Support vector machines

Recap: linear classifiers (with $\mathcal{Y} = \{-1, +1\}$)

Setting: linearly separable data

Assume there is a linear classifier that perfectly classifies the training data S: for some $\boldsymbol{w}_\star \in \mathbb{R}^d$ and $t_\star \in \mathbb{R}$,

$$y\big(\langle \boldsymbol{w}_\star, \boldsymbol{x} \rangle - t_\star\big) > 0 \quad \text{for all } (\boldsymbol{x}, y) \in S\,.$$

Linear programming

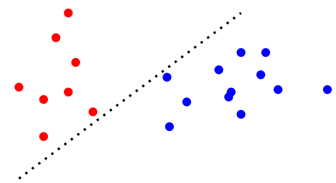
Solve linear feasibility problem: find $\boldsymbol{w} \in \mathbb{R}^d$ and $t \in \mathbb{R}$ such that

$$y\big(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - t\big) > 0 \quad \text{for all } (\boldsymbol{x}, y) \in S\,.$$

Can find *some* linear separator in polynomial time.

Perceptron algorithm

Finds *some* linear separator quickly if there is a large margin.



Support vector machines (SVMs)

Motivation

- Ambiguity and potential instability in what LP and Perceptron returns.
- What to do when S is not linearly separable?
  (Some possibilities are logistic regression and Online Perceptron.)

Support vector machines (Vapnik and Chervonenkis, 1963)

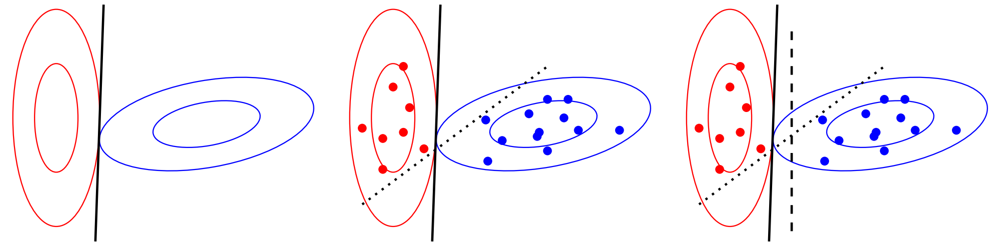
- Characterize a *stable* solution for linearly separable problems—the **maximum margin solution**.
- SVM specified as solution to a **convex optimization problem** that can be solved in polynomial time.
- Kernelizable via **convex duality**. (SVM gets its name from its dual form.)
- Slight alteration to optimization problem gives natural way to handle non-separable cases via **convex surrogate losses**.

Three main points about SVMs

1. The maximum margin solution can be characterized as the solution to a optimization problem.
2. The dual of this optimization problem reveals properties of the solution; leads to Kernel SVMs.
3. The optimization problem can be easily modified to handle the case where data are not linearly separable.

Maximum margin solution

Maximum margin solution



Best linear classifier on population

Possible Perceptron or LP solution on training data S

"Maximum margin" solution on training data S

Why use the "maximum margin" solution?
(i) Uniquely determined by S (except in degenerate cases), unlike LP's/Perceptron's.

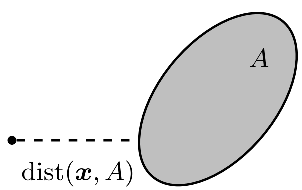
(ii) It is a particular "learning bias"—i.e., an assumption about the problem—that seems to be commonly useful.

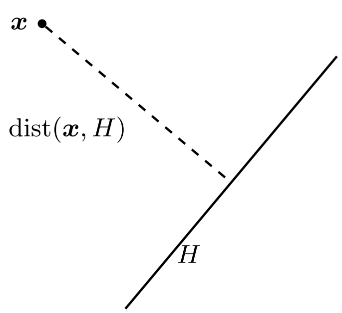
Our goal: Precisely characterize the maximum margin solution as the solution to a **mathematical optimization problem**.
(For now, don't worry about how to *solve* the optimization problem.)

Distance to a set

The **distance between a point x and a set A** is the Euclidean distance between $\boldsymbol{x}$ and the closest point in A:

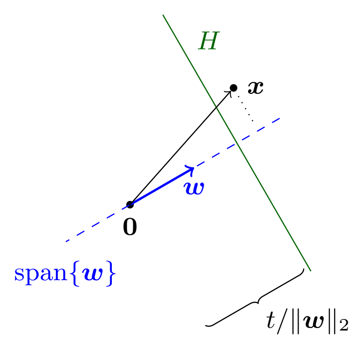
$$\operatorname{dist}(\boldsymbol{x}, A) \;:=\; \min_{\boldsymbol{z} \in A} \|\boldsymbol{x} - \boldsymbol{z}\|_2 \,.$$





Distance to the decision boundary

Consider linear classifier $f_{\boldsymbol{w},t}$ (where $\boldsymbol{w} \in \mathbb{R}^d \setminus \{\boldsymbol{0}\}$ and $t \in \mathbb{R}$).



