.7 has an equivalent formulation as an optimization problem without constraints.

To do this, we note that we can replace the constraint $y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i$ with an equality constraint. This follows from the following simple argument:
If the constraint is satisfied with a strict inequality at the optimum, we could reduce ξ_i to create an equality while also reducing the objective function.
Thus, we can perform the variable substitution $\xi_i := 1 - y_i(\langle w, x_i\rangle + b)$.
From the non-negativity condition, we finally obtain $\xi_i = \max(0, 1 - y_i(\langle w, x_i\rangle + b))$. This term is also called **Hinge Loss**. Substituting this into 2.2.7 yields the equivalent problem

2.2.8

$$\min_{\substack{w\in\mathbb{R}^d \\ b\in\mathbb{R}}} \left\{ \frac{1}{N}\sum_{i=1}^{N} \max(0, 1 - y_i(\langle w, x_i\rangle + b)) + \frac{\lambda}{2}||w||^2 \right\}$$

Note that this is a convex problem in the variables w and b we are looking for. Unfortunately, the objective function, similar to the Perceptron loss in 2.2.3, is not differentiable in w and b when $y_i(\langle w, x_i\rangle + b) = 1$. We again compute subgradients of the function

$$f_i(w, b) := \max(0, 1 - y_i(\langle w, x_i\rangle + b)) + \frac{\lambda}{2}||w||^2$$

with respect to the two variables $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$:

$$\partial_w f_i(w, b) := -y_i x_i \mathbf{1}_{y_i(\langle w, x_i\rangle + b) < 1} + \lambda w$$

$$\partial_b f_i(w, b) := -y_i \mathbf{1}_{y_i(\langle w, x_i\rangle + b) < 1}$$

The objective function in 2.2.8 is given by $\frac{1}{N}\sum_{i=1}^{N} f_i(w, b)$, and thus we can now perform a deterministic or stochastic subgradient method:

$$\begin{cases} w_{k+1} := w_k - \frac{\tau}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \partial_w f_i(w_k, b_k) \\ b_{k+1} := b_k - \frac{\tau}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \partial_b f_i(w_k, b_k) \end{cases}$$

Here, $\mathcal{I} = \{1, \ldots, N\}$ corresponds to the deterministic case, and in the stochastic case, $\mathcal{I}$ is a batch of size $B \leq N$
The natural question arises: how can SVMs be generalized for classification of data that are not

or poorly linearly separable? One conceptual approach for this is so-called kernel SVMs, which replace all inner products $\langle x, y\rangle$ for $x, y \in \mathbb{R}^d$ with values of a symmetric function $k(x, y)$. Addressing this is beyond the scope of this lecture.

In fact, this approach is equivalent to embedding the data points $x_i \in \mathbb{R}^d$ into a higher-dimensional space and thus classifying points $\phi(x_i) \in \mathbb{R}^D$. A possible embedding would be $\phi : \mathbb{R}^d \to \mathbb{R}^{2d}$ with

$$\phi(x) := (x_1, \dots, x_d, x_1^2, \dots, x_d^2)^T$$

Here, x_i denotes the i-th coordinate of $x \in \mathbb{R}^d$ and not the i-th data point $x_i \in \mathbb{R}^d$!
A hyperplane in $\mathbb{R}^{2d}$ then, in the original coordinates, is expressed as:

$$0 = b + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} w_{d+1} x_i^2 = b + \langle w, x\rangle + x^T D x$$

with $w = (w_1, \dots, w_d)^T \in \mathbb{R}^d$ and $D := \operatorname{diag}(w_{d+1}, \dots, w_{2d}) \in \mathbb{R}^{d \times d}$.
This is a quadratic equation in $\mathbb{R}^d$ and the solutions are parabolas. In a similar way, arbitrarily complex classifiers can be generated in $\mathbb{R}^d$.
As another example, consider the embedding $\phi : \mathbb{R}^2 \to \mathbb{R}^3$ with $\phi(x) := (x_1, x_2, x_1^2 + x_2^2)$. The following hyperplane

$$0 = b + (x_1^2 + x_2^2)$$

with $b \leq 0$ is a circle equation in $\mathbb{R}^2$