EECS 545: Machine Learning

Lecture 9 & 10. Kernel methods: support vector machines

Honglak Lee

02/10/2020

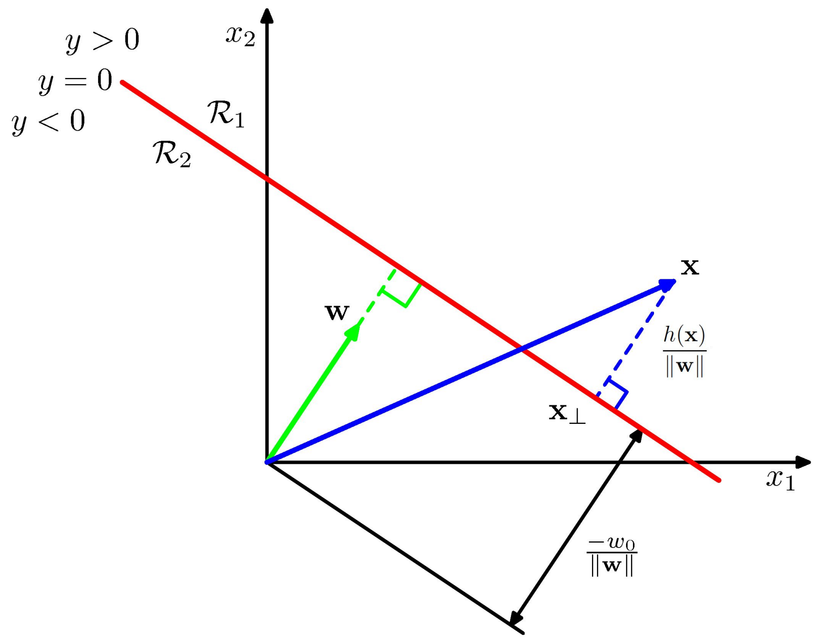
Overview

- 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

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



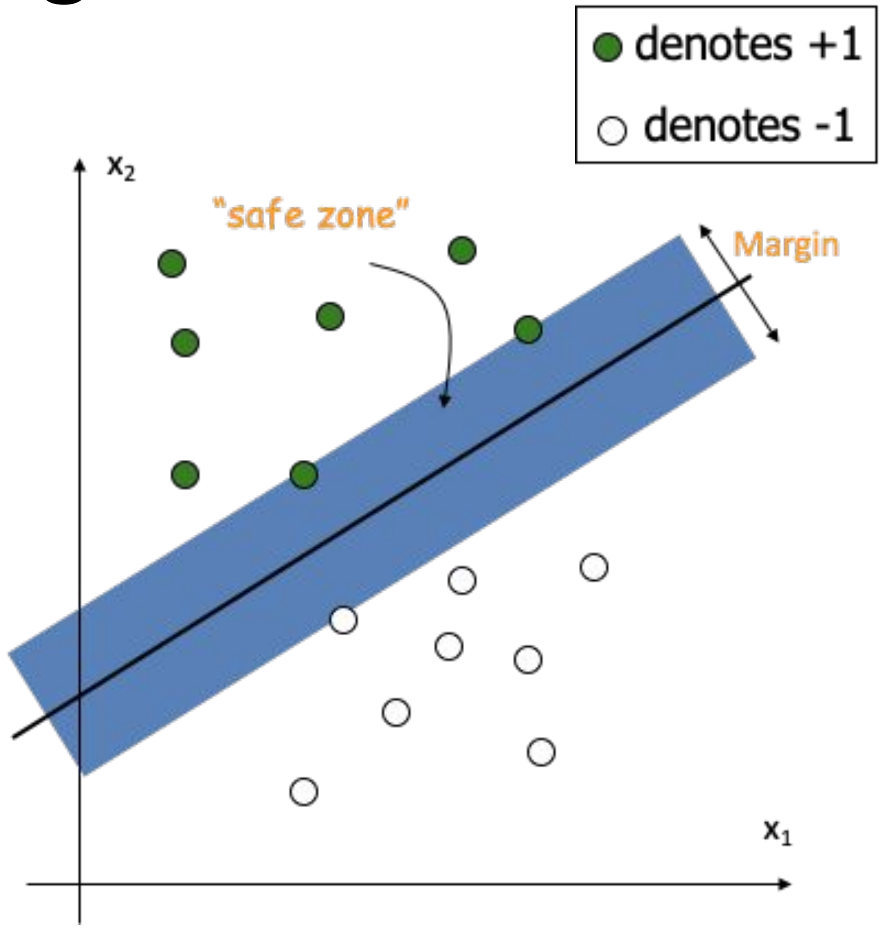
- Decision boundary is the hyperplane
  $$\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) + b = 0.$$
  - **w** determines direction
  - *b* determines offset

