Support Vector Machines

CMSC 422

MARINE CARPUAT
marine@cs.umd.edu

Slides credit: Piyush Rai

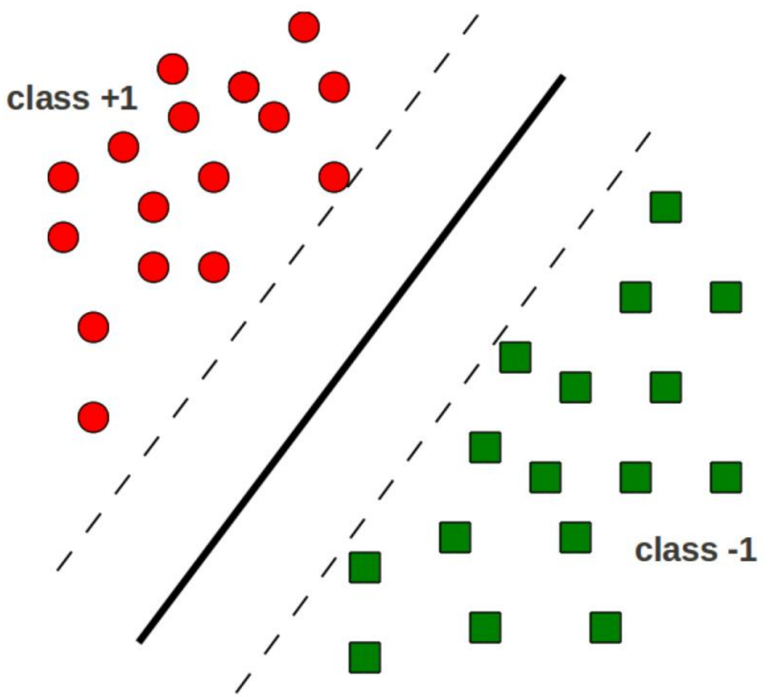
Back to linear classification

- Last time: we've seen that kernels can help capture non-linear patterns in data while keeping the advantages of a linear classifier

- Today: Support Vector Machines
  - A hyperplane-based classification algorithm
  - Highly influential
  - Backed by solid theoretical grounding (Vapnik & Cortes, 1995)
  - Easy to kernelize

The Maximum Margin Principle

- Find the hyperplane with maximum separation margin on the training data



Margin of a data set D

$$margin(\mathbf{D}, \boldsymbol{w}, b) = \begin{cases} \min_{(x,y)\in\mathbf{D}} y(\boldsymbol{w}\cdot\boldsymbol{x} + b) & \text{if } \boldsymbol{w} \text{ separates } \mathbf{D} \\ -\infty & \text{otherwise} \end{cases} \quad (3.8)$$

Distance between the hyperplane (w,b) and the nearest point in D

$$margin(\mathbf{D}) = \sup_{\boldsymbol{w},b} margin(\mathbf{D}, \boldsymbol{w}, b) \quad (3.9)$$

Largest attainable margin on D

Support Vector Machine (SVM)

A hyperplane based linear classifier defined by $\mathbf{w}$ and b

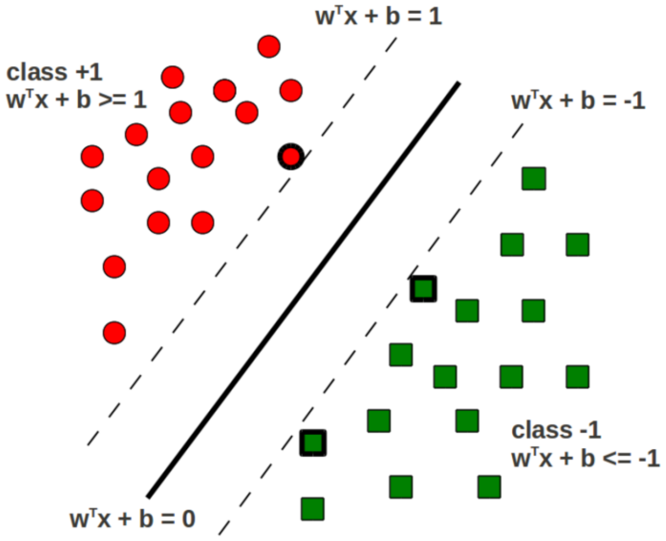
Prediction rule: $y = sign(\mathbf{w}^T\mathbf{x} + b)$

Given: Training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

Goal: Learn $\mathbf{w}$ and b that achieve the maximum margin

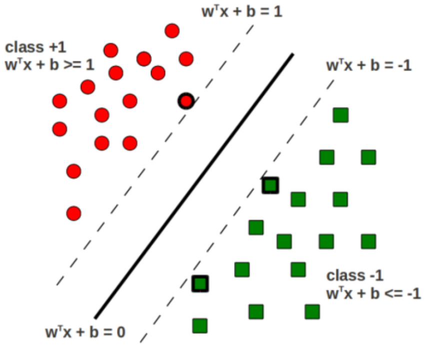
Characterizing the margin

Let's assume the entire training data is correctly classified by (**w**,b) that achieves max margin γ, then $\gamma = \frac{1}{||w||}$



