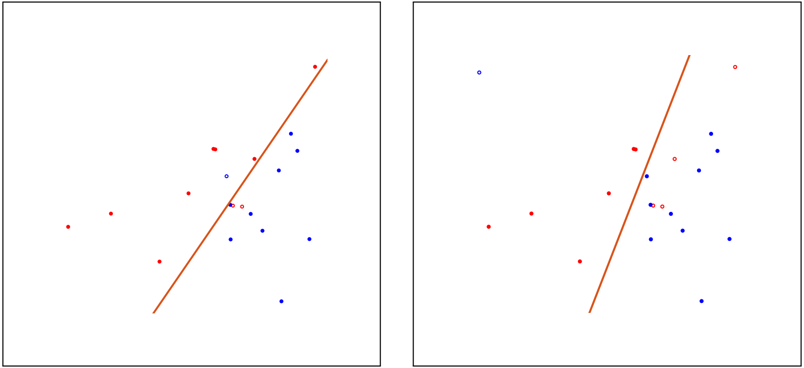
Linear, Binary SVM Classifiers

COMPSCI 371D — Machine Learning

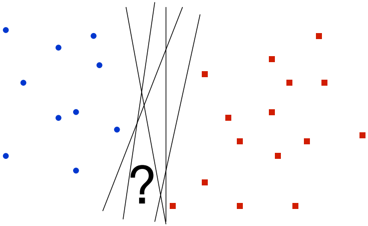
Outline

An Issue with the Logistic Regression Classifier



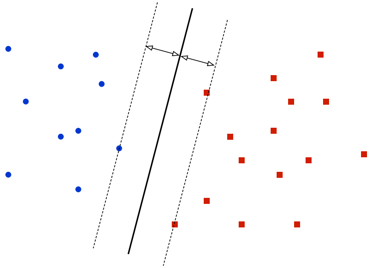
- LRC boundary depends on all the points
- The landscape near the boundary should matter most

The Separable Case



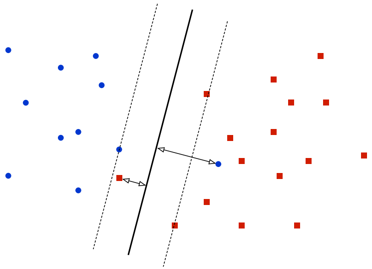
- Where to place the boundary?
- The number of degrees of freedom grows with d

SVMs Maximize the Smallest *Margin*



- Placing the boundary as far as possible from the nearest samples improves generalization
- Leave as much empty space around the boundary as possible
- Only the points that barely make the margin matter
- These are the *support vectors*
- Initially, we don't know which points will be support vectors

The General Case: Soft SVMs



- If the data is not linearly separable, there *must* be misclassified samples. These have a negative margin
- Assign a penalty that penalizes a narrow band around the boundary *and* the number of samples that fall into it or on the incorrect side of the boundary
- Give different weights to the two penalties (cross-validation!)
- Find the optimal compromise: minimum risk (total penalty)

Separating Hyperplane

- $X = \mathbb{R}^d$ and $Y = \{-1, 1\}$
  (more convenient labels than $\{0, 1\}$)
- Hyperplane: $\mathbf{n}^T\mathbf{x} + c = 0$ with $\|\mathbf{n}\| = 1$
- Decision rule: $\hat{y} = h(\mathbf{x}) = \text{sign}(\mathbf{n}^T\mathbf{x} + c)$
- $\mathbf{n}$ points towards the $\hat{y} = 1$ half-space
- If y is the true label, decision is correct if
  $$\begin{cases} \mathbf{n}^T\mathbf{x} + c \geq 0 & \text{if } y = 1 \\ \mathbf{n}^T\mathbf{x} + c \leq 0 & \text{if } y = -1 \end{cases}$$
- More compactly,
  decision is correct if $y(\mathbf{n}^T\mathbf{x} + c) \geq 0$
- SVMs want this inequality to hold with a *margin*

Margin

- The margin of $(\mathbf{x}, y)$ is the signed distance of $\mathbf{x}$ from the boundary: Positive if $\mathbf{x}$ is on the correct side of the boundary, negative otherwise

