The Support vector machine is a linear algorithm similar to the perceptron.

The Perceptron guarantees that you find a hyperplane if it exists

The SVM finds the **maximum** margin seperating hyperplane

Setting: Define a linear classifier

$$h(\vec{x}) = \text{sign}(\vec{w}^T\vec{x} + b)$$

and we assume a binary classification setting with labels $\{-1, 1\}$

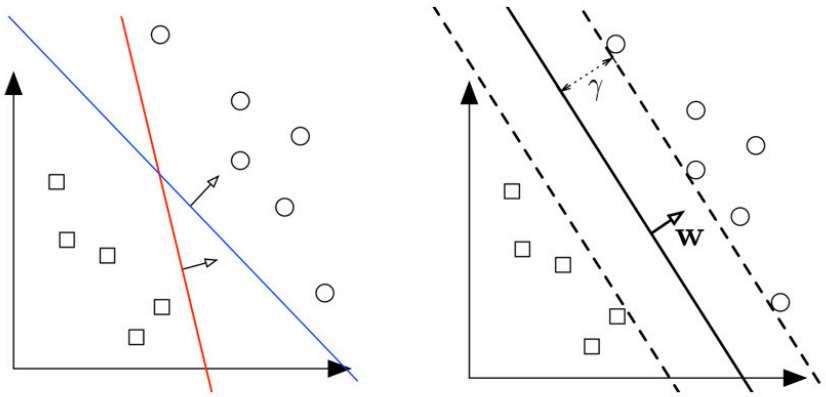


Figure 1: (Left:) Two different separating hyperplanes for the same data set. (Right:) The maximum margin hyperplane. The margin, γ, is the distance from the hyperplane (solid line) to the closest points in either class (which touch the parallel dotted lines).

If a dataset is seperable then there are infinitely many seperating hyperplanes.

SVM says the hyperplane w/ the maximum margin is the best one.

Margin

A hyperplane is defined through $\vec{w}, b$ as a set of points such that

$$\mathcal{H} = \{\vec{x} \mid \vec{w}^T\vec{x} + b = 0\}$$

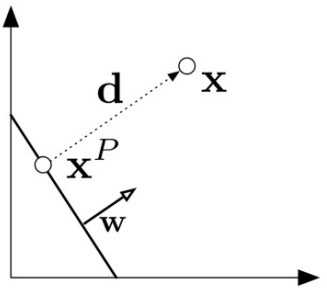
Let the margin γ be defined as the distance from the hyperplane to the closest point across both classes.

What is the distance from a point $\vec{x}$ to a hyperplane $\mathcal{H}$?

Consider some point $\vec{x}$.

Let $\vec{d}$ be a vector from $\mathcal{H}$ to $\vec{x}$ of minimum length.

Let $\vec{x}^P$ be a projection of $\vec{x}$ onto $\mathcal{H}$.



It then follows that:

$$\vec{x}^P = \vec{x} - \vec{d}$$

$$\vec{d} \parallel \vec{w} \Rightarrow \vec{d} = \alpha\vec{w} \text{ for } \alpha \in \mathbb{R}$$

$$\vec{x}^P \in \mathcal{H} \Rightarrow \vec{w}^T\vec{x}^P + b = 0$$

thus $\vec{w}^T\vec{x}^P + b = \vec{w}^T(\vec{x} - \vec{d}) + b = 0 = \vec{w}^T(\vec{x} - \alpha\vec{w}) + b = 0$

and $$\alpha = \frac{\vec{w}^T\vec{x} + b}{\vec{w}^T\vec{w}}$$

The length of $\vec{d}$ is then:

$$\|\vec{d}\|_2 = \sqrt{\vec{d}^T\vec{d}} = \sqrt{\alpha^2\vec{w}^T\vec{w}} = \alpha\sqrt{\vec{w}^T\vec{w}} = \frac{(\vec{w}^T\vec{x} + b)}{\sqrt{\vec{w}^T\vec{w}}}$$

Margin of $\mathcal{H}$ wl respect to D:

$$\gamma(\vec{w}, b) = \min_{\vec{x} \in D} \frac{(\vec{w}^T\vec{x} + b)}{\sqrt{\vec{w}^T\vec{w}}}$$

scale invariant by definition.

$\gamma(\beta\vec{w}, \beta b) = \gamma(\vec{w}, b) \quad \forall \beta \neq 0$

Max Margin Classifier

We formulate our search for the maximum margin seperating hyperplane as a constrained optimization problem.

