EECS 545: Machine Learning

Lectures 9 & 10. Kernel methods: Support Vector Machines

Honglak Lee

02/10/2025

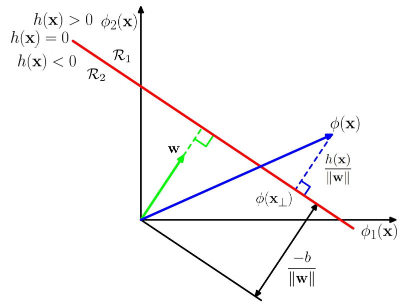
Overview

- Support Vector Machine (SVM)
- Soft-margin SVM
- Primal optimization
  - Soft-margin SVM
- Dual optimization (next lecture)
  - hard-margin SVM
  - soft-margin SVM

Support Vector Machines: Motivation and Formulation

Linear Discriminant Function

$h(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$

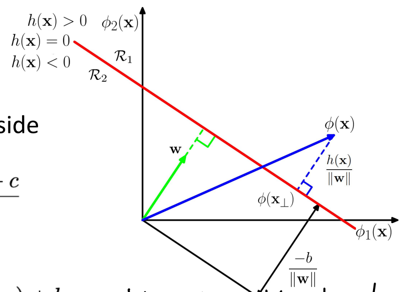


- Decision boundary is the hyperplane
  $\mathbf{w}^\top \phi(\mathbf{x}) + b = 0$
  - **w** determines direction
  - *b* determines offset

Distance of a point from a hyperplane

- 2D Case:
  - Line: $ax + by + c = 0$
  - Point: (x_0, y_0)
  - +/- depending on which side of line

$$\text{distance} = \frac{ax_0 + by_0 + c}{\sqrt{a^2 + b^2}}$$



- M - dimensional:
  - Hyperplane: $h(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$
  - Point: $\phi(\mathbf{x})$

$$\text{distance} = \frac{\mathbf{w}^\top \phi(\mathbf{x}) + b}{\|\mathbf{w}\|}$$

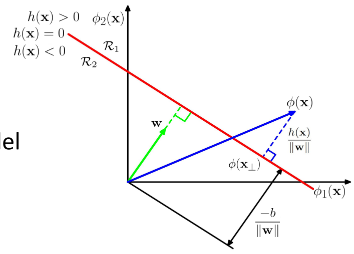
$\forall x,\ \phi(x) - d\frac{w}{\|w\|} \in$ hyperplane

$\Rightarrow w^T(\phi(x) - d\frac{w}{\|w\|}) + c = 0$ $\quad w^T z + c = 0$

$\Rightarrow d = \ldots$

Distance of a point from a hyperplane

- Derivation:
  - Let $\phi(\mathbf{x}_\perp)$ be the point on the hyperplane closest to $\phi(\mathbf{x})$
  - $\phi(\mathbf{x}) - \phi(\mathbf{x}_\perp)$ is perpendicular to the hyperplane and hence parallel to $\mathbf{w}$
  - Distance = $\pm\|\phi(\mathbf{x}) - \phi(\mathbf{x}_\perp)\|$

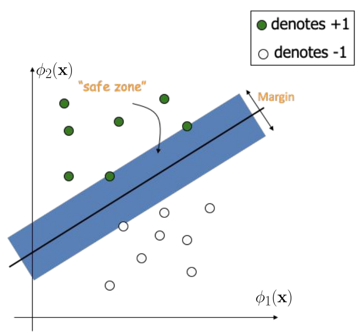


  - Note that $\mathbf{w}^\top(\phi(\mathbf{x}) - \phi(\mathbf{x}_\perp)) = \|\mathbf{w}\|\|\phi(\mathbf{x}) - \phi(\mathbf{x}_\perp)\|\cos(0)$
  - Thus, $\|\phi(\mathbf{x}) - \phi(\mathbf{x}_\perp)\| = \dfrac{\mathbf{w}^\top\phi(\mathbf{x}) - \mathbf{w}^\top\phi(\mathbf{x}_\perp)}{\|\mathbf{w}\|}$

$$= \frac{\mathbf{w}^\top\phi(\mathbf{x}) + b}{\|\mathbf{w}\|} \qquad \because \mathbf{w}^\top\phi(\mathbf{x}_\perp) + b = 0$$

Maximum Margin Classifier

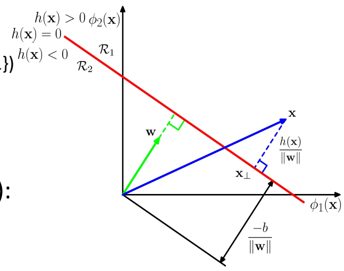
- The linear discriminant function (classifier) with the maximum margin is a good classifier.
- Margin is defined as the width that the boundary could be increased by before hitting a data point
- Why is it the "good" one?
  - Robust to outliers and thus strong generalization ability



