

STA 314: Statistical Methods for Machine Learning I

Lecture - Support Vector Machine

Xin Bing

Department of Statistical Sciences
University of Toronto

Linear decision boundaries

In binary classification problems, we have seen examples of classifiers with decision boundaries **linear** in the feature space.

- Logistic regression:

$$\log \frac{\mathbb{P}(Y = 1 \mid X = \mathbf{x})}{\mathbb{P}(Y = 0 \mid X = \mathbf{x})} = \beta_0 + \boldsymbol{\beta}^T \mathbf{x}.$$

Hence, $\mathbb{P}(Y = 1 \mid X = \mathbf{x}) \geq \mathbb{P}(Y = 0 \mid X = \mathbf{x})$ if and only if

$$\beta_0 + \boldsymbol{\beta}^T \mathbf{x} \geq 0.$$

The decision boundary is

$$\left\{ \mathbf{x} \in \mathbb{R}^p : \beta_0 + \boldsymbol{\beta}^T \mathbf{x} = 0 \right\}.$$

Linear decision boundaries

- LDA:

$$\delta_k(\mathbf{x}) = \mathbf{x}^\top \Sigma^{-1} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k^\top \Sigma^{-1} \boldsymbol{\mu}_k + \log \pi_k, \quad \forall k \in \{0, 1\}.$$

Hence, $\delta_1(\mathbf{x}) \geq \delta_0(\mathbf{x})$ if and only if

$$\left(\mathbf{x} - \frac{\boldsymbol{\mu}_0 + \boldsymbol{\mu}_1}{2} \right)^\top \Sigma^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) + \log \frac{\pi_1}{\pi_0} \geq 0.$$

The decision boundary is

$$\left\{ \mathbf{x} \in \mathbb{R}^p : \alpha_0 + \boldsymbol{\alpha}^\top \mathbf{x} = 0 \right\}$$

for some α_0 and $\boldsymbol{\alpha} \in \mathbb{R}^p$.

A general formulation of linear classifiers

Binary classification: predicting a target with two values, $y \in \{-1, +1\}$, (notational change from the past).

- Consider the linear decision boundary

$$\mathbf{w}^T \mathbf{x} + b = 0$$

for some weights $\mathbf{w} \in \mathbb{R}^p$ and $b \in \mathbb{R}$.

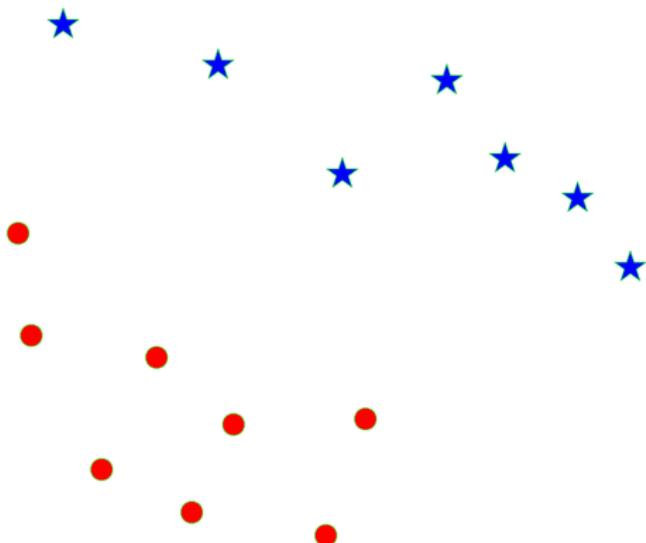
- A good decision boundary should satisfy: for a given point (\mathbf{x}, y) ,