The objective is to maximize the margin under the constraints that all data points must lie on the correct side of the hyperplane

$$\max_{\vec{w},b} \gamma(\vec{w},b) \text{ such that } \forall i \; y_i(\vec{w}^T\vec{x}_i + b) \geq 0$$

Plugging in for $\gamma(\vec{w},b)$:

$$\max_{\vec{w},b} \underbrace{\underbrace{\frac{1}{\|\vec{w}\|_2} \min_{\vec{x}_i \in D} |\vec{w}^T\vec{x}_i + b|}_{\gamma(\vec{w},b)}}_{\text{maximize margin}} \text{ such that } \forall i \; y_i(\vec{w}^T\vec{x}_i + b) \geq 0$$

Because the hyperplane is scale invariant, we can fix the scale of $\vec{w}, b$ **ANYWAY** we want!

Let's be clever about it, and choose it such that

$$\min_{\vec{x}_i \in D} |\vec{w}^T\vec{x}_i + b| = 1$$

We can add this rescaling as an equality constraint.

Then our objective becomes

$$\max_{\vec{w},b} \frac{1}{\|\vec{w}\|_2} \cdot 1 = \min_{\vec{w},b} \|\vec{w}\|_2 = \min_{w,b} \vec{w}^T\vec{w}$$

$f(z) = z^2$ monotonically ↑ for $z \geq 0$ and $\|w\| \geq 0$.
So $\vec{w}$ maximizing $\|w\|_2$ also maximizes $\vec{w}^T\vec{w}$

The new optimization becomes:

$$\min_{\vec{w},b} \vec{w}^T\vec{w} \quad \forall i,\ y_i(\vec{w}^T\vec{x}_i+b) \geq 0 \quad \text{s.t.} \quad \min_i |\vec{w}^T\vec{x}_i+b| = 1$$

we can show that for the optimal solution the above is equivalent to

$$\min_{\vec{w},b} \vec{w}^T\vec{w} \quad \text{s.t} \quad \forall i\ y_i(\vec{w}^T\vec{x}_i+b) \geq 1$$

This is now a quadratic optimization problem

The objective is quadratic and the constraints are all linear.
It has a unique solution whenever one exists.
It also has a nice interpretation: find the simplest hyperplane (simpler means smaller $\vec{w}^T\vec{w}$) such that all inputs lie at least 1 unit away from the hyperplane on the correct side.

Support Vectors

For the new optimal $\vec{w}, b$ pair, some training points will have tight constraints, i.e.

$$y_i(\vec{w}^T\vec{x}_i+b) = 1$$

this must be the case, because if for all training points we had a strict $>$ inequality, it would be possible to scale down both parameters $\vec{w}, b$ until the constraints are tight and obtained an even lower objective value.

We refer to these points as <u>support vectors.</u>

Support vectors are special because they are the training points that define the maximum margin of the hyperplane to the data set and they therefore determine the shape of the hyperplane.

SVM with Soft Constraints

If the data is low dimensional it is often the case that there is no seperating hyperplane between the two classes.

In this case, there is **NO** solution to the optimization problem stated above. We can fix this by allowing the constraints to be violated ever so slight w/ the introduction of slack variables:

$$\min_{\vec{w},b} \vec{w}^T\vec{w} + C\sum_{i=1}^{n}\xi_i \quad \text{s.t. } \forall i\ y_i(\vec{w}^T\vec{x}_i + b) \geq 1 - \xi_i \quad \forall i\ \xi_i \geq 0$$

The slack variable ξ_i allows the input $\vec{x}_i$ to be closer to the hyperplane (or even be on the wrong side), but there is a penalty in the objective function for such "slack".

If C is very large, the SVM becomes very strict and tries to get all points to be on the right side of the hyperplane.

If C is very small, the SVM becomes very loose and may "sacrifice" some points to obtain a simpler (i.e lower $\|\vec{w}\|_2$) solution.

Unconstrained Formulation

Let us consider the value of ζ_i for the case $C \neq 0$.

Because the objective will always try to minimize ζ_i as much as possible, the equation must hold as an equality and we have:

$$\zeta_i = \begin{cases} 1 - y_i(\vec{w}^T \vec{x}_i + b) & \text{if } y_i(\vec{w}^T \vec{x}_i + b) < 1 \\ 0 & \text{if } y_i(\vec{w}^T \vec{x}_i + b) \geq 1 \end{cases},$$

which is equivalent to the closed form

$$\zeta_i = \max\left(1 - y_i(\vec{w}^T \vec{x}_i + b), 0\right)$$

If we plug this closed form into the objective of our SVM optimization problem, we obtain the following **unconstrained** version as loss function and regularizer:

$$\min_{\vec{w}, b} \; \underbrace{\vec{w}^T \vec{w}}_{L_2 \text{ regularizer}} + C \sum_{i=1}^{n} \underbrace{\max\left[1 - y_i(\vec{w}^T \vec{x}_i + b), 0\right]}_{\text{hinge-loss}}$$

This formulation allows us to optimize the SVM parameters $(\vec{w}, b)$ just like logistic regression. The only difference is that we have the hinge-loss instead of the logistic loss.