Maximum Margin Classifier

- Distance from $\phi(\mathbf{x})$ to the hyperplane $\mathbf{w}^\top\phi(\mathbf{x}) + b = 0$
  (assuming data is linearly separable, y ∈ {-1, 1})

$$\frac{y(\mathbf{w}^\top\phi(\mathbf{x}) + b)}{\|\mathbf{w}\|}$$



- Margin (defined over training data):

$$\min_n \frac{y^{(n)}(\mathbf{w}^\top\phi(\mathbf{x}^{(n)}) + b)}{\|\mathbf{w}\|}$$

± 1

Maximum Margin Classifier

- Optimization problem:

$$\underset{\mathbf{w},b}{\operatorname{argmax}} \left\{ \frac{1}{\|\mathbf{w}\|} \min_{n} \left[y^{(n)} \left(\mathbf{w}^\top \phi\left(\mathbf{x}^{(n)}\right) + b \right) \right] \right\}$$

$y = 1$

- Rescale **w** and b such that:

$$y^{(n)} \left(\mathbf{w}^\top \phi\left(\mathbf{x}^{(n)}\right) + b \right) \geq 1 \qquad n = 1, \dots, N$$

- Optimization is equivalent to:

$$\boxed{\begin{aligned} &\underset{\mathbf{w},b}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to } y^{(n)} \left(\mathbf{w}^\top \phi\left(\mathbf{x}^{(n)}\right) + b \right) \geq 1 \qquad n = 1, \dots, N \end{aligned}}$$

Maximum Margin Classifier

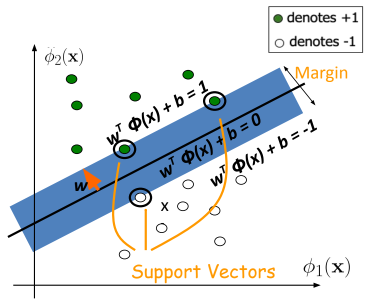
- Optimization problem:

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to

For $y^{(n)} = 1, \quad \mathbf{w}^\top \phi\left(\mathbf{x}^{(n)}\right) + b \geq 1$

For $y^{(n)} = -1, \quad \mathbf{w}^\top \phi\left(\mathbf{x}^{(n)}\right) + b \leq -1$



Solving the optimization problem

- Optimization problem (Hard SVM):

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to } y^{(n)} \left(\mathbf{w}^\top \phi\left(\mathbf{x}^{(n)}\right) + b \right) \geq 1 \qquad n = 1, \dots, N$$

- This is a constrained optimization problem.
  - We solve this using Lagrange multipliers (convex optimization).

- Problem of "Hard SVM":
  - formulation is based on the assumption that the training data linearly separable
  - What happens if this assumption is not satisfied? ⇒ ×
  - Note: Hard-margin SVM is not practically useful.

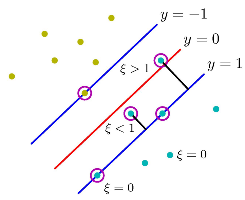
Support Vector Machines

- Hard SVM requires separable sets

$$y^{(n)} h\left(\mathbf{x}^{(n)}\right) - 1 \geq 0$$

- Soft SVM introduces *slack variables* for each data point

$$y^{(n)} h\left(\mathbf{x}^{(n)}\right) \geq 1 - \xi^{(n)}$$



Recall: $h(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$

Formulation of soft-margin SVM

- Maximize the margin, and also penalize for the slack variables

- Primal optimization
  - Optimization w.r.t

$$\min_{\mathbf{w},b,\xi} C \sum_{n=1}^{N} \xi^{(n)} + \frac{1}{2}\|\mathbf{w}\|^2$$

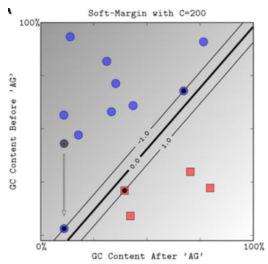
$$\text{subject to } y^{(n)} h\left(\mathbf{x}^{(n)}\right) \geq 1 - \xi^{(n)}, \forall n$$

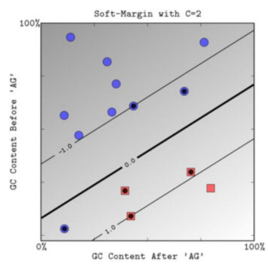
$$\xi^{(n)} \geq 0, \forall n$$

Recall: $h(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b$

Soft SVM

- A little slack can give much better margin.





Primal optimization

Optimization

- We can directly optimize the SVM objective function using gradient descent or stochastic gradient
  - Applicable when we have direct access to feature vectors $\phi(\mathbf{x})$
  - This is also called "linear SVM" (due to the use of linear kernels).

