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 that use **linear decision** boundaries.

- Logistic regression:

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

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

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

  The decision boundary is

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

Linear decision boundaries

- LDA:

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

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

$$\left(\mathbf{x} - \frac{u_0 + u_1}{2}\right)^\top \Sigma^{-1}(u_1 - u_0) + \log \frac{\pi_1}{\pi_0} \geq 0.$$

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 $\boldsymbol{\alpha}$.

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

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

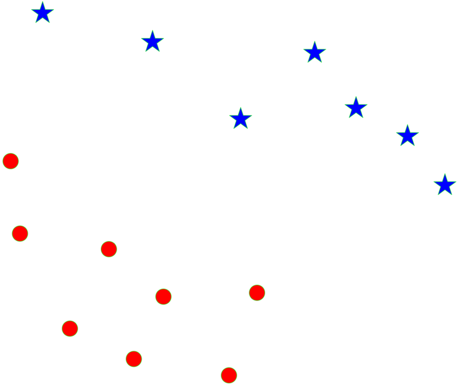
for some weights $\mathbf{w} \in \mathbb{R}^p$ and $b \in \mathbb{R}$.

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

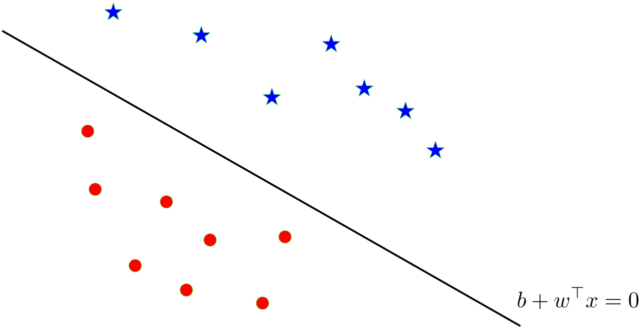
$$\begin{aligned} \mathbf{w}^\top \mathbf{x} + b > 0, &\quad \text{if } y = 1 \\ \mathbf{w}^\top \mathbf{x} + b < 0, &\quad \text{if } y = -1. \end{aligned}$$

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 $(p-1)$ dimensional space , a.k.a. hyperplane.

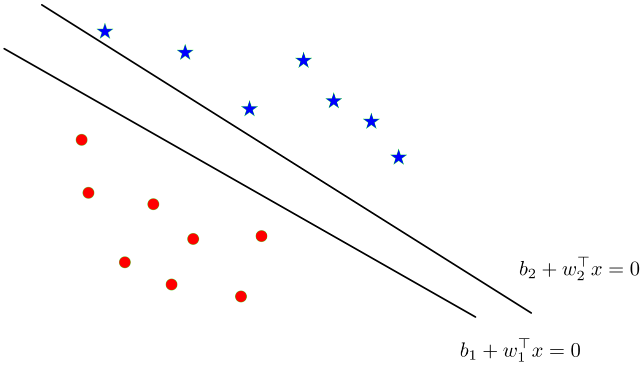
Simple Intuition and Potential Issues

To correctly classify all points we require that

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

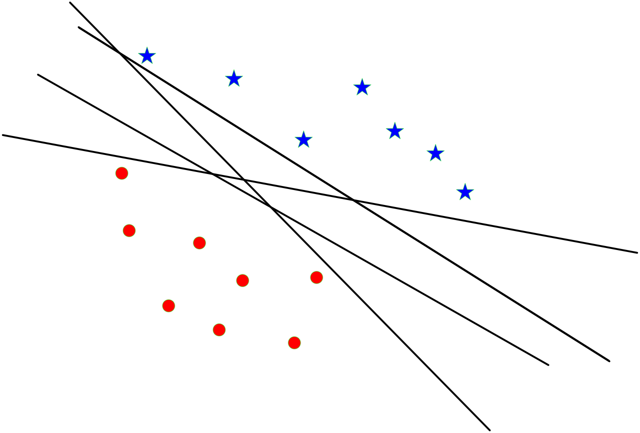
- We should find $\mathbf{w}$ and b to meet the above goal.
- However:
  - When the data is separable, there exists multiple solutions of $\mathbf{w}$ and b. Which to choose?
  - When the data is not separable, it is infeasible.