$$\mathbf{w}^T \mathbf{x} + b > 0, \quad \text{if } y = 1$$

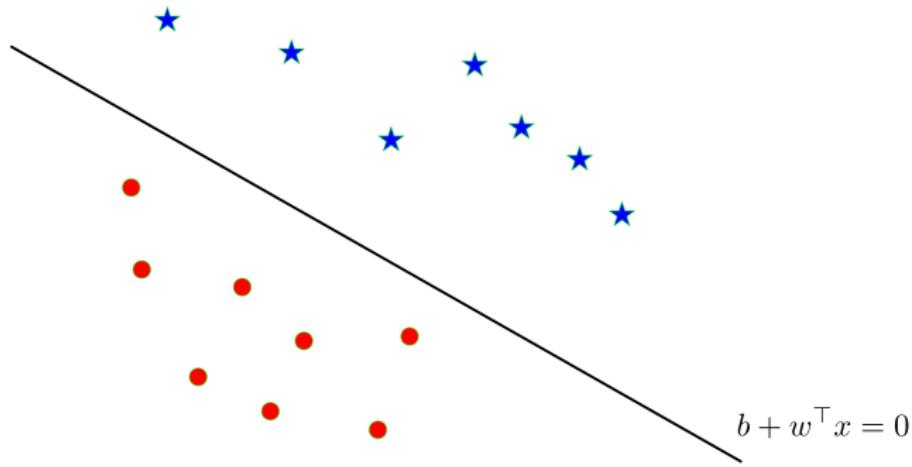
$$\mathbf{w}^T \mathbf{x} + b < 0, \quad \text{if } y = -1.$$

Separating Hyperplanes

Suppose we are given these data points from two different classes and want to find a linear classifier that separates them.



Separating Hyperplanes



- The decision boundary is a line in \mathbb{R}^2
- $\{\mathbf{x} \in \mathbb{R}^p : \mathbf{w}^\top \mathbf{x} + b = 0\}$ is a hyperplane in $(p - 1)$ dimensional space.

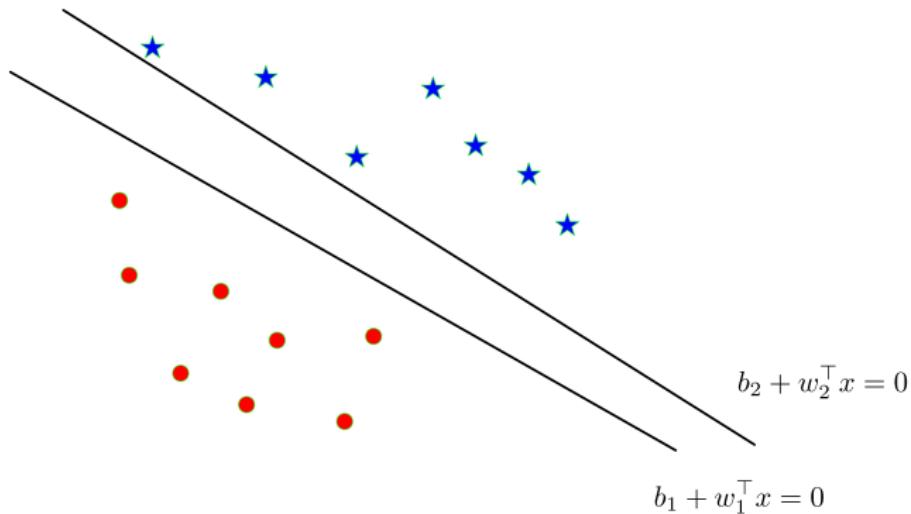
Simple Intuition and Potential Issues

To correctly classify all points we require that

$$\text{sign}(\mathbf{w}^\top \mathbf{x}_i + b) = y_i \quad \text{for all } i \in [n].$$

