Linear, Binary SVM Classifiers

COMPSCI 371D — Machine Learning
Outline

1. What Linear, Binary SVM Classifiers Do
2. Margin
3. Loss and Regularized Risk
4. Training an SVM is a Quadratic Program
5. The KKT Conditions and the Support Vectors
6. The Dual Problem
The Separable Case

- Where to place the boundary?
- The number of choices 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
What Linear, Binary SVM Classifiers Do

The General Case

- If the data is not linearly separable, there must be misclassified samples. These have a negative margin.
- Assign a penalty inversely proportional to the margin, and a penalty that grows affinely with the negative of the margin.
- 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** $\xi(x, y)$:
  \[
  \frac{1}{\mu^*} \max\{0, \mu^* - \mu_v(x, y)\}
  \]
- 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
**The Training Risk**

- The training risk for SVMs is not just $\frac{1}{N} \sum_{n=0}^{N} \xi(x_n, y_n)$.
- A *regularization term* is added to force $\mu^*$ to be small.
- 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 only $w$ is changed: Large margin, small $\omega$.
- Make risk higher when $\omega$ is large (small margin):

$$L_T(w, b) \overset{\text{def}}{=} \frac{1}{2} ||w||^2 + \frac{C}{N} \sum_{n=0}^{N} \xi(x_n, y_n, w, b)$$

where $\xi(x, y) = \frac{1}{\mu^*} \max\{0, \mu^* - \mu v(x, y)\}$

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

- ERM classifier:
  \[(w^*, b^*) = \text{ERM}_T(w, b) = \arg \min_{(w, b)} L_T(w, b)\]
  where
  \[L_T(w, b) \overset{\text{def}}{=} \frac{1}{2} \|w\|^2 + \frac{C}{N} \sum_{n=0}^{N} \xi(x_n, y_n, w, b)\]

- \(\xi(x_n, y_n, w, b) \overset{\text{def}}{=} \max\{0, 1 - y_n(w^T x_n + b)\}\)

- \(C\) determines a trade-off
- Large \(C\) \(\Rightarrow\) \(\|w\|\) 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
- Cross-validation!
Rephrasing as Training a Quadratic Program

\[(w^*, b^*) = \arg\min_{(w, b)} \frac{1}{2} \|w\|^2 + \frac{C}{N} \sum_{n=0}^{N} \xi(x_n, y_n, w, b)\]

where \(\xi(x_n, y_n, w, b) \equiv \max\{0, 1 - y_n(w^T x_n + b)\}\)

- Not differentiable because of the \(\max\): Bummer!
- Neat trick: Instead of minimizing \(L_T\) over \(w, b\), we introduce new variables \(\xi_n\) and minimize \(L_T\) over \(w, b, \xi_1, \ldots, \xi_N\) with the constraint \(\xi_n \geq \xi(x_n, y_n, w, b)\)
- We would not decrease \(L_T\) if we made \(\xi_n\) bigger
- Advantage: We can now split \(\xi_n \geq \xi(x_n, y_n, w, b)\) into two separate, affine constraints because \(\xi\) is a hinge:
  - \(\xi_n \geq 0\) everywhere
  - When \(1 - y_n(w^T x_n + b) \geq 0\), we want \(\xi_n \geq 1 - y_n(w^T x_n + b)\)
  - Thus, \(\xi_n \geq 0\) and \(y_n(w^T x_n + b) - 1 + \xi_n \geq 0\)
Training an SVM is a Quadratic Program

Quadratic Program Formulation

- We achieve differentiability at the cost of adding $N$ slack variables $\xi_n$:
- Old: $\min_{(w,b)} \frac{1}{2} \|w\|^2 + \frac{C}{N} \sum_{n=0}^{N} \xi(x_n, y_n, w, b)$
  where $\xi(x_n, y_n, w, b) \overset{\text{def}}{=} \max\{0, 1 - y_n(w^T x_n + b)\}$
- New: $\min_{w,b,\xi} f(w, \xi)$ where $f(w, \xi) = \frac{1}{2} \|w\|^2 + \gamma \sum_{n=1}^{N} \xi_n$
  subject to the constraints
    