- Correct classification on $(\boldsymbol{x}, y)$:

$$f_{\boldsymbol{w},t}(\boldsymbol{x}) \;=\; y \quad \text{iff} \quad y\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - t\big) \;>\; 0\,.$$

- Proj. of $\boldsymbol{x}$ onto $\operatorname{span}\{\boldsymbol{w}\}$: $\dfrac{\langle \boldsymbol{w}, \boldsymbol{x}\rangle}{\|\boldsymbol{w}\|_2} \cdot \dfrac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$.
- Distance to affine hyperplane H is

$$\operatorname{dist}(\boldsymbol{x}, H) \;=\; \frac{|\langle \boldsymbol{w}, \boldsymbol{x}\rangle - t|}{\|\boldsymbol{w}\|_2}\,.$$

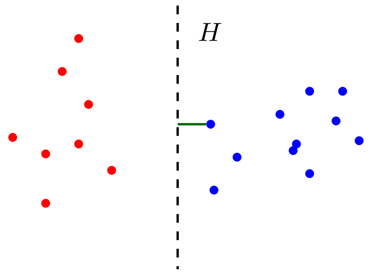
- If $f_{\boldsymbol{w},t}(\boldsymbol{x}) = y$, then

$$\operatorname{dist}(\boldsymbol{x}, H) \;=\; \frac{y\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - t\big)}{\|\boldsymbol{w}\|_2}\,.$$

Margin of a linear separator

If $f_{\boldsymbol{w},t}(\boldsymbol{x}) = y$ for all $(\boldsymbol{x}, y) \in S$, then the **(minimum) margin of $f_{\boldsymbol{w},t}$ on S** (i.e., smallest distance to decision boundary H) is

$$\min_{(\boldsymbol{x},y)\in S} \operatorname{dist}(\boldsymbol{x}, H) \;=\; \frac{\min_{(\boldsymbol{x},y)\in S} y\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - t\big)}{\|\boldsymbol{w}\|_2}\,.$$



To find $f_{\boldsymbol{w},t}$ that *maximizes* the **margin**:

- Require numerator to be ≥ 1 via *linear constraints*:

  $$y\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - t\big) \;\geq\; 1 \quad \text{for all } (\boldsymbol{x}, y) \in S\,.$$

- Then *minimize* the denominator $\|\boldsymbol{w}\|_2$ subject to these constraints.

Maximum margin linear separator

The solution $(\hat{\boldsymbol{w}}, \hat{t})$ to the following mathematical optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w}\in\mathbb{R}^d, t\in\mathbb{R}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y\big(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - t\big) \;\geq\; 1 \quad \text{for all } (\boldsymbol{x}, y) \in S \end{aligned}$$

gives the **linear classifier with the maximum margin on S**.

The linear classifier obtained by solving this optimization problem is called a **support vector machine** (SVM).

The optimization problem is a **convex optimization problem** that can be solved in polynomial time. (Actual algorithm to come later.)

If there is a solution (i.e., the problem is separable), then the solution is *unique*. (Compare to LP's and Perceptron's lack of determinism from S.)

Convex duality

SVM problem

$$\begin{aligned} \min_{\boldsymbol{w}\in\mathbb{R}^d, t\in\mathbb{R}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle - t\big) \;\geq\; 1 \quad \text{for all } i = 1, 2, \dots, n\,. \end{aligned}$$

Every convex optimization problem has corresponding **dual problem** with *same optimum value*.

SVM dual problem

$$\begin{aligned} \max_{\alpha_1,\alpha_2,\dots,\alpha_n \geq 0} \quad & \sum_{i=1}^n \alpha_i - \sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i \;=\; 0\,. \end{aligned}$$

Fact: optimal solutions $(\hat{\boldsymbol{w}}, \hat{t})$ and $\hat{\boldsymbol{\alpha}}$ satisfy

$$\hat{\boldsymbol{w}} \;=\; \sum_{i=1}^n \hat{\alpha}_i y_i \boldsymbol{x}_i\,,$$

$$\hat{\alpha}_i > 0 \;\Rightarrow\; y_i\Big(\langle \hat{\boldsymbol{w}}, \boldsymbol{x}_i\rangle - \hat{t}\Big) \;=\; 1 \quad \text{for all } i = 1, 2, \dots, n\,.$$

Kernel SVMs (Boser, Guyon, and Vapnik, 1992)

- SVM solution *entirely determined by* $(\boldsymbol{x}_i, y_i)$ *where* $\hat{\alpha}_i > 0$.
  These data points are called the *support vectors*:

  $$\hat{\alpha}_i > 0 \;\Rightarrow\; y_i\Big(\langle \hat{\boldsymbol{w}}, \boldsymbol{x}_i\rangle - \hat{t}\Big) \;=\; 1 \quad \text{for all } i = 1, 2, \dots, n\,.$$

  - Support vectors satisfy "margin" constraints with equality.
  - Can throw away all data except the support vectors, re-solve SVM problem, and get the same solution.