The Optimization Problem

We want to maximize the margin $\gamma = \frac{1}{||\mathbf{w}||}$



Maximizing the margin γ = minimizing $||\mathbf{w}||$ (the norm)

Our optimization problem would be:

$$\text{Minimize} \quad f(\mathbf{w}, b) = \frac{||\mathbf{w}||^2}{2}$$
$$\text{subject to} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, \qquad n = 1, \dots, N$$

Large Margin = Good Generalization

- Intuitively, large margins mean good generalization
  - Large margin => small $||\mathbf{w}||$
  - small $||\mathbf{w}||$ => regularized/simple solutions

- (Learning theory gives a more formal justification)

Solving the SVM Optimization Problem

Our optimization problem is:

$$\text{Minimize} \quad f(\mathbf{w}, b) = \frac{||\mathbf{w}||^2}{2}$$
$$\text{subject to} \quad 1 \leq y_n(\mathbf{w}^T\mathbf{x}_n + b), \qquad n = 1, \ldots, N$$

Introducing Lagrange Multipliers α_n $(n = \{1, \ldots, N\})$, one for each constraint, leads to the **Lagrangian**:

$$\text{Minimize} \quad L(\mathbf{w}, b, \alpha) = \frac{||\mathbf{w}||^2}{2} + \sum_{n=1}^{N} \alpha_n \{1 - y_n(\mathbf{w}^T\mathbf{x}_n + b)\}$$
$$\text{subject to} \quad \alpha_n \geq 0; \quad n = 1, \ldots, N$$

Solving the SVM Optimization Problem

Take (partial) derivatives of L_P w.r.t. $\mathbf{w}$, b and set them to zero

$$\frac{\partial L_P}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n, \quad \frac{\partial L_P}{\partial b} = 0 \Rightarrow \sum_{n=1}^{N} \alpha_n y_n = 0$$

Substituting these in the Primal Lagrangian L_P gives the Dual Lagrangian

$$\text{Maximize} \quad L_D(\mathbf{w}, b, \alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{m,n=1}^{N} \alpha_m \alpha_n y_m y_n (\mathbf{x}_m^T \mathbf{x}_n)$$

$$\text{subject to} \quad \sum_{n=1}^{N} \alpha_n y_n = 0, \quad \alpha_n \geq 0; \quad n = 1, \ldots, N$$

Solving the SVM Optimization Problem

Take (partial) derivatives of L_P w.r.t. $\mathbf{w}$, b and set them to zero

A Quadratic Program for which many off-the-shelf solvers exist

$$= \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n, \qquad \frac{\partial L_P}{\partial b} = 0 \Rightarrow \sum_{n=1}^{N} \alpha_n y_n = 0$$

Substituting thes[...] the Primal Lagrangian L_P gives the Dual Lagrangian

$$\text{Maximize} \quad L_D(\mathbf{w}, b, \alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{m,n=1}^{N} \alpha_m \alpha_n y_m y_n (\mathbf{x}_m^T \mathbf{x}_n)$$

$$\text{subject to} \quad \sum_{n=1}^{N} \alpha_n y_n = 0, \quad \alpha_n \geq 0; \quad n = 1, \ldots, N$$

SVM: the solution!

Once we have the α_n's, $\mathbf{w}$ and b can be computed as:

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$$

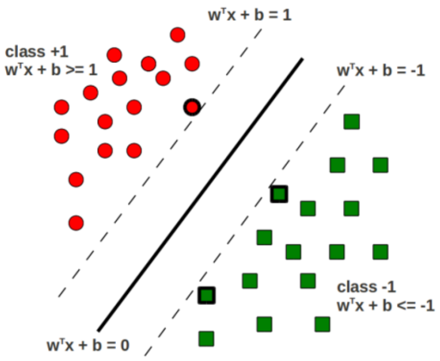
$$b = -\frac{1}{2}\left(\min_{n:y_n=+1} \mathbf{w}^T\mathbf{x}_n + \max_{n:y_n=-1} \mathbf{w}^T\mathbf{x}_n\right)$$

Note: Most α_n's in the solution are zero (sparse solution)

- Reason: Karush-Kuhn-Tucker (KKT) conditions
- For the optimal α_n's

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

- α_n is non-zero only if $\mathbf{x}_n$ lies on one of the two margin boundaries, i.e., for which $y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$
- These examples are called support vectors
- Support vectors "support" the margin boundaries



Support Vector Machines

- Find the max margin linear classifier for a dataset
- Discovers "support vectors", the training examples that "support" the margin boundaries
- Hard margin vs soft margin SVM
  - Hard margin: assme the data is linearly separable (today's lecture)
  - Soft margin: more general case (next time!)