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 it is the "good"?
  – Robust to outliners and thus strong generalization ability



Maximum Margin Classifier

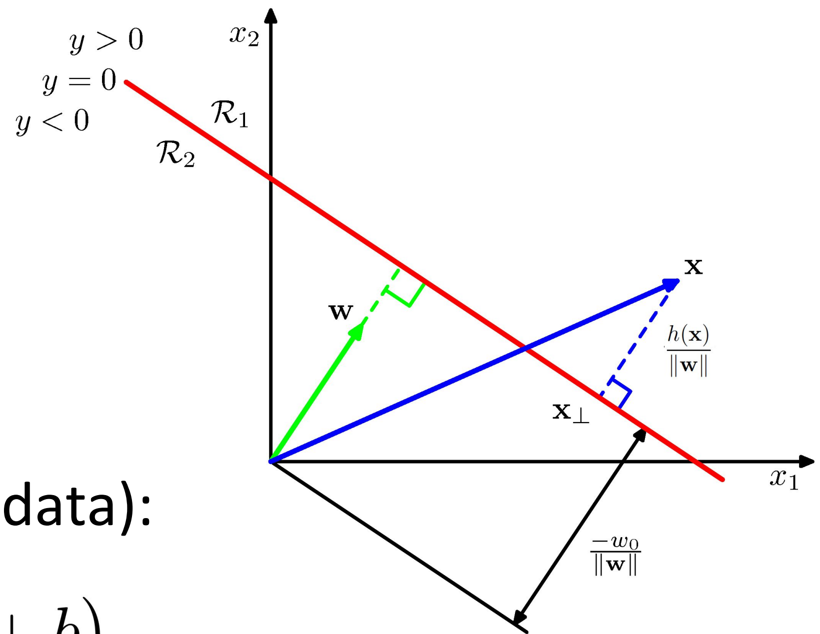
- Distance from $\phi(\mathbf{x})$ to the hyperplane $\mathbf{w}^T\phi(\mathbf{x}) + b = 0$.

  (assuming data are linearly separable)

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

- Margin (defined over training data):

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



Maximum Margin Classifier

- Optimization problem:

$$\arg\max_{w,b}\left\{\frac{1}{\|\mathbf{w}\|}\min_{n}\left[y^{(n)}\left(\mathbf{w}^T\phi\left(\mathbf{x}^{(n)}\right)+b\right)\right]\right\}$$

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

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

- Optimization is equivalent to:

$$\arg\min_{\mathbf{w},b}\frac{1}{2}\|\mathbf{w}\|^2$$

subject to $y^{(n)}\left(\mathbf{w}^T\phi\left(\mathbf{x}^{(n)}\right)+b\right)\geq 1 \;\; n=1,\ldots,N.$