- Dual problem only depends on $\boldsymbol{x}_i$ through inner products: $\langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle$.
  Can replace with $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for any kernel K.

$$\begin{aligned} \max_{\alpha_1,\alpha_2,\dots,\alpha_n \geq 0} \quad & \sum_{i=1}^n \alpha_i - \sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \text{s.t.} \quad & \sum_{i=1}^m \alpha_i y_i \;=\; 0\,. \end{aligned}$$

Non-separable case

Soft-margin SVMs (Cortes and Vapnik, 1995)

When $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ is not linearly separable, the (primal) SVM optimization problem

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^d, t \in \mathbb{R}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - t\big) \ \geq \ 1 \qquad \text{for all } i = 1, 2, \ldots, n \end{aligned}$$

has no solution.

Introduce **slack variables** $\xi_1, \xi_2, \ldots, \xi_n \geq 0$, and a trade-off parameter $C > 0$:

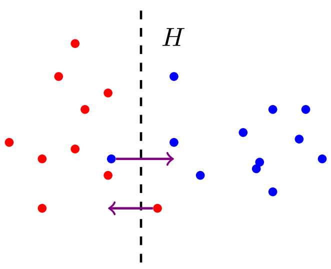
$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^d, t \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}^n} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - t\big) \ \geq \ 1 - \xi_i \qquad \text{for all } i = 1, 2, \ldots, n\,, \\ & \xi_i \ \geq \ 0 \qquad \text{for all } i = 1, 2, \ldots, n\,, \end{aligned}$$

which is **always feasible**. This is called **soft margin SVM**.

(Slack variables are *auxiliary variables*; not needed to form the linear classifier.)

Interpretation of slack varables

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^d, t \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}^n} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - t\big) \ \geq \ 1 - \xi_i \qquad \text{for all } i = 1, 2, \ldots, n\,, \\ & \xi_i \ \geq \ 0 \qquad \text{for all } i = 1, 2, \ldots, n\,. \end{aligned}$$



For given $(\boldsymbol{w}, t)$, $\xi_i / \|\boldsymbol{w}\|_2$ is distance that $\boldsymbol{x}_i$ would have to move to satisfy

$$y_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - t) \ \geq \ 1\,.$$

Another interpretation of slack variables

Constraints with non-negative slack variables: (using $\lambda := 1/(nC)$)

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^d, t \in \mathbb{R}, \boldsymbol{\xi} \in \mathbb{R}^n} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{n} \sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i\big(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle - t\big) \ \geq \ 1 - \xi_i \qquad \text{for all } i = 1, 2, \ldots, n\,, \\ & \xi_i \ \geq \ 0 \qquad \text{for all } i = 1, 2, \ldots, n\,. \end{aligned}$$

Equivalent unconstrained form:

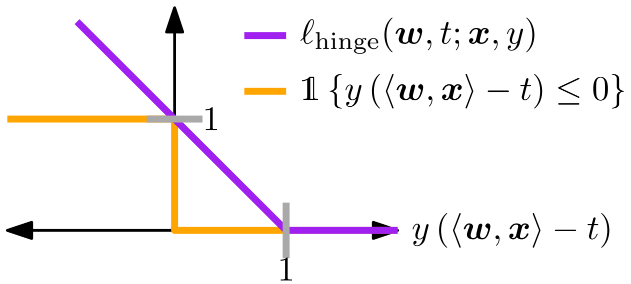
$$\min_{\boldsymbol{w} \in \mathbb{R}^d, t \in \mathbb{R}} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{n} \sum_{i=1}^n \ell_{\text{hinge}}(\boldsymbol{w}, t; \boldsymbol{x}_i, y_i)\,.$$

Notation: $[a]_+ := \max\{0, a\}$.

The **hinge loss** of a linear classifier $f_{\boldsymbol{w},t}$ on an example $(\boldsymbol{x}, y)$ is defined to be

$$\ell_{\text{hinge}}(\boldsymbol{w}, t; \boldsymbol{x}, y) \ := \ \Big[1 - y\big(\langle \boldsymbol{w}, \boldsymbol{x} \rangle - t\big)\Big]_+\,.$$

Zero-one loss vs. hinge loss



Hinge loss: an upper-bound on zero-one loss.

$$\mathbb{1}\left\{y\left(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - t\right) \leq 0\right\} \;\leq\; \left[1 - y\left(\langle \boldsymbol{w}, \boldsymbol{x}\rangle - t\right)\right]_+ \;=\; \ell_{\text{hinge}}(\boldsymbol{w}, t; \boldsymbol{x}, y)\,.$$

Soft-margin SVM minimizes an upper-bound on the training error rate, plus a term that encourages large margins.

This is **computationally tractable** (unlike minimizing training error rate) because the hinge loss is a **convex function** of $(\boldsymbol{w}, t)$, and so is $\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$.

Key takeaways

1. Formulation of learning an SVM as a mathematical optimization problem defined by training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$.
2. High-level idea of convex duality; properties of SVM solution via convex duality, and how to “kernelize” SVMs.
3. Role of slack variables and hinge-loss in soft-margin SVMs.