Linear, Binary SVM Classifiers
Outline

1. What Linear, Binary SVM Classifiers Do
2. Margin
3. Loss and Regularized Risk
4. Training an SVM
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

• If the data is not linearly separable, there must be misclassified samples. These have a negative margin.
• Assign a penalty that increases when the smallest margin diminishes (penalize a small margin between classes), and grows with any negative margin (penalize misclassified samples).
• 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)
- 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 \((x, y)\) is the signed distance of \(x\) from the boundary: Positive if \(x\) is on the correct side of the boundary, negative otherwise

\[
\mu_v(x, y) \overset{\text{def}}{=} y (n^T x + c)
\]

• \(v = (n, c)\)

• Margin of a training set \(T\):

\[
\mu_v(T) \overset{\text{def}}{=} \min_{(x,y) \in T} \mu_v(x, y)
\]

• Boundary separates \(T\) if

\[
\mu_v(T) > 0
\]
The Hinge Loss

- **Reference margin** $\mu^* > 0$ (unknown, to be determined)
- **Hinge loss** $\ell_v(x, y)$:
  \[
  \frac{1}{\mu^*} \max\{0, \mu^* - \mu_v(x, y)\}
  \]
- Training samples with $\mu_v(x, y) \geq \mu^*$ are classified correctly with a margin at least $\mu^*$
- Some loss incurred as soon as $\mu_v(x, y) < \mu^*$ even if the sample is classified correctly
Loss and Regularized Risk

The Training Risk

- The training risk for SVMs is not just \( \frac{1}{N} \sum_{n=1}^{N} \ell_v(x_n, y_n) \)
- A \textit{regularization term} is added to force \( \mu^* \) to be large
- Separating hyperplane is \( n^T x + c = 0 \)
- Let \( w^T x + b = 0 \) with \( w = \omega n, \ b = \omega c \)
  and \( \omega = \|w\| = \frac{1}{\mu^*} \)
- \( \omega \) is a reciprocal scaling factor if \( w \) is changed for a fixed \( b \):
  Large margin, small \( \omega \)
- Make risk higher when \( \omega \) is large (small margin):
  \[
  L_T(w, b) \overset{\text{def}}{=} \frac{1}{2} \|w\|^2 + \frac{C_0}{N} \sum_{n=1}^{N} \ell(w, b)(x_n, y_n)
  \]
  where \( \ell(w, b)(x, y) = \frac{1}{\mu^*} \max\{0, \mu^* - \mu(w, b)(x, y)\} \)
  \[
  = \frac{1}{\mu^*} \max\{0, \mu^* - y(n^T x + c)\} = \max\{0, 1 - y(w^T x + b)\}
  \]
Regularized Risk

- ERM classifier:
\[
(w^*, b^*) = \text{ERM}_T(w, b) = \arg \min_{(w, b)} L_T(w, b)
\]
where
\[
L_T(w, b) \overset{\text{def}}{=} \frac{1}{2} \|w\|^2 + \frac{C_0}{N} \sum_{n=1}^{N} \ell_{(w, b)}(x_n, y_n)
\]

- \(\ell_{(w, b)}(x_n, y_n) \overset{\text{def}}{=} \max\{0, 1 - y_n(w^T x_n + b)\}\)
- \(C_0\) determines a trade-off
- Large \(C_0 \Rightarrow \|w\|\) less important \(\Rightarrow\) larger \(\omega\) \(\Rightarrow\) smaller margin
  \(\Rightarrow\) fewer samples within the margin
- We buy a larger margin by accepting more samples inside it
- \(C_0\) is a hyper-parameter: Cross-validation!
Training an SVM

- \((w^*, b^*) = \arg \min_{(w,b)} L_T(w, b)\) where
  \(L_T(w, b) = \frac{1}{2} \|w\|^2 + \frac{C_0}{N} \sum_{n=1}^{N} \ell_n\) and
  \(\ell_n = \ell_{(w,b)}(\nu_n) \overset{\text{def}}{=} \max\{0, 1 - y_n(w^T x_n + b)\}\)

- \(= \max\{0, 1 - \nu_n\} = \rho(1 - \nu_n)\)

- \(\rho(z) = \max\{0, z\}\) is the hinge function

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

- \((w^*, b^*) = \arg \min_{(w,b)} L_T(w, b)\) where

\[
L_T(w, b) = \frac{1}{2} \|w\|^2 + \frac{C_0}{N} \sum_{n=1}^{N} \rho(1 - y_n(w^T x_n + b))
\]

- Use gradient or stochastic gradient descent on \(L_T(w, b)\)
- \(\rho\) not differentiable → use the sub-gradient

\[
\rho(z) = \begin{cases} 
1 & \text{for } z > 0 \\
0 & \text{elsewhere.}
\end{cases}
\]
Sub-Gradient of the Risk

- \((w^*, b^*) = \text{arg min}_{(w, b)} L_T(w, b)\) where
  
  \[L_T(w, b) = \frac{1}{2} ||w||^2 + \frac{C_0}{N} \sum_{n=1}^{N} \rho(1 - y_n(w^T x_n + b))\]

\[
\frac{\partial L_T}{\partial w} = w - \frac{C_0}{M} \sum_{n=1}^{M} \rho'(1 - y_n(w^T x_n + b)) y_n x_n
\]

\[
\frac{\partial L_T}{\partial b} = -\frac{C_0}{M} \sum_{n=1}^{M} \rho'(1 - y_n(w^T x_n + b)) y_n.
\]

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