Separable Cases



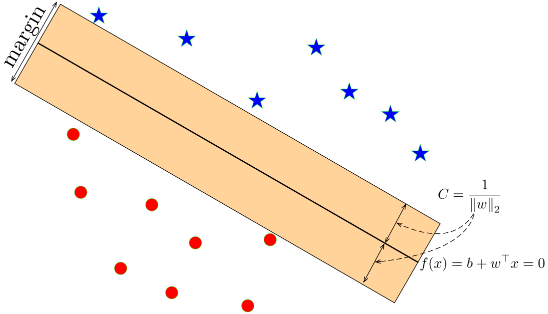
- There are multiple separating hyperplanes, determined by different parameters $(\mathbf{w}, b)$.

Separable Cases



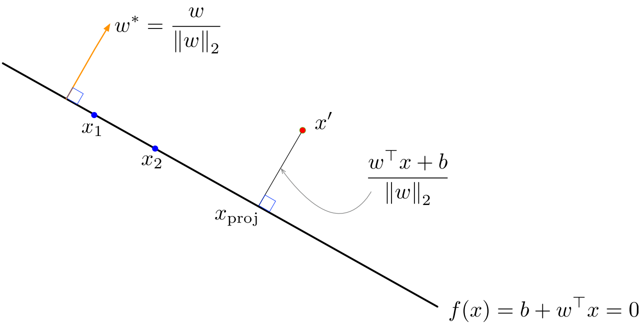
Optimal Separating Hyperplane

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.



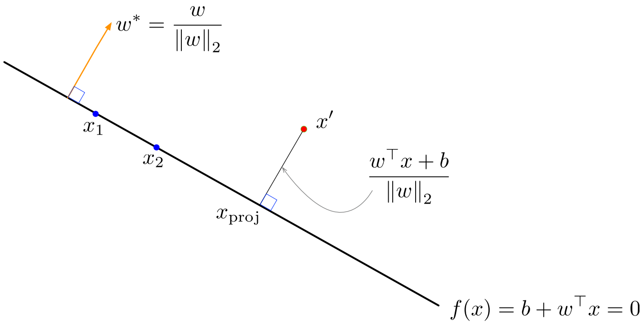
Intuitively, ensuring that a classifier is not too close to any data points leads to better generalization on the test data.

Geometry of Points and Planes



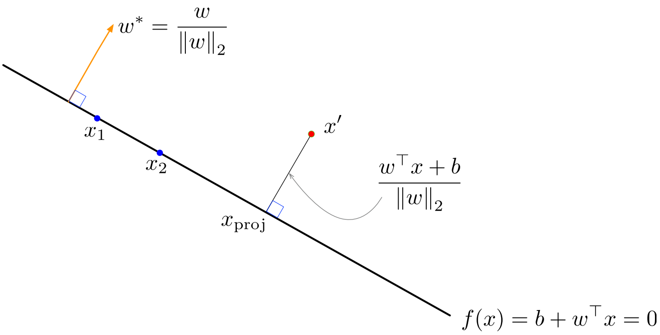
- Recall that the decision hyperplane is orthogonal (perpendicular) to $\mathbf{w}$. I.e., for any two points $\mathbf{x}_1$ and $\mathbf{x}_2$ on the decision hyperplane we have that $\mathbf{w}^\top(\mathbf{x}_1 - \mathbf{x}_2) = 0$.

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 $\mathbf{x}'$ to the hyperplane $\{\mathbf{x} : b + \mathbf{w}^\top \mathbf{x} = 0\}$.