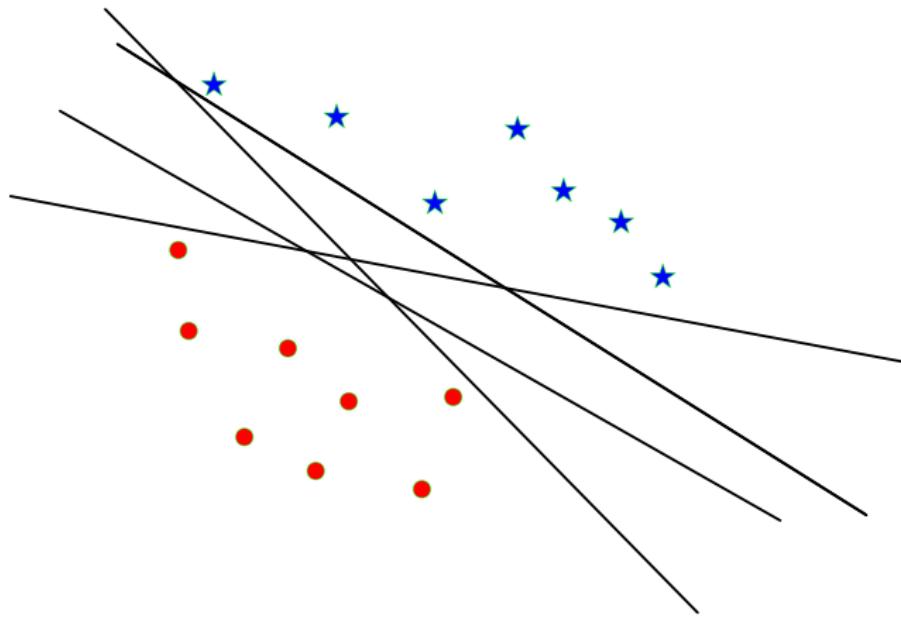
- We should find \mathbf{w} and b to meet the above goal.
- However:
 - ▶ When the data is separable, there exists multiple solutions of \mathbf{w} and b . Which to choose?
 - ▶ When the data is not separable, it is infeasible.

Separable Cases



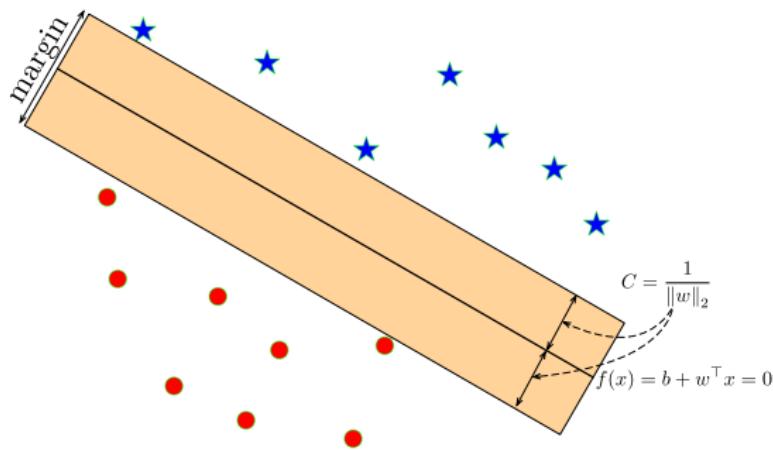
- There are multiple separating hyperplanes, determined by different parameters (\mathbf{w}, b) .

Separable Cases



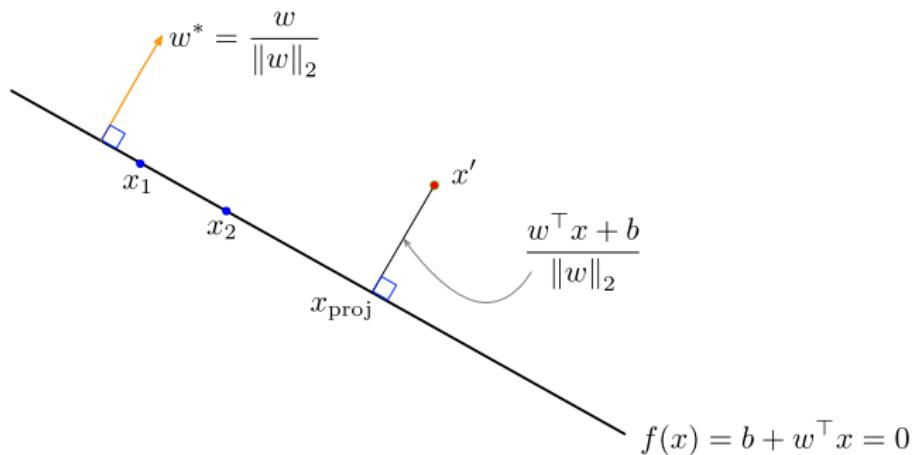
Optimal Separating Hyperplane

Optimal Separating Hyperplane: A hyperplane that separates two classes and maximizes the distance to the closest point from either class, i.e., maximize the **margin** of the classifier.