    $y_n(w^T x_n + b) - 1 + \xi_n \geq 0$
    $\xi_n \geq 0$

  and with $\gamma \overset{\text{def}}{=} \frac{C}{N}$

- We have our quadratic program!
The Lagrangian

\[
\min \limits_{\mathbf{w}, b, \xi} f(\mathbf{w}, \xi) \quad \text{where} \quad f(\mathbf{w}, \xi) = \frac{1}{2} \| \mathbf{w} \|^2 + \gamma \sum_{n=1}^{N} \xi_n
\]

subject to the constraints

\[
y_n(\mathbf{w}^T \mathbf{x}_n + b) - 1 + \xi_n \geq 0 \\
\xi_n \geq 0
\]

and with \( \gamma \overset{\text{def}}{=} \frac{C}{N} \)

- Lagrangian \( \mathcal{L}(\mathbf{u}, \alpha) = f(\mathbf{u}) - \alpha^T \mathbf{c} \)

- \( \mathcal{L}(\mathbf{w}, b, \xi, \alpha, \beta) \overset{\text{def}}{=} \frac{1}{2} \| \mathbf{w} \|^2 + \gamma \sum_{n=1}^{N} \xi_n \\
- \sum_{n=1}^{N} \alpha_n [y_n(\mathbf{w}^T \mathbf{x}_n + b) - 1 + \xi_n] - \sum_{n=1}^{N} \beta_n \xi_n
\]

- \( \mathbf{u} \) is \((\mathbf{w}, b, \xi)\) and \( \alpha \) has both \( \alpha_n \) and \( \beta_n \)
The KKT Conditions

\[ \mathcal{L}(\mathbf{w}, b, \xi, \alpha, \beta) \overset{\text{def}}{=} \frac{1}{2} ||\mathbf{w}||^2 + \gamma \sum_{m=1}^{N} \xi_m - \sum_{m=1}^{N} \alpha_m [y_m (\mathbf{w}^T \mathbf{x}_m + b) - 1 + \xi_m] - \sum_{m=1}^{N} \beta_m \xi_m \]

\[ \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n = 0 \]

\[ \frac{\partial \mathcal{L}}{\partial b} = - \sum_{n=1}^{N} \alpha_n y_n = 0 \]

\[ \frac{\partial \mathcal{L}}{\partial \xi_n} = \gamma - \alpha_n - \beta_n = 0 \]

\[ y_n (\mathbf{w}^T \mathbf{x}_n + b) - 1 + \xi_n \geq 0 \]

\[ \xi_n \geq 0 \]

\[ \alpha_n \geq 0 \]

\[ \beta_n \geq 0 \]

\[ \alpha_n [y_n (\mathbf{w}^T \mathbf{x}_n + b) - 1 + \xi_n] = 0 \]

\[ \beta_n \xi_n = 0 \]

\[ \overline{\mathbf{u}} = \begin{bmatrix} \overline{\mathbf{w}} \\ \overline{b} \\ \overline{\xi} \end{bmatrix} \]
The KKT Conditions and the Support Vectors

The Support Vectors

\[ w = \sum_{n=1}^{N} \alpha_n y_n x_n \]

- The separating-hyperplane parameters \( w \) are a linear combination of the training data points \( x_n \)
- In the separable case, only the data points on the margin boundaries have nonzero \( \alpha_n \)
- In the non-separable case, it’s more complicated, but only data points near the margin have nonzero \( \alpha_n \)
- Either way, these data points are called the support vectors
The Dual

- Some crank-turning: Solve KKT for $w, \xi$ and plug back in
- This yields a very simple dual:

$$D(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{m=1}^{N} \sum_{n=1}^{N} \alpha_m \alpha_n y_m y_n x_m^T x_n.$$  

To be *maximized* with constraints:
$$\sum_{n=1}^{N} \alpha_n y_n = 0 \quad \text{and} \quad 0 \leq \alpha_n \leq \gamma$$

- $D$ turns out to be concave. Minimize $-D$ for a convex program
- *Key feature:* $x_n$ appears only in inner products with other $x_m$
- This will have very important consequences