See Plots Below

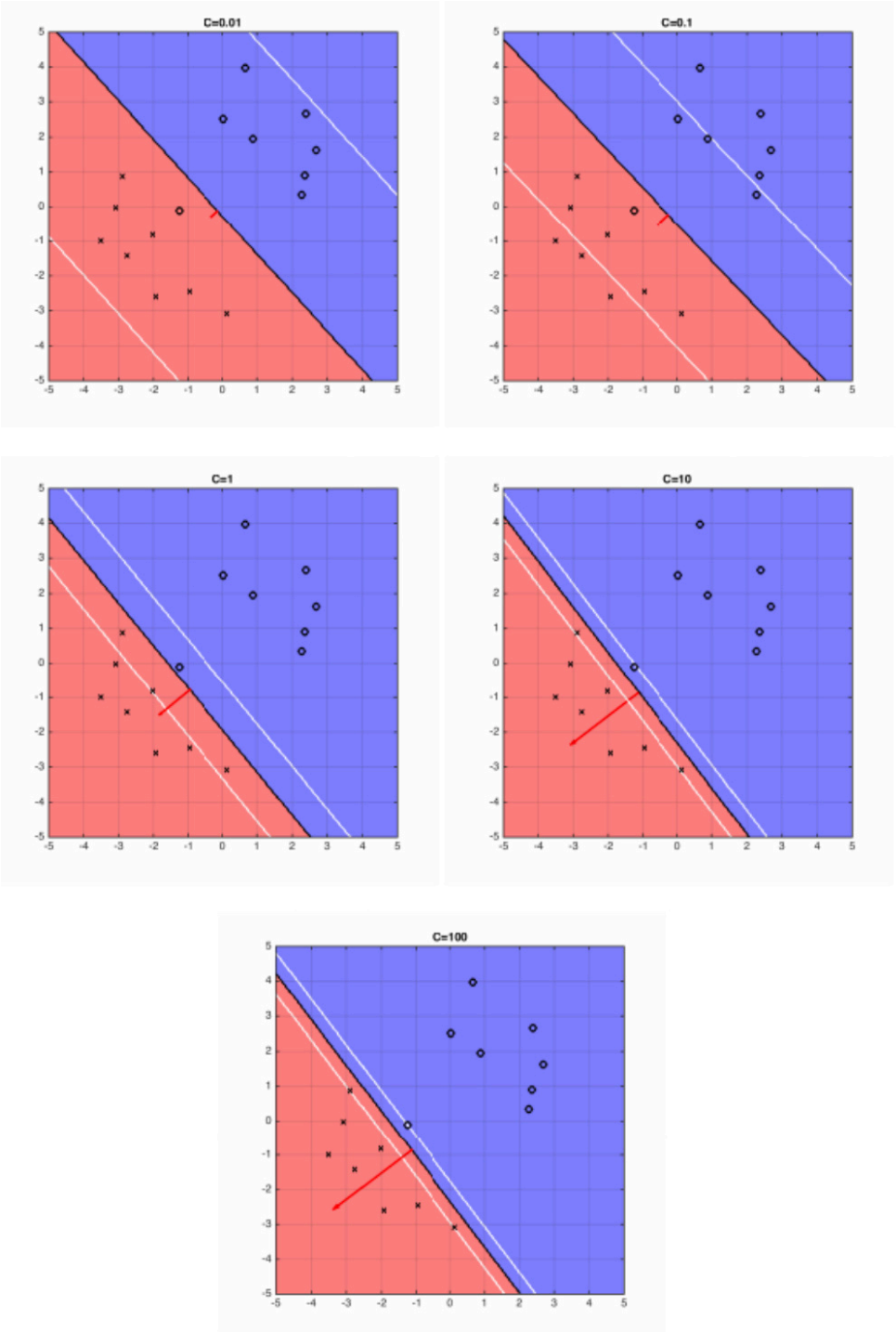


Figure 2: The five plots above show different boundary of hyperplane and the optimal hyperplane separating example data, when C=0.01, 0.1, 1, 10, 100.