

Support Vector Machines

⇒ SVMs are among the best "off-the-shelf" supervised learning algorithms.

* Margins

⇒ Consider logistic regression, where the probability $P(y=1|x;\theta)$ is modeled by $h_\theta(x) = g(\theta^T x)$

If $h_\theta(x) \geq 0.5 \rightarrow y=0$ (Prediction)
or

$$\theta^T x \geq 0$$

⇒ The larger $\theta^T x$ is, the larger also is $h_\theta(x)$ and thus also the higher our degree of "confidence" that the label is 1.

⇒ Given a training set, we'd have found a good fit to the training data if we find θ so that:

$$\theta^T x^{(i)} \geq 0 \quad \text{if } y^{(i)} = 1$$

$$\text{and } \theta^T x^{(i)} < 0 \quad \text{if } y^{(i)} = 0$$

* Notation

⇒ We will be considering a linear classifier for a binary classification problem with labels y and features x .

⇒ We will use $y \in \{-1, 1\}$ to denote class labels.

⇒ Also, in addition to parameterizing our linear classifier with the vector θ , we will use parameters w, b and write our classifier as

$$h_{w,b}(x) = g(w^T x + b)$$

⇒ Here, $g(z) = 1 \text{ if } z \geq 0 \text{ & } g(z) = -1 \text{ otherwise.}$

* Functional and geometric margins

Functional margin

→ Given a training example $(x^{(i)}, y^{(i)})$, we define the functional margin of (w, b) with respect to the training example

$$\hat{y}^{(i)} = y^{(i)}(w^T x^{(i)} + b)$$

→ Given g , if we replace w with $2w$ & b with $2b$, then since $g(w^T x + b) = g(2w^T x + 2b)$ this would not change $h_{w,b}(x)$ at all.

→ So $h_{w,b}(x)$ depends only on the sign, but not on the magnitude of $w^T x + b$.

However, replacing (w, b) with $(2w, 2b)$ also results in multiplying our functional margin by 2.

It might therefore make sense to impose some sort of normalization condition.

$$(\omega, b) \implies \left(\frac{\omega}{\|\omega\|_2}, \frac{b}{\|b\|_2} \right)$$

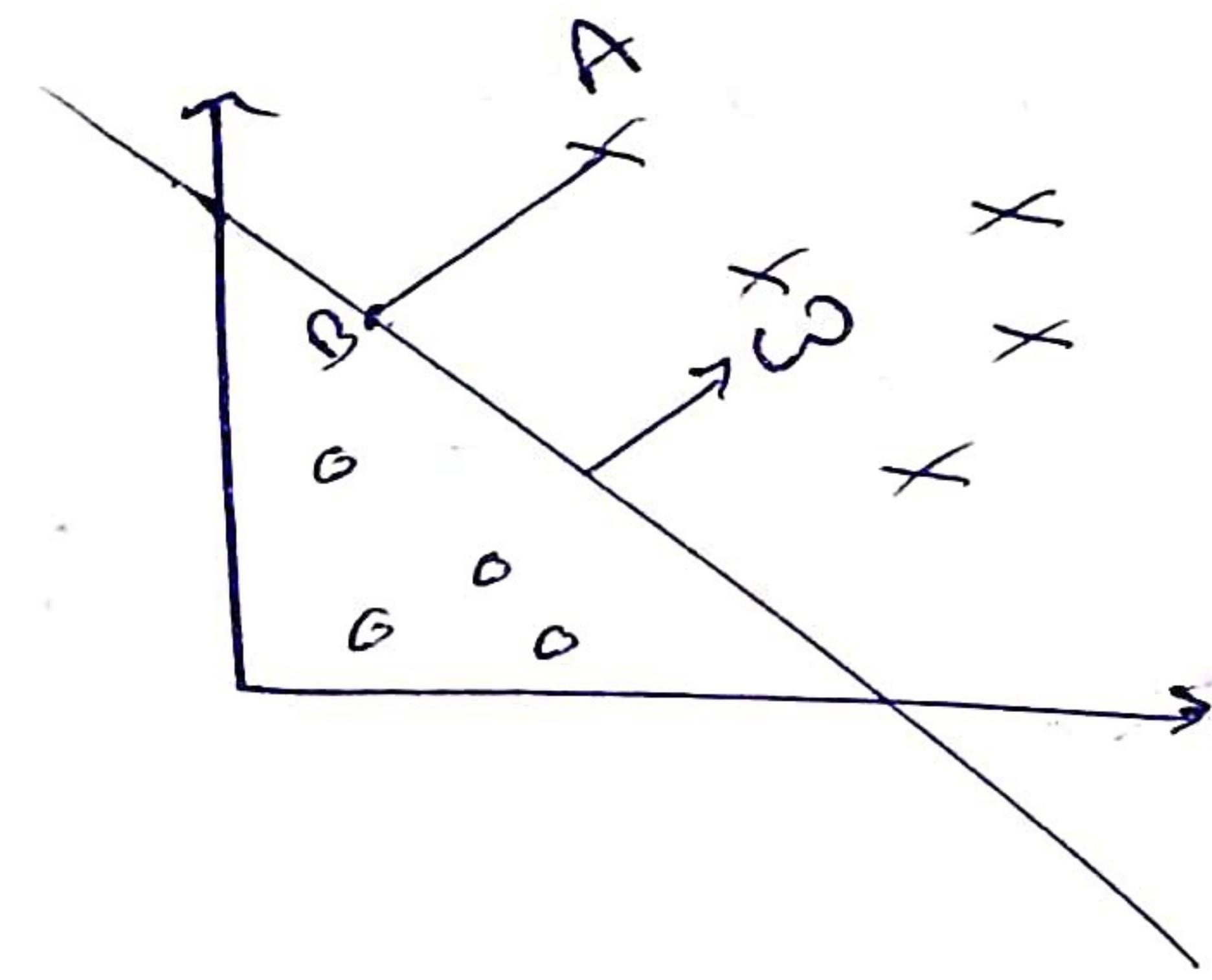
\Rightarrow Given a training set $S = \{(x^{(i)}, y^{(i)}) ; i=1, \dots, m\}$
 we also define the functional margin (ω, b) with respect to S as the smallest of the functional margin of the individual training examples.