Distance to a Given Hyperplane

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

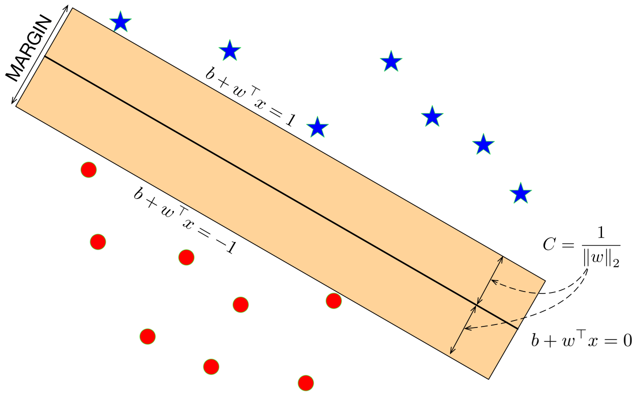
- Take the closest point $\mathbf{x}_{\text{proj}}$ on the hyperplane, which satisfies

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

- 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



- Now consider the two parallel hyperplanes

$$\mathbf{w}^\top \mathbf{x} + b = 1 \qquad \mathbf{w}^\top \mathbf{x} + b = -1$$

- Using the distance formula, can see that **the margin** is $2/\|\mathbf{w}\|_2$.

Maximizing Margin as an Optimization Problem

- Recall: to correctly classify all points we require that

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

- Let's impose a stronger requirement: correctly classify all points **and** prevent them from falling in the margin. For some $M > 0$,

$$\begin{aligned} \mathbf{w}^\top \mathbf{x}_i + b &\geq M & \text{if } y_i = 1 \\ \mathbf{w}^\top \mathbf{x}_i + b &\leq -M & \text{if } y_i = -1 \end{aligned}$$

- This is equivalent to

$$y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq M \qquad \text{for all } i \in [n]$$

which we call the **margin constraints**.

Maximizing Margin as an Optimization Problem

- There might exist multiple $(\mathbf{w}, b)$ satisfy the margin constraints. We want to pick the one that maximizes the width of the margin,

$$\frac{\left|\mathbf{x}^\top \mathbf{w} + b\right|}{||\mathbf{w}||_2} = \frac{M}{||\mathbf{w}||_2}.$$

- This leads to the max-margin objective:

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

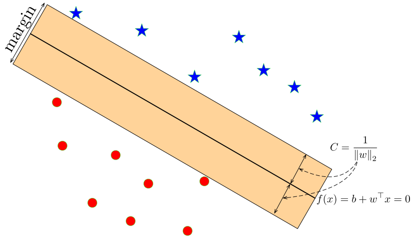
W.l.o.g. we can set $M = 1$. (Why?)

Maximizing Margin as an Optimization Problem

Max-margin objective:

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

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



- 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**.

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

Computation of the hard-margin SVM

Primal-formulation:

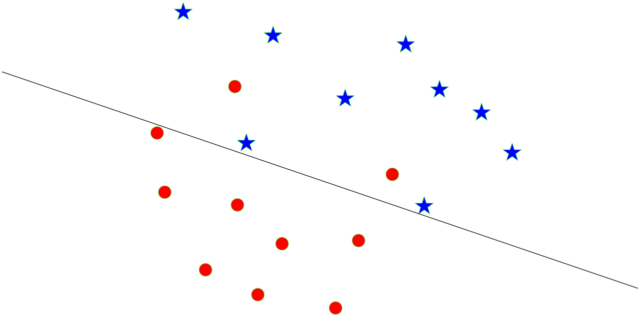
$$\min_{\mathbf{w},b} ||\mathbf{w}||_2^2$$
$$\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad i = 1, \ldots, n$$

- Convex, in fact, a quadratic program. (Stochastic) Gradient descent can be directly used.
- In practice, it is more common to solve the optimization problem based on its dual formulation.[2]

[2] See the suggested reading.

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 $\zeta = (\zeta_1, \ldots, \zeta_n)$ and consider

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

- Misclassification occurs if $\zeta_i > 1$.
- $\sum_{i=1}^n \zeta_i \leq K$ restricts 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 $C = C(K)$,

$$\min_{\mathbf{w},b,\zeta} \ \|\mathbf{w}\|_2^2 + C\sum_{i=1}^{n}\zeta_i$$

$$\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0, \quad i = 1, \ldots, n.$$

- This is further equivalent to

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

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

Limitations of SVM

- The classifier based on SVM is

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

  Hence, SVM does not estimate the posterior probability.

- For multi-class classification problems,
  - 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 in the suggested reading material.

LDA vs SVM vs Logistic Regression (LR)

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