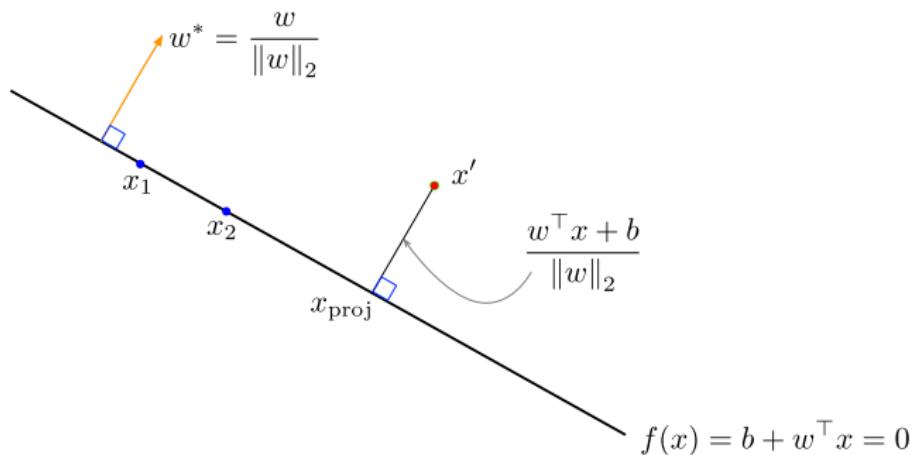
Intuitively, ensuring that a classifier is not too close to any data points leads to better generalization on the test data.

Geometry of Points and Planes



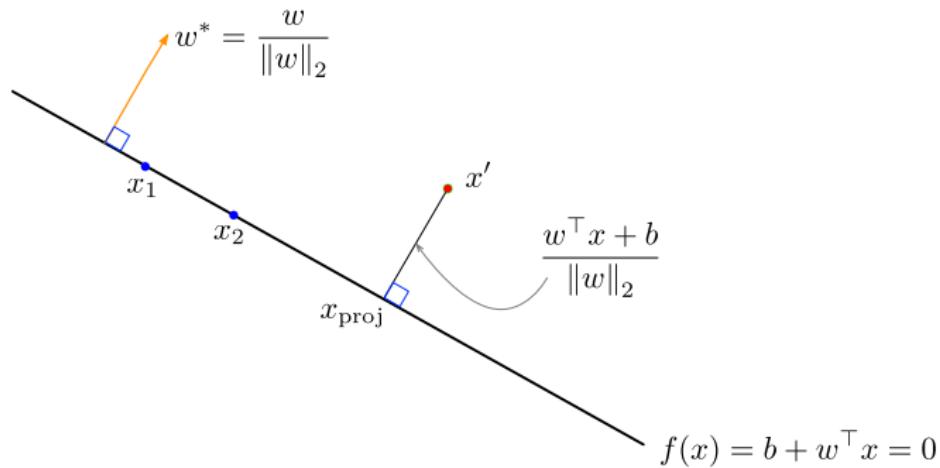
- Recall that the decision hyperplane is orthogonal (perpendicular) to \mathbf{w} . I.e., for any two points \mathbf{x}_1 and \mathbf{x}_2 on the decision hyperplane we have that $\mathbf{w}^\top (\mathbf{x}_1 - \mathbf{x}_2) = 0$.

Geometry of Points and Planes



- The vector $\mathbf{w}^* = \frac{\mathbf{w}}{\|\mathbf{w}\|_2}$ is a unit vector pointing in the same direction as \mathbf{w} .
- The same hyperplane could equivalently be defined in terms of \mathbf{w}^* .

Geometry of Points and Planes



- Question: how to compute the distance from a point x' to the hyperplane $\{x : b + w^\top x = 0\}$.

Distance to a Given Hyperplane

Fix the point \mathbf{x}' as well as \mathbf{w} and b which determine the hyperplane.