$$\hat{y} = \min_{i=1, \dots, m} \hat{y}^{(i)}$$

Geometric margin

\Rightarrow Note: ω is orthogonal to the separating hyperplane.

\Rightarrow Let A represent the input $x^{(i)}$ and $y^{(i)} = 1$



$\Rightarrow y^{(i)}$ distance of A from decision boundary.

Given by AB .

$$(x^{(i)} - y^{(i)} \frac{\omega}{\|\omega\|_2})$$

Point B

$$\omega^T (x^{(i)} - y^{(i)} \frac{\omega}{\|\omega\|_2}) + b = 0 \quad \left\{ \begin{array}{l} \text{Point B lies} \\ \text{on the hyperplane} \end{array} \right\}$$

$$y^{(i)} = \left(\frac{\omega}{\|\omega\|_2}\right)^T x^{(i)} + \frac{b}{\|\omega\|_2} \quad \left\{ \begin{array}{l} \text{For positive} \\ \text{training example} \end{array} \right\}$$

⇒ More generally, we define the geometric margin of (ω, b) with respect to a training example $(x^{(i)}, y^{(i)})$ to be

$$\boxed{y^{(i)} = y^{(i)} \left(\left(\frac{\omega}{\|\omega\|_2} \right)^T x^{(i)} + \frac{b}{\|\omega\|_2} \right)}$$

⇒ If we replace ω with 2ω and b with $2b$, then the geometric margin does not change.

⇒ Finally, given a training set $S = \{(x^{(i)}, y^{(i)}); i=1, \dots, m\}$, we also define the geometric margin of (ω, b) with respect to S to be the smallest of the geometric margins on the individual training examples:

$$\boxed{Y = \min_{i=1, \dots, m} y^{(i)}}$$

* The Optimal margin classifier

⇒ Given a training set, try to find a decision boundary that maximizes the (geometric) margins. Since this would reflect a very confident set of predictions on the training set.

→ For now, we will assume that we are given a training set that is linearly separable.

↳ It is possible to separate the positive & negative example using some separating hyperplane.

⇒ To achieve the maximum geometric margin, we can pose the following optimization problem:

$$\max_{\gamma, \omega, b} \gamma$$

$$\text{st. } y^{(i)}(\omega^T x^{(i)} + b) \geq \gamma \quad i=1, \dots, m$$

$$\|\omega\|=1$$

⇒ If we could solve the optimization problem above, we'd be done.

↳ But the $\|\omega\|=1$ constraint is non-convex one, and this problem certainly isn't in any format that we can plug into standard optimization software to solve.

⇒ So, let's try transforming the problem into a nicer one.

⇒ Consider,

$$\max_{\gamma, \omega, b} \frac{\gamma}{\|\omega\|}$$

$$\text{st. } y^{(i)}(\omega^T x^{(i)} + b) \geq \gamma \quad i=1, \dots, m$$

①
⇒ We've gotten rid of the constraint $\|\omega\|=1$. The downside is that we have a nasty (again non-convex) objective $\frac{1}{\|\omega\|_2}$ function.

