Linear, Binary SVM Classifiers
Outline

1. What Linear, Binary SVM Classifiers Do
2. Margin
3. Loss and Regularized Risk
4. Training an SVM
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 \((x, y)\) 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** $\ell_v(x, y)$:
  \[
  \frac{1}{\mu^*} \max\{0, \mu^* - v(x, y)\}
  \]
- Training samples with $v(x, y) \geq \mu^*$ are classified correctly with a margin at least $\mu^*$
- Some loss incurred as soon as $v(x, y) < \mu^*$ even if the sample is classified correctly
The Training Risk

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

\[
\ell_v(x, y) = \frac{1}{\mu^*} \max\{0, \mu^* - \mu_v(x, y)\} \\
= \frac{1}{\mu^*} \max\{0, \mu^* - y(n^T x + c)\} = \max\{0, 1 - y(w^T x + b)\} \\
= \ell_{(w,b)}(x, y)
\]

where the decision boundary is \( w^T x + b = 0 \)
with \( w = \frac{n}{\mu^*}, \ b = \frac{c}{\mu^*} \) and \( \|w\| = \frac{1}{\mu^*} \)

- Make risk higher when \( \frac{1}{\mu^*} \) is large (small margin):

\[
L_T(w, b) \overset{\text{def}}{=} \frac{1}{2} \|w\|^2 + \frac{C_0}{N} \sum_{n=1}^{N} \ell_{(w,b)}(x_n, y_n)
\]
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
- \(C_0\) is a hyper-parameter: Cross-validation!
- Large \(C_0\) \(\Rightarrow\) \(\|w\|\) less important \(\Rightarrow\) smaller margin \(\mu^*\) \(\Rightarrow\) fewer samples within the margin
- We buy a larger margin at the cost of more samples inside it
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

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

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

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

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

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