- Take the closest point \mathbf{x}_{proj} on the hyperplane, which satisfies

$$\mathbf{w}^\top \mathbf{x}_{\text{proj}} + b = 0.$$

- We know that $\mathbf{x}' - \mathbf{x}_{\text{proj}}$ is parallel to $\mathbf{w}^* = \mathbf{w}/\|\mathbf{w}\|_2$
- The distance is

$$\begin{aligned}\|\mathbf{x}' - \mathbf{x}_{\text{proj}}\|_2 &= \left\| (\mathbf{x}' - \mathbf{x}_{\text{proj}})^\top \frac{\mathbf{w}}{\|\mathbf{w}\|_2} \right\| \\ &= \frac{\left| \mathbf{w}^\top \mathbf{x}' - \mathbf{w}^\top \mathbf{x}_{\text{proj}} \right|}{\|\mathbf{w}\|_2} = \frac{\left| \mathbf{w}^\top \mathbf{x}' + b \right|}{\|\mathbf{w}\|_2}\end{aligned}$$

Maximizing Margin as an Optimization Problem

We want to choose \mathbf{w} and b such that:

- correctly classify all points:

$$\text{sign}(\mathbf{w}^\top \mathbf{x}_i + b) = y_i \quad \text{for all } i \in [n]$$

- The smallest distance of \mathbf{x}_i to $\{\mathbf{x} : \mathbf{w}^\top \mathbf{x} + b = 0\}$,

$$\frac{|\mathbf{w}^\top \mathbf{x}_i + b|}{\|\mathbf{w}\|_2},$$

is as large as possible.

This leads to the max-margin objective:

$$\begin{aligned} & \max_{\mathbf{w}, b} \min_{i \in [n]} \frac{|\mathbf{w}^\top \mathbf{x}_i + b|}{\|\mathbf{w}\|_2} \\ & \text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 0, \quad \text{for all } i \in [n] \end{aligned}$$

Maximizing Margin as an Optimization Problem

Equivalently,

$$\begin{aligned} \max_{\mathbf{w}, b} \quad & \min_{i \in [n]} \frac{y_i^2 (\mathbf{w}^\top \mathbf{x}_i + b)^2}{\|\mathbf{w}\|_2^2} \\ \text{s.t.} \quad & y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq 0, \quad \text{for all } i \in [n] \end{aligned}$$

More compactly,

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{\|\mathbf{w}\|_2^2}{M^2} \\ \text{s.t.} \quad & y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq M, \quad \text{for all } i = 1, \dots, n \end{aligned}$$

The constraints are called the **margin constraints**.

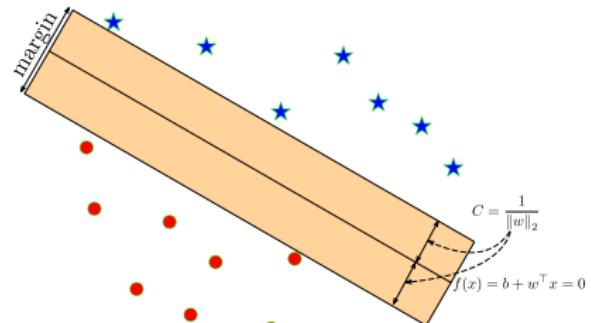
W.l.o.g. we can set $M = 1$. (Why?)

Maximizing Margin as an Optimization Problem

Max-margin objective:

$$\min_{\mathbf{w}, b} \|\mathbf{w}\|_2^2$$

$$\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad i = 1, \dots, n$$



- Intuitively, if the margin constraint is not tight for \mathbf{x}_i , we could remove \mathbf{x}_i from the training set and the optimal hyperplane would be the same.¹
- The important training points are those with equality constraints, and are called **support vectors**.
- Hence, this algorithm is called the (hard-margin) **Support Vector Machine (SVM)**. SVM-like algorithms are often called **max-margin** or **large-margin**.

¹This can be rigorously shown via the K.K.T. conditions.

Computation of the hard-margin SVM

Primal-formulation:

$$\min_{\mathbf{w}, b} \|\mathbf{w}\|_2^2$$

$$\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad i = 1, \dots, n$$

- Convex, in fact, a quadratic program. (Stochastic) Gradient descent can be directly used.
- In practice, it is more common to solve this optimization problem based on its dual formulation.