- Main idea
  - Convert the constraint into a penalty function

Converting constraints into penalty

- Note: objective is dependent on

$$\min_{\mathbf{w},b,\xi} C\sum_{n=1}^{N}\xi^{(n)} + \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to } \; y^{(n)}h\left(\mathbf{x}^{(n)}\right) \geq 1-\xi^{(n)}, \forall n$$

$$\xi^{(n)} \geq 0, \forall n$$

  - We want to minimize $\xi^{(n)}$ under the constraints

Recall: $h(\mathbf{x}) = \mathbf{w}^\top\phi(\mathbf{x}) + b$

Converting constraints into penalty

- Note: objective is dependent on $\xi^{(n)}$

$$\min_{\mathbf{w},b,\xi} C\sum_{n=1}^{N}\xi^{(n)} + \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to } \; y^{(n)}h\left(\mathbf{x}^{(n)}\right) \geq 1-\xi^{(n)}, \forall n$$

$$\xi^{(n)} \geq 0, \forall n$$

  - We want to minimize $\xi^{(n)}$ under the constraints

- Rewriting the constraints: for each n,

$$\xi^{(n)} \geq 1 - y^{(n)}h\left(\mathbf{x}^{(n)}\right)$$
$$\xi^{(n)} \geq 0$$

$$\Rightarrow \quad \xi^{(n)} \geq \max\left(0,\; 1-y^{(n)}h\left(\mathbf{x}^{(n)}\right)\right)$$

When equality holds, all constraints are satisfied and the objective is minimized!

Recall: $h(\mathbf{x}) = \mathbf{w}^\top\phi(\mathbf{x}) + b$

Converting constraints into penalty

- Original optimization problem

$$\min_{\mathbf{w},b,\xi} C\sum_{n=1}^{N}\xi^{(n)} + \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to } \; y^{(n)}h\left(\mathbf{x}^{(n)}\right) \geq 1-\xi^{(n)}, \forall n$$

$$\xi^{(n)} \geq 0, \forall n$$

Recall: $h(\mathbf{x}) = \mathbf{w}^\top\phi(\mathbf{x}) + b$

- An equivalent optimization problem

$$\min_{\mathbf{w},b} C\sum_{n=1}^{N}\max\left(0,\; 1-y^{(n)}h\left(\mathbf{x}^{(n)}\right)\right) + \frac{1}{2}\|\mathbf{w}\|^2$$

  - This can be optimized using gradient-based methods! (batch/stochastic gradient descent)

Gradients

- Computing the (sub) gradient with respect to w and b:
  - Recall: $h(\mathbf{x}) = \mathbf{w}^\top\phi(\mathbf{x}) + b$

$$\min_{\mathbf{w},b} C\sum_{n=1}^{N}\max\left(0,\; 1-y^{(n)}h\left(\mathbf{x}^{(n)}\right)\right) + \frac{1}{2}\|\mathbf{w}\|^2$$

$$\nabla_{\mathbf{w}}\mathcal{L} = -C\sum_{n=1}^{N} y^{(n)}\phi\left(\mathbf{x}^{(n)}\right)\mathbb{I}\left(1-y^{(n)}h\left(\mathbf{x}^{(n)}\right)\geq 0\right) + \mathbf{w}$$

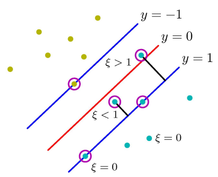
$$\nabla_{b}\mathcal{L} = -C\sum_{n=1}^{N} y^{(n)}\mathbb{I}\left(1-y^{(n)}h\left(\mathbf{x}^{(n)}\right)\geq 0\right)$$

- The gradient can be used to optimize ***w*** over the training data
  - Similar trick can be applied for stochastic gradient.

Support vectors

- In SVM, only the training points that have margin of 1 or less actually affect the final solution (***w***, ***b***).

- These are called "support vectors"



Summary

Hard SVM (Max Margin classifier): Assumes data is separable in feature space

$$\operatorname*{argmax}_{\mathbf{w},b}\left\{\frac{1}{\|\mathbf{w}\|}\min_{n}\left[y^{(n)}\left(\mathbf{w}^\top\phi\left(\mathbf{x}^{(n)}\right)+b\right)\right]\right\} \Leftrightarrow \operatorname*{argmin}_{\mathbf{w},b}\frac{1}{2}\|\mathbf{w}\|^2 \quad \text{s.t. } y^{(n)}\left(\mathbf{w}^\top\phi\left(\mathbf{x}^{(n)}\right)+b\right)\geq 1 \quad n=1,\ldots,N$$

Need to use constrained convex optimization to solve this problem

Relax the constraints

Soft SVM: No separability assumption: adding slack variables (for better robustness)

$$\min_{\mathbf{w},b,\xi} C\sum_{n=1}^{N}\xi^{(n)} + \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to } \; y^{(n)}h\left(\mathbf{x}^{(n)}\right) \geq 1-\xi^{(n)}, \forall n \quad \xi^{(n)} \geq 0, \forall n$$

$$\Leftrightarrow \min_{\mathbf{w},b} C\sum_{n=1}^{N}\max\left(0,\; 1-y^{(n)}h\left(\mathbf{x}^{(n)}\right)\right) + \frac{1}{2}\|\mathbf{w}\|^2$$

Primal problem can be solved using gradient methods.