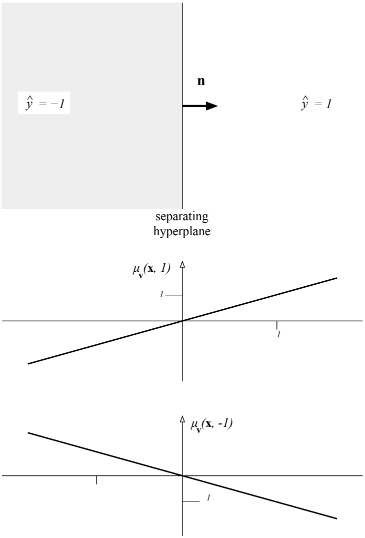
$$\mu_{\mathbf{v}}(\mathbf{x}, y) \stackrel{\text{def}}{=} y\,(\mathbf{n}^T\mathbf{x} + c)$$

- $\mathbf{v} = (\mathbf{n}, c)$
- Margin of a training set T:

$$\mu_{\mathbf{v}}(T) \stackrel{\text{def}}{=} \min_{(\mathbf{x},y)\in T} \mu_{\mathbf{v}}(\mathbf{x}, y)$$

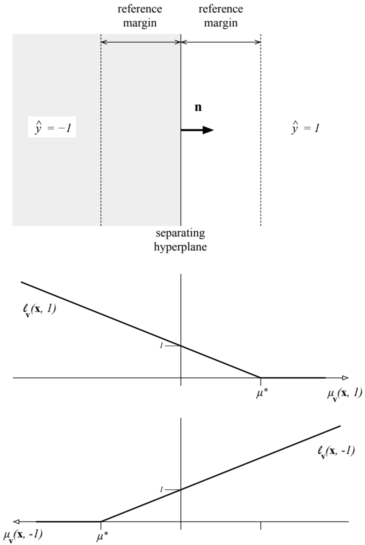
- Boundary separates T if

$$\mu_{\mathbf{v}}(T) > 0$$



The Hinge Loss

- *Reference margin* $\mu^* > 0$ (unknown, to be determined)
- *Hinge loss* $\ell_{\mathbf{v}}(\mathbf{x}, y)$:
  $\frac{1}{\mu^*} \max\{0, \mu^* - \mu_{\mathbf{v}}(\mathbf{x}, y)\}$
- Training samples with $\mu_{\mathbf{v}}(\mathbf{x}, y) \geq \mu^*$ are classified correctly with a margin at least μ^*
- Some loss incurred as soon as $\mu_{\mathbf{v}}(\mathbf{x}, y) < \mu^*$ *even if the sample is classified correctly*



The Training Risk

- The training risk for SVMs is not just $\frac{1}{N}\sum_{n=1}^{N} \ell_{\mathbf{v}}(\mathbf{x}_n, y_n)$
- A *regularization term* is added to force μ^* to be large
- Decision boundary is $\mathbf{n}^T\mathbf{x} + c = 0$

  $\ell_{\mathbf{v}}(\mathbf{x}, y) = \frac{1}{\mu^*} \max\{0, \mu^* - \mu_{\mathbf{v}}(\mathbf{x}, y)\}$

  $= \frac{1}{\mu^*} \max\{0, \mu^* - y\,(\mathbf{n}^T\mathbf{x} + c)\} = \max\{0, 1 - y(\mathbf{w}^T\mathbf{x} + b)\}$

  $= \ell_{(\mathbf{w},b)}(\mathbf{x}, y)$

  where the decision boundary is $\mathbf{w}^T\mathbf{x} + b = 0$

  with $\mathbf{w} = \frac{\mathbf{n}}{\mu^*}, \quad b = \frac{c}{\mu^*}$ and $\quad \|\mathbf{w}\| = \frac{1}{\mu^*}$
- Make risk higher when $\frac{1}{\mu^*}$ is large (small margin):