Dual-formulation of the hard-margin SVM

For $\alpha_i \geq 0$ for all $i = 1, \dots, n$, write the Lagrangian function

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \|\mathbf{w}\|_2^2 + \sum_{i=1}^n \alpha_i [1 - y_i (\mathbf{w}^\top \mathbf{x}_i + b)],$$

Taking the derivative w.r.t. \mathbf{w} and b yields

$$\mathbf{w} = \frac{1}{2} \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i, \quad \sum_{i=1}^n \alpha_i y_i = 0.$$

Plugging into $L(\mathbf{w}, b, \boldsymbol{\alpha})$ yields

$$\begin{aligned} & \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j + \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j - b \sum_{i=1}^n \alpha_i y_i \\ &= \sum_{i=1}^n \alpha_i - \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j. \end{aligned}$$

Dual-formulation of the hard-margin SVM

The dual problem is

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad i = 1, \dots, n. \end{aligned}$$

- This is a quadratic program in α and can be easily solved.
- It only depends on $\mathbf{x}_i^\top \mathbf{x}_j$, which is very convenient to extend to cases where some basis functions $\phi(\mathbf{x}_i)$ are used (so-called **kernel trick**.)

K.K.T. conditions

The K.K.T. conditions ensure the following relationships between the primal and dual formulations.

- Their optimal objective values are equal.
- The optimal solutions $\hat{\mathbf{w}}$ and $\hat{\alpha}$ satisfy

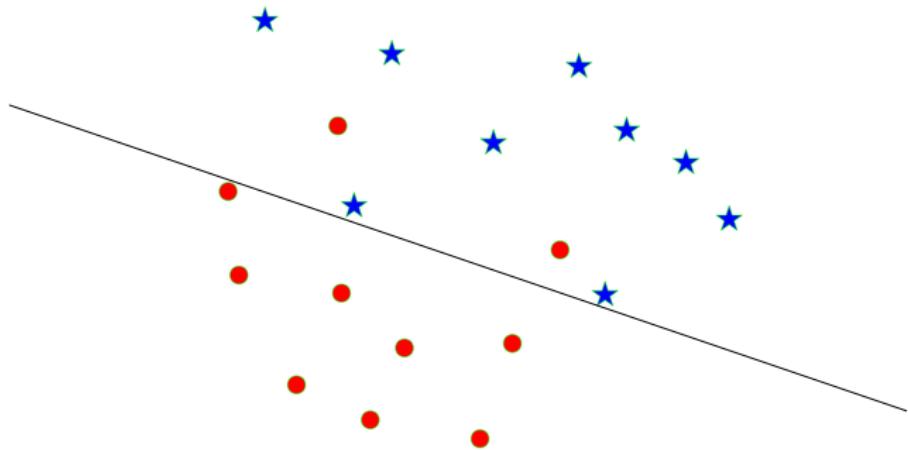
$$\hat{\mathbf{w}} = \frac{1}{2} \sum_{i=1}^n \hat{\alpha}_i y_i \mathbf{x}_i, \quad \begin{aligned} \hat{\alpha}_i > 0, & \text{ if } y_i(\hat{\mathbf{w}}^\top \mathbf{x}_i + \hat{b}) = 1 \\ \hat{\alpha}_i = 0, & \text{ if } y_i(\hat{\mathbf{w}}^\top \mathbf{x}_i + \hat{b}) > 1 \end{aligned} .$$

- The predicted label for any \mathbf{x} is

$$\text{sign}(\hat{\mathbf{w}}^\top \mathbf{x} + \hat{b}).$$

Extension to Non-Separable Data Points

How can we apply the max-margin principle if the data are **not** linearly separable?



Soft-margin SVM

We introduce slack variables $\zeta = (\zeta_1, \dots, \zeta_n)$ and consider

$$\begin{aligned} & \min_{\mathbf{w}, b, \zeta} \|\mathbf{w}\|_2^2 \\ \text{s.t. } & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0, \quad \text{for all } i = 1, \dots, n \\ & \sum_{i=1}^n \zeta_i \leq K. \end{aligned}$$

- Misclassification occurs if $\zeta_i > 1$.
- $\sum_{i=1}^n \zeta_i \leq K \Rightarrow$ the total number of misclassified points less than K .
- $K \geq 0$ is a tuning parameter.
 $K = 0$ reduces to the hard-margin SVM.