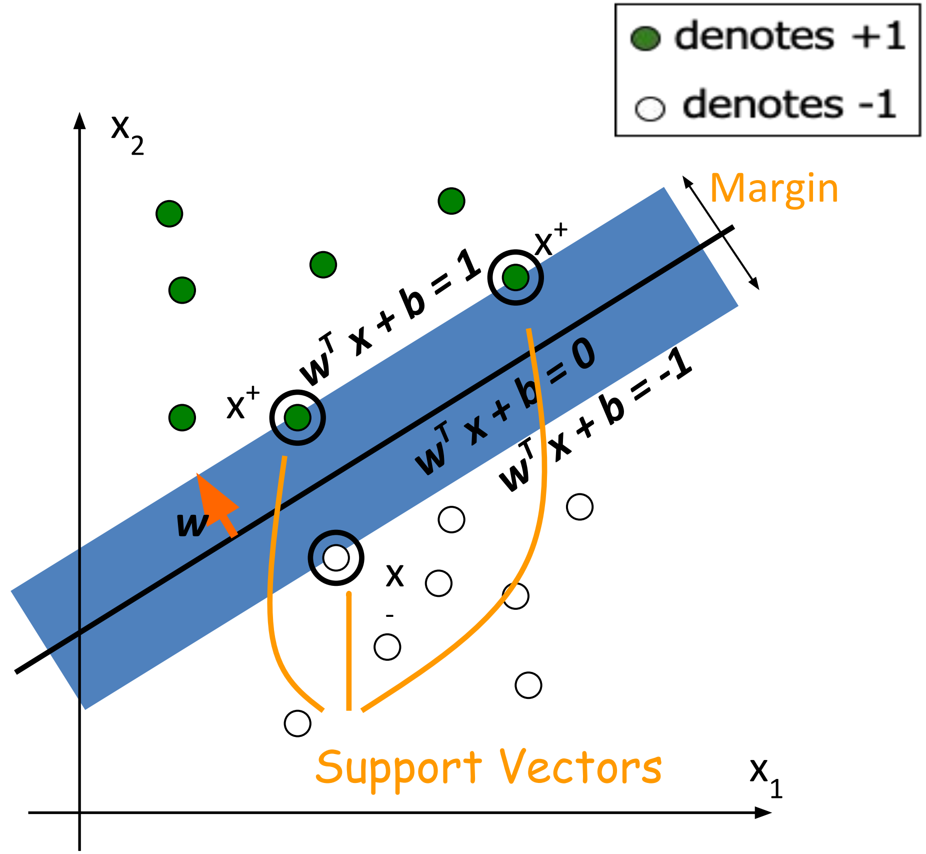
Maximum Margin Classifier

- Optimization problem:

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

subject to

$$\text{For } y^{(n)} = 1 \quad \mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b \geq 1$$
$$\text{For } y^{(n)} = -1 \quad \mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b \leq -1$$



Solving the optimization problem

- Optimization problem:

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{subject to } y^{(n)}\left(\mathbf{w}^T\phi\left(\mathbf{x}^{(n)}\right)+b\right)\geq 1, \quad n=1,\ldots,N.$$

