CSCI-567: Machine Learning (Fall 2019)

Prof. Victor Adamchik

U of Southern California

Oct. 22, 2019

Outline

Outline

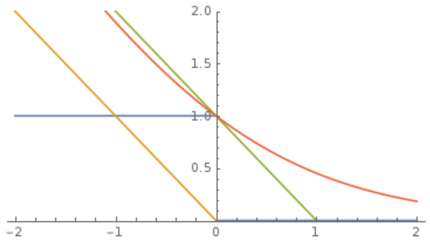
Support vector machines (SVM)

- One of the most commonly used classification algorithms
- Works well with the kernel trick
- Strong theoretical guarantees

We focus on **binary classification** here.

Primal formulation

In one sentence: linear model with L2 regularized hinge loss. Recall



- perceptron loss $\ell_{\text{perceptron}}(z) = \max\{0, -z\} \to$ Perceptron
- logistic loss $\ell_{\text{logistic}}(z) = \log(1 + \exp(-z)) \to$ logistic regression
- hinge loss $\ell_{\text{hinge}}(z) = \max\{0, 1 - z\} \to$ **SVM**

Primal formulation

For a linear model $(\boldsymbol{w}, b)$, this means

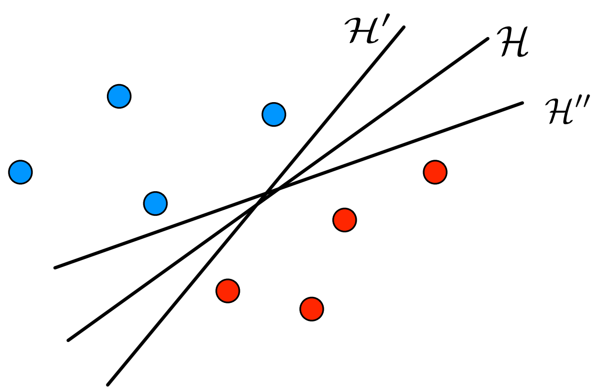
$$\min_{\boldsymbol{w},b} \sum_n \max\left\{0, 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)\right\} + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

- recall $y_n \in \{-1, +1\}$
- a nonlinear mapping ϕ is applied
- the bias/intercept term b is used explicitly (think about why after this lecture)

So why L2 regularized hinge loss?

Geometric motivation: separable case

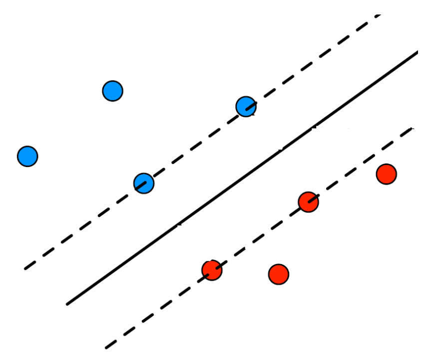
When data is **linearly separable**, there are *infinitely many hyperplanes with zero training error*:



So which one should we choose?

Intuition

The further away from data points the better.



How to formalize this intuition?

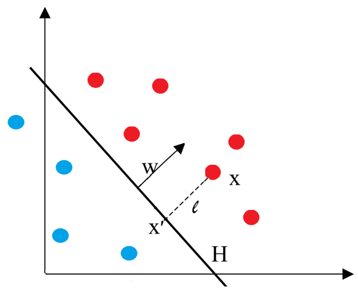
Distance to hyperplane

What is the **distance** from a point $\boldsymbol{x}$ to a hyperplane $H: \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b = 0$?

$\boldsymbol{w}$ is a normal vector perpendicular to H.

$\boldsymbol{x}' \in H$ is the **projection** of $\boldsymbol{x}$.

Then, $\boldsymbol{x}' = \boldsymbol{x} - \ell\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$, we go ℓ units parallel to $\boldsymbol{w}$.



Since $\boldsymbol{x}'$ belongs to a hyperplane, then

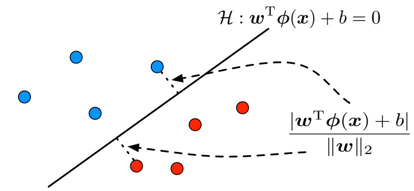
$$0 = \boldsymbol{w}^{\mathrm{T}}\left(\boldsymbol{x} - \ell\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}\right) + b = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} - \ell\|\boldsymbol{w}\| + b$$

From this we find the distance $\ell = \frac{|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b|}{\|\boldsymbol{w}\|_2}$.

Maximizing margin

Margin: the *smallest* distance from all training points to the hyperplane

$$\text{MARGIN OF } (\boldsymbol{w}, b) = \min_n \frac{|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b|}{\|\boldsymbol{w}\|_2}$$



The intuition "**the further away the better**" translates to solving

$$\max_{\boldsymbol{w},b} \min_n \frac{|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b|}{\|\boldsymbol{w}\|_2}$$

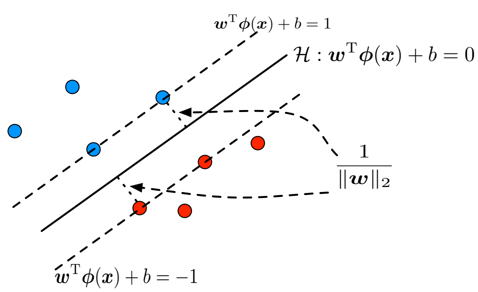
Rescaling

Note: rescaling $(\boldsymbol{w}, b)$ does not change the hyperplane at all.

We can thus always scale $(\boldsymbol{w}, b)$ s.t. $\min_n |\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b| = 1$

The margin then becomes

$$\begin{aligned}&\text{MARGIN OF } (\boldsymbol{w}, b)\\ &= \min_n \frac{|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b|}{\|\boldsymbol{w}\|_2}\\ &= \frac{1}{\|\boldsymbol{w}\|_2}\end{aligned}$$



Summary for separable data

Observe that $|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b| = y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b)$.

Therefore, for a separable training set, we aim to solve the following optimization problem

$$\max_{\boldsymbol{w},b} \frac{1}{\|\boldsymbol{w}\|_2} \quad \text{s.t.} \quad \min_n y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) = 1$$

This is equivalent to

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

subject to

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \geq 1, \quad \forall n$$

SVM is thus also called *max-margin* classifier. The constraints above are called *hard-margin* constraints.

General non-separable case

If data is not linearly separable, the constraints

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \geq 1, \quad \forall n$$

are obviously *not feasible*.

To deal with this issue, we relax them to **soft-margin** constraints:

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \geq 1 - \xi_n, \quad \forall n$$

where we introduce **slack variables** $\xi_n \geq 0$.

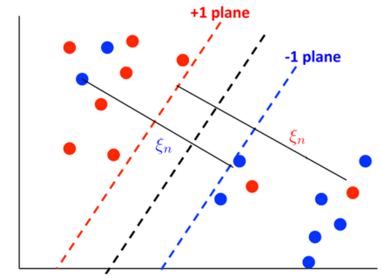
We want ξ_n to be as small as possible.

Meaning of slack variables ξ_n

The goal is to minimize the training errors (the number of misclassified points).

Instead we will minimize the distance between misclassified points and their correct hyperplane.

$0 < \xi_n \leq 1$ - data point falls within the margin on the correct side of the separating hyperplane; $\xi_n > 1$ - on the wrong side of the separating hyperplane.



We will introduce a hyperparameter C that represents a penalty for misclassifying points.

SVM Primal formulation

The objective function becomes

$$\min_{\boