Support Vector Machines

In this section we briefly explain the Support Vector Machines (SVM) classification algorithm for binary classification. We consider a set of training data $\{(\mathrm{x}_i, y_i)\}_{i=1}^m$ where each vector $\mathrm{x}_i \in \mathbb{R}^N$ is a collection of N features and is labelled $y_i \in \{-1, +1\}$. The basic idea of SVMs is that we can find some boundary in $\mathbb{R}^N$which "best" seperates the vectors labelled -1 from the vectors labelled $+1$. We first consider boundaries that are hyperplanes, before we go on to more complicated non-linear boundaries.

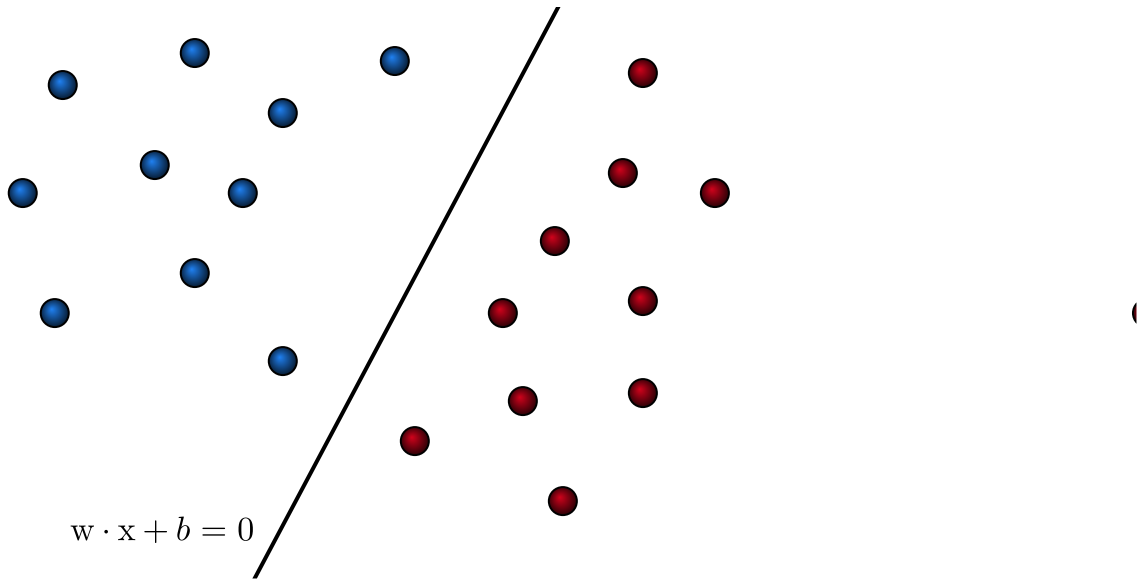
Linearly Separable Case

To start out simple we will first consider the case where the vectors x_i are *linearly seperable*. This means that we can find a hyperplane $\mathrm{w} \cdot \mathrm{x} + b = 0,\ \ \mathrm{w} \neq 0$ such that all the vectors labelled -1 fall on one side and all the vectors labelled $+1$ fall on the other. More specifically, we want to find w and b such that all the vectors labelled -1 satisfy $\mathrm{w} \cdot \mathrm{x}_i + b < 0$ and all the vectors labelled $+1$ to satisfy $\mathrm{w} \cdot \mathrm{x}_i + b \geq 0$. We can write this condition compactly as

$$y_i(\mathrm{w} \cdot \mathrm{x}_i + b) \geq 0,\ \ \forall i \in [m]$$

Once we've found w and b that satisfies this condition for all the training data, we can label *new* data using a linear classifier

$$h(\mathrm{x}) = \mathrm{sgn}(\mathrm{w} \cdot \mathrm{x} + b)$$



However, from the figure we realize that the are infinitely many hyperplanes which satisfies the condition, so which one do we choose? It might be tempting to use *any* hyperplane that satisfies the condition, but misclassification of new data is always a possibility. And intuitively, it would make sense that in order to avoid misclassification, we must not pick a hyperplane too close to either the vectors labelled -1 or $+1$, but square in the middle. This extra condition restricts us to a single hyperplane. An useful quantity for us in order to solve this problem is the so-called geometric margin.

Definition of the Geometric margin: The geometric margin $\rho_h(\mathrm{x})$ of a linear classifier $h : \mathrm{x} \mapsto \mathrm{sgn}(\mathrm{w} \cdot \mathrm{x} + b)$ is the shortest Euclidean distance between x and the hyperplane $\mathrm{w} \cdot \mathrm{x} + b = 0$.

To find an explicit expression of $\rho_h(\mathrm{x})$, consider the orthogonal projection $\widehat{\mathrm{x}}$ of the point x onto the hyperplane. The distance between $\widehat{\mathrm{x}}$ and x is $\rho_h(\mathrm{x})$ by definition. The direction of the vector $\mathrm{x} - \widehat{\mathrm{x}}$ is normal to the hyperplane, so it is given by the normal vector w. Thus we have

$$\rho_h(\mathrm{x}) \frac{\mathrm{w}}{|\mathrm{w}|} = \mathrm{x} - \widehat{\mathrm{x}}$$