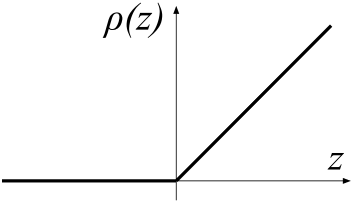
  $L_T(\mathbf{w}, b) \stackrel{\text{def}}{=} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C_0}{N}\sum_{n=1}^{N} \ell_{(\mathbf{w},b)}(\mathbf{x}_n, y_n)$

Regularized Risk

- ERM classifier:
  $(\mathbf{w}^*, b^*) = \text{ERM}_T(\mathbf{w}, b) = \arg\min_{(\mathbf{w},b)} L_T(\mathbf{w}, b)$
  where $L_T(\mathbf{w}, b) \stackrel{\text{def}}{=} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C_0}{N}\sum_{n=1}^{N} \ell_{(\mathbf{w},b)}(\mathbf{x}_n, y_n)$
- $\ell_{(\mathbf{w},b)}(\mathbf{x}_n, y_n) \stackrel{\text{def}}{=} \max\{0, 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b)\}$
- C_0 determines a trade-off
- C_0 is a hyper-parameter: Cross-validation!
- Large $C_0 \Rightarrow \|\mathbf{w}\|$ less important $\Rightarrow$ smaller margin μ^*
  $\Rightarrow$ fewer samples within the margin
- We buy a larger margin at the cost of more samples inside it

Training an SVM

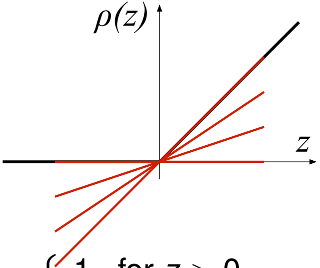
- $(\mathbf{w}^*, b^*) = \arg\min_{(\mathbf{w},b)} L_T(\mathbf{w}, b)$ where
  $L_T(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C_0}{N}\sum_{n=1}^{N} \ell_n$ and
  $\ell_n = \ell_{(\mathbf{w},b)}(\nu_n) \stackrel{\text{def}}{=} \max\{0, 1 - \underbrace{y_n(\mathbf{w}^T\mathbf{x}_n + b)}_{\nu_n}\}$
  $= \max\{0, 1 - \nu_n\} = \rho(\overbrace{1 - \nu_n}^{z})$
- $\rho(z) = \max\{0, z\}$ is the *hinge function*



- A.k.a. Rectified Linear Unit (ReLU) in deep learning

Training an SVM

- $(\mathbf{w}^*, b^*) = \arg\min_{(\mathbf{w},b)} L_T(\mathbf{w}, b)$ where
  $L_T(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C_0}{N}\sum_{n=1}^{N} \rho(1 - y_n(\mathbf{w}^T\mathbf{x}_n + b))$
- Use gradient or stochastic gradient descent on $L_T(\mathbf{w}, b)$
- ρ not differentiable $\rightarrow$ use the sub-gradient



- For instance, $\rho'(z) = \begin{cases} 1 & \text{for } z > 0 \\ 0 & \text{elsewhere.} \end{cases}$

Sub-Gradient of the Risk

- SGD: Mini-batch B of size M with $1 \leq M \leq N$
- $(\mathbf{w}^*, b^*) = \arg\min_{(\mathbf{w},b)} L_B(\mathbf{w}, b) \quad$ where
  $L_B(\mathbf{w}, b) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C_0}{M} \sum_{n=1}^{M} \rho(1 - y_n(\mathbf{w}^T \mathbf{x}_n + b))$

$$
\begin{aligned}
\frac{\partial L_B}{\partial \mathbf{w}} &= \mathbf{w} - \frac{C_0}{M} \sum_{n=1}^{M} \rho'(1 - y_n(\mathbf{w}^T \mathbf{x}_n + b))\, y_n \mathbf{x}_n \\
\frac{\partial L_B}{\partial b} &= -\frac{C_0}{M} \sum_{n=1}^{M} \rho'(1 - y_n(\mathbf{w}^T \mathbf{x}_n + b))\, y_n \,.
\end{aligned}
$$

- Use (stochastic) gradient descent to find $\mathbf{w}^*, b^*$
- Recall that the risk is *convex*