⇒ Recall, we can add an arbitrary scaling constraint on $\omega \& b$ without changing anything.

↳ functional margin being multiplied by that same constant.

⇒ Maximizing $\frac{1}{\|\omega\|}$ ⇒ minimizing $\|\omega\|^2$

⇒ We now have the following optimization problem

$$\boxed{\begin{array}{l} \text{min}_{y, \omega, b} \frac{1}{2} \|\omega\|^2 \\ \text{st. } y^{(i)}(\omega^T \omega^{(i)} + b) \geq 1 \quad i=1, \dots \end{array}}$$

Optimization problem with a
Convex Quadratic Objective &
Only Linear Constraints

Can be Solved using Commercial
Quadratic Programming Code
(QP)

* Lagrange duality

⇒ Consider a problem of the following form:

$$\min_{\omega} f(\omega)$$

$$\text{s.t. } h_i(\omega) = 0 \quad i=1, \dots, l$$

⇒ Let the Lagrangian be:

$$L(\omega, \beta) = f(\omega) + \sum_{i=1}^l \beta_i h_i(\omega)$$

Lagrange multipliers

⇒ $\frac{\partial L}{\partial \omega_i} = 0 ; \frac{\partial L}{\partial \beta_i} = 0$ and solve for ω and β .

⇒ In this Section, we will generalize this to constrained optimization problems, in which we may have inequality as well as equality constraints.

⇒ Consider the following, which we'll call the Primal optimization problem:

$$\min_{\omega} f(\omega)$$

$$\text{s.t. } g_i(\omega) \leq 0, \quad i=1, \dots, K$$

$$h_i(\omega) = 0 \quad i=1, \dots, l$$

⇒ To solve it, we start by defining the generalized Lagrangian

$$L(\omega, \alpha, \beta) = f(\omega) + \sum_{i=1}^K \alpha_i g_i(\omega) + \sum_{i=1}^l \beta_i h_i(\omega)$$

⇒ Here, the α_i 's and B 's are the Lagrange multipliers.

$$\Theta_p(\omega) = \max_{\alpha, B; \alpha_i \geq 0} L(\omega, \alpha, B)$$

\downarrow
 Stands for
 Primal

⇒ If ω violates any of the primal constraints, then you should be able to verify that

$$\Theta_p(\omega) = \infty$$

⇒ Conversely, if the constraints are indeed satisfied for a particular value of ω then $\Theta_p(\omega) = f(\omega)$

$$\Theta_p(\omega) = \begin{cases} f(\omega) & \text{if } \omega \text{ satisfies primal} \\ & \text{constraints} \\ \infty & \text{otherwise} \end{cases}$$

⇒ Hence if we consider the minimization problem

$$\min_{\omega} \Theta_p(\omega) = \min_{\omega} \max_{\alpha, B; \alpha_i \geq 0} L(\omega, \alpha, B)$$

We see that it is the same problem our original, primal problem.

$$\text{Let } P^* = \min_{\omega} \Theta_p(\omega)$$

⇒ Now, lets look at a slightly different problem.

$$\Theta_D(\alpha, B) = \min_{\omega} L(\omega, \alpha, B)$$

\downarrow
 Dual

\Rightarrow We can now pose the dual optimization problem:

$$\max_{\alpha, \beta: \alpha_i > 0} \Theta_D(\alpha, \beta) = \max_{\alpha, \beta, \omega} \min_{\omega} L(\omega, \alpha, \beta)$$

$$d^* = \max_{\alpha, \beta, \omega} \Theta_D(\omega)$$

$$d^* \leq p^*$$

\Rightarrow Under certain condition $d^* = p^*$

Karush-Kuhn-Tucker (KKT) Condition



$\left\{ \begin{array}{l} \omega^*, \alpha^*, \beta^* \text{ Satisfy the KKT} \\ \text{Condition} \end{array} \right\}$

* Optimal margin Classifiers

⇒ Optimization problem for finding the optimal margin classifier:

Convex
Optimization
Problem

$$\text{Min}_{\omega, b} \frac{1}{2} \|\omega\|^2$$

$$\text{St. } y^{(i)} (\omega^T x^{(i)} + b) \geq 1 \quad i = 1, \dots, m$$

Choose $\|\omega\| = Y$ as scaling factor

$$\text{Max}_{Y, \omega, b} Y$$

$$\text{St. } \frac{y^{(i)} (\omega^T x^{(i)} + b)}{\|\omega\|} \geq Y \quad i = 1, \dots, m$$

This optimization problem
can be solved using commercial
Quadratic Programming Code
(QP)