Since the point $\widehat{\mathrm{x}}$ lies in the hyperplane, it satisfies the equation $\mathrm{w} \cdot \widehat{\mathrm{x}} + b = 0$, or $\mathrm{w} \cdot \widehat{\mathrm{x}} = -b$. Taking the dot product of w and both sides of the equation we get

$$\rho_h(\mathrm{x})\frac{\mathrm{w}\cdot\mathrm{w}}{|\mathrm{w}|} = \mathrm{w}\cdot\mathrm{x} - \mathrm{w}\cdot\widehat{\mathrm{x}}$$

$$\rho_h(\mathrm{x})|\mathrm{w}| = \mathrm{w}\cdot\mathrm{x} + b$$

$$\rho_h(\mathrm{x}) = \frac{\mathrm{w}\cdot\mathrm{x}+b}{|\mathrm{w}|}$$

Since $\rho_h(\mathrm{x})$ represents a distance it should be a positive function. Since $\mathrm{w}\cdot\mathrm{x}+b$ may be negative we can take the absolute value and redefine

$$\rho_h(\mathrm{x}) \equiv \frac{|\mathrm{w}\cdot\mathrm{x}+b|}{|\mathrm{w}|} = \frac{y_i(\mathrm{w}\cdot\mathrm{x}+b)}{|\mathrm{w}|}$$

Specifically, we'll use the smallest geometric margin over the training set $\rho_h \equiv \min_{i\in[m]}\rho_h(\mathrm{x}_i)$. The SVM solution is the hyperplane $\mathrm{w}\cdot\mathrm{x}+b=0$ which maximizes ρ_h under the constraint [EQUATION HERE]. The distance between this hyperplane and the closest training vector is given by

$$\rho = \max_{\mathrm{w},b}\ \min_{i\in[m]} \frac{y_i(\mathrm{w}\cdot\mathrm{x}+b)}{|\mathrm{w}|}\ :\ y_i(\mathrm{w}\cdot\mathrm{x}_i+b)\geq 0,\ \ \forall i\in[m]$$

An observation that will simplify the optimization problem substantially is that the geometric margin is invariant under multiplication of (w, b) by a positive scalar. This freedom allows us to choose (w, b) such that we can set $\min_{i\in[m]}y_i(\mathrm{w}\cdot\mathrm{x}+b)=1$. Under this constraint the optimization problem is

$$\rho = \max_{\mathrm{w},b}\frac{1}{|\mathrm{w}|}\ :\ y_i(\mathrm{w}\cdot\mathrm{x}_i+b)\geq 1,\ \ \forall i\in[m]$$

Since $|\mathrm{w}|^{-1}$ is maximized when $|\mathrm{w}|^2$ is minimized, which in contrast is infinitely differentiable everywhere and thus easier to deal with, we can reformulate the optimization problem as

$$\min_{\mathrm{w},b}\frac{1}{2}|\mathrm{w}|^2\ :\ y_i(\mathrm{w}\cdot\mathrm{x}_i+b)\geq 1,\ \ \forall i\in[m]$$

We can solve the optimization problem using the Lagrange multiplier method. The Lagrangian to be minimized is given by

$$\mathcal{L}(\mathrm{w}, b, \alpha) = \frac{1}{2}|\mathrm{w}|^2 - \sum_i^m \alpha_i[y_i(\mathrm{w} \cdot \mathrm{x}_i + b) - 1]$$

where the Lagrange multipliers are $\alpha_i \geq 0 \;\forall\; i \in [m]$. Taking the gradient of $\mathcal{L}$ with respect to w and the derivative with respect to b and setting both to zero we obtain

$$\nabla_{\mathrm{w}}\mathcal{L} = \mathrm{w} - \sum_i^m \alpha_i y_i \mathrm{x}_i = 0 \rightarrow \mathrm{w} = \sum_i^m \alpha_i y_i \mathrm{x}_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_i^m \alpha_i y_i = 0 \rightarrow \sum_i^m \alpha_i y_i = 0$$

Furthermore, according to Karush-Kuhn-Tucker's theorem (see Theorem B.30 in [BOOK REFERENCE HERE]), the following equation is satisfied at the minimum of $\mathcal{L}$.

$$\alpha_i[y_i(\mathrm{w} \cdot \mathrm{x}_i + b) - 1] = 0 \;\forall\; i \in [m] \rightarrow \alpha_i = 0 \vee y_i(\mathrm{w} \cdot \mathrm{x}_i + b) = 1 \;\forall\; i \in [m]$$

This means that all the α_i's corresponding to all *but* the vectors x_i closest to the hyperplane are zero. These vectors are called support vectors and they satisfy $\mathrm{w} \cdot \mathrm{x}_i + b = \pm 1$.

The *dual optimization problem* can be obtained by putting [EQUATIONS HERE] back into the Lagrangian and *maximizing* with respect to α. I.e

$$\max_\alpha \mathcal{L}(\alpha) = \max_\alpha \sum_i^m \alpha_i - \frac{1}{2}\sum_i^m \sum_j^m \alpha_i \alpha_j y_i y_j \mathrm{x}_i \cdot \mathrm{x}_j \;:\; \alpha_i \geq 0 \;\forall\; i \in [m],\; \sum_i^m \alpha_i y_i = 0$$

Once the optimal α has been found, we can calculate w using [EQUATION HERE]. If x_j is a support vector we can calculate b using the equation

$$y_j(\mathrm{w} \cdot \mathrm{x}_j + b) = 1 \;\rightarrow b = y_j - \mathrm{w} \cdot \mathrm{x}_j$$

where we've used that $(y_j)^{-1} = y_j$. Once the optimal w and b has been found, we classify new data according to the equation

$$h(\mathrm{x}) = \mathrm{sgn}\left(\sum_{i}^{m} \alpha_i y_i \mathrm{x}_i \cdot \mathrm{x} + b\right)$$

At this point we can make two important observations: The first observation is that since $\alpha_i = 0$ except for all the support vectors, the solution depends only on the support vectors. The second observation is that the solution [REFERENCE TO THE DUAL PROBLEM] depends only on inner products between vectors. This observation leads to a major simplification when we'll consider non-linear boundaries which, after all these mathematics for just the simplest form of boundary, would otherwise seem outright frightening.