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

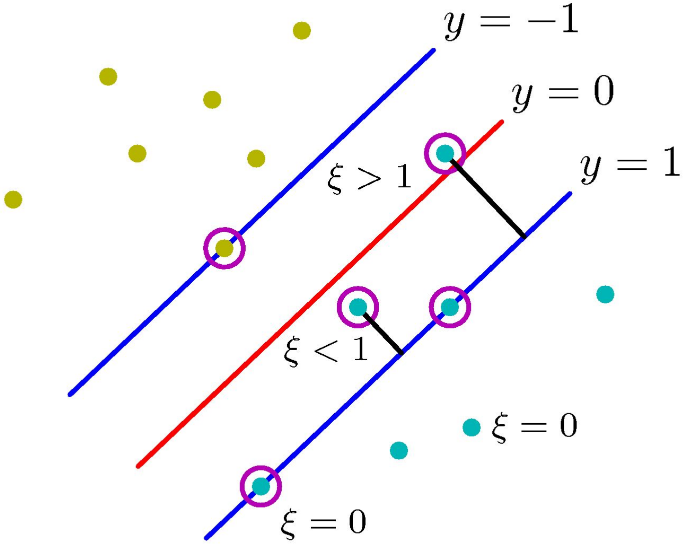
Support Vector Machines

- Hard SVM requires separable sets

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

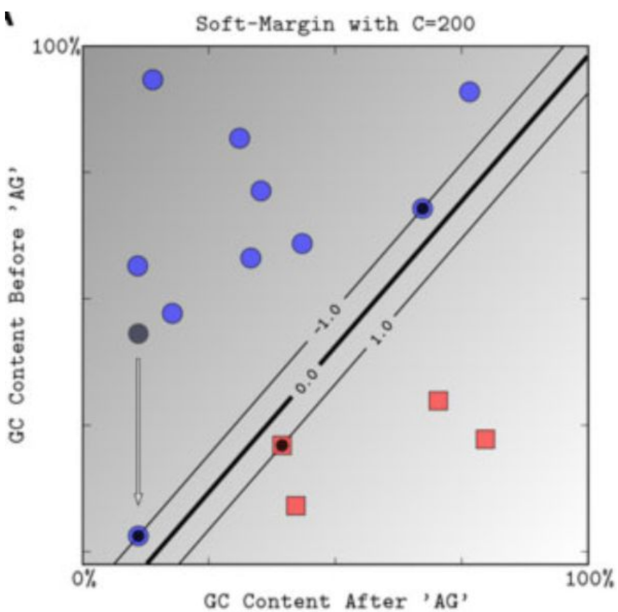
- Soft SVM introduces *slack variables* for each data point

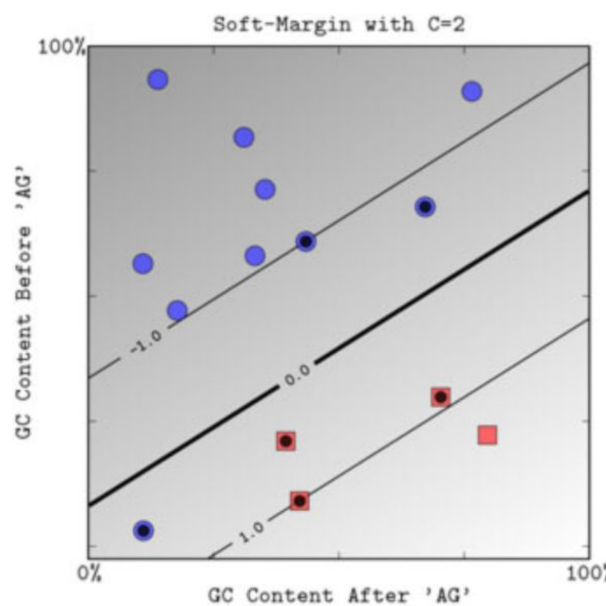
$$y^{(n)} h\left(\mathbf{x}^{(n)}\right) \geq 1 - \xi^{(n)}$$



Soft SVM

- A little slack can give much better margin.





Soft SVM

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

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

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

Formulation of soft-margin SVM

- Maximize the margin, and also penalize for the slack variables
- Primal optimization
  - Optimization w.r.t

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

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

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

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

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

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

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

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

Converting constraints into penalty

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

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

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

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

  – We want to <u>minimize</u> $\xi^{(n)}$ 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!

Converting constraints into penalty

- Original optimization problem

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & C\sum_{n=1}^{N}\xi^{(n)} + \tfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{Subject to} \quad & y^{(n)}h\left(\mathbf{x}^{(n)}\right) \geq 1-\xi^{(n)},\ \forall n \\ & \xi^{(n)} \geq 0, \forall n \end{aligned}$$

- An equivalent optimization problem

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

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

Gradients

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

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

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

- 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"