Another interpretation of the soft-margin SVM

- Soft-margin SVM is equivalent to, for some $C = C(K)$,

$$\begin{aligned} \min_{\mathbf{w}, b, \zeta} \quad & \|\mathbf{w}\|_2^2 + C \sum_{i=1}^n \zeta_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0, \quad i = 1, \dots, n. \end{aligned}$$

- This is further equivalent to

$$\min_{\mathbf{w}, b} \frac{1}{n} \sum_{i=1}^n \underbrace{\max \left\{ 0, 1 - y_i (\mathbf{w}^\top \mathbf{x}_i + b) \right\}}_{\text{hinge loss}} + \lambda \|\mathbf{w}\|_2^2$$

with $\lambda = 1/(nC)$. Hence, the soft-margin SVM can be seen as a linear classifier with the **hinge loss** and the ridge penalty.

Hinge Loss

The **hinge loss**:

$$L_{\text{hinge}}(\mathbf{w}, b) = \max \left\{ 0, 1 - y_i (\mathbf{w}^\top \mathbf{x}_i + b) \right\}$$

We only want to minimize $1 - y_i (\mathbf{w}^\top \mathbf{x}_i + b)$ when it is positive.

| | | |
|-----------------------------------------------------|---------------|---------------------|
| $y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$ | \Rightarrow | ✓ + out of margin |
| $y_i (\mathbf{w}^\top \mathbf{x}_i + b) \in [0, 1]$ | \Rightarrow | ✓ but within margin |
| $y_i (\mathbf{w}^\top \mathbf{x}_i + b) < 0$ | \Rightarrow | ✗ |

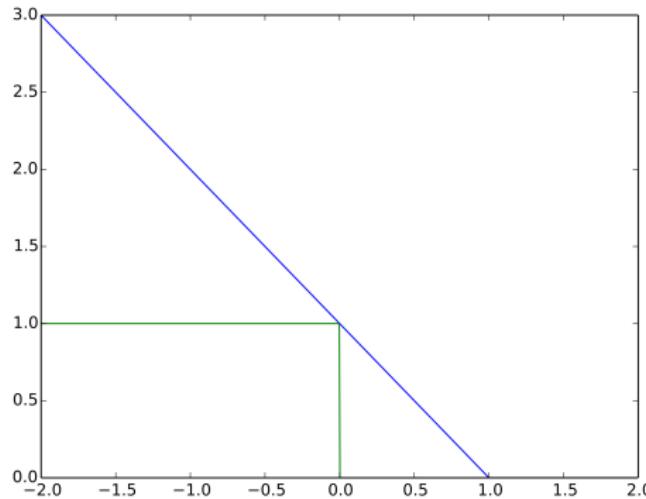
The 0-1 loss

$$L_{0-1}(\mathbf{w}, b) = 1 \left\{ y_i (\mathbf{w}^\top \mathbf{x}_i + b) < 0 \right\}.$$

Revisiting Loss Functions for Classification

Hinge loss compared with the 0-1 loss:

$$y = \max\{0, 1 - x\} \quad \text{v.s.} \quad y = 1\{x < 0\}.$$



Prime-formulation of the soft-margin SVM

Soft-margin SVM is equivalent to, for some $C = C(K)$,

$$\begin{aligned} \min_{\mathbf{w}, b, \zeta} \quad & \|\mathbf{w}\|_2^2 + C \sum_{i=1}^n \zeta_i \\ \text{s.t. } & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0, \quad i = 1, \dots, n. \end{aligned}$$

Dual-formulation of the soft-margin SVM

It can be shown² that the dual-formulation of the soft-margin SVM is

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, n. \end{aligned}$$

Here $C > 0$ is the tuning parameter.

²Chapter 12.2.1 in ESL.

Kernel SVM: extension to non-linear boundary

Recall

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j \\ \text{s.t.} \quad & \sum_{i=1}^n \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, n. \end{aligned}$$

Represent \mathbf{x}_i in different bases, $h(\mathbf{x}_i)$, to have non-linear boundary (in \mathbf{x}_i).

The only change is the objective function

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{4} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{h}(\mathbf{x}_i)^\top \mathbf{h}(\mathbf{x}_j).$$

Kernel trick

- We can represent the inner-product $h(\mathbf{x}_i)^\top h(\mathbf{x}_j)$ by using

$$K(\mathbf{x}_i, \mathbf{x}_j) = h(\mathbf{x}_i)^\top h(\mathbf{x}_j), \quad \forall i \neq j \in \{1, \dots, n\}.$$

The function K is called **kernel** that quantifies the similarity of two feature vectors.

- Regardless how large the space of $h(\mathbf{x}_i)$ is, all we need to compute is the pairwise kernel

$$K(\mathbf{x}_i, \mathbf{x}_j), \quad \forall i \neq j \in \{1, \dots, n\}.$$

This is known as the **kernel trick**.

Examples of kernel SVM

- Linear:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$$

with the corresponding $h(\mathbf{x}_i) = \mathbf{x}_i$.

- d th-Degree polynomial:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^\top \mathbf{x}_j)^d.$$

The corresponding h would be polynomials. For example, consider $d = 2$, $\mathbf{x}_i = x_i$ and $h(\mathbf{x}_i) = [1, \sqrt{2}x_i, x_i^2]$, then

$$K(\mathbf{x}_i, \mathbf{x}_j) = h(\mathbf{x}_i)^\top h(\mathbf{x}_j) = 1 + 2x_i x_j + x_i^2 x_j^2 = (1 + \mathbf{x}_i^\top \mathbf{x}_j)^2.$$

- Radial basis: for some $\gamma > 0$,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|_2^2).$$

The corresponding $h(\mathbf{x}_i)$ has **infinite** dimensions!

Limitations of SVM

- The classifier based on SVM is

$$\text{sign}(\hat{\mathbf{w}}^T \mathbf{x} + \hat{b}).$$

Hence, SVM does not estimate the posterior probability.

- For multi-class classification problems with $C = \{1, 2, \dots, K\}$,
 - ▶ It is non-trivial to generalize the notion of a margin to multiclass setting.
 - ▶ Many different proposals for multi-class SVMs. We discuss two commonly used ad-hoc approaches.

SVM: One-Versus-One

- Construct $\binom{K}{2}$ SVMs for each pair of classes.
 - ▶ For classes {1,2}, consider data (\mathbf{x}_i, y_i) with $y_i \in \{1, 2\}$. Let

$$z_i = -1\{y_i = 1\} + 1\{y_i = 2\}.$$

Fit SVM by using (\mathbf{x}_i, z_i) with $y_i \in \{1, 2\}$.

- ▶ For classes {1,3}, consider data (\mathbf{x}_i, y_i) with $y_i \in \{1, 3\}$. Let

$$z_i = -1\{y_i = 1\} + 1\{y_i = 3\}.$$

Fit SVM by using (\mathbf{x}_i, z_i) with $y_i \in \{1, 3\}$.

- ▶ Repeat for all pairs.

- For each test point \mathbf{x}_0 , assign it to the majority class predicted by $\binom{K}{2}$ SVMs.

SVM:One-Versus-All

- Construct K SVMs by choosing each class one at a time.
 - ▶ For class $\{1\}$, consider ALL data (\mathbf{x}_i, y_i) , $i = 1, \dots, n$. Let

$$z_i = 2 \cdot 1\{y_i = 1\} - 1.$$

Fit SVM and let its parameter be $(\hat{b}^{(1)}, \hat{\mathbf{w}}^{(1)})$.

- ▶ For class $\{2\}$, consider ALL data (\mathbf{x}_i, y_i) , $i = 1, \dots, n$. Let

$$z_i = 2 \cdot 1\{y_i = 2\} - 1.$$

Fit SVM and let its parameter be $(\hat{b}^{(2)}, \hat{\mathbf{w}}^{(2)})$.

- ▶ Repeat for all classes.

- For each test point \mathbf{x}_0 , assign it to the class

$$\arg \max_{k \in C} \left(\hat{b}^{(k)} + \mathbf{x}_0^T \hat{\mathbf{w}}^{(k)} \right).$$

LDA vs SVM vs LR

1. Since LDA requires additional Gaussianity, SVM is more similar as LR than LDA.
When Gaussianity can be justified, LDA has the best performance.
2. SVM is less used for multi-class classification problems.
3. SVM does not estimate the conditional probabilities, such as $\mathbb{P}(Y = 1 | X)$, but LDA and LR do.
4. When classes are separable, SVM and LDA perform better than LR.
When classes are non-separable, LR (with ridge penalty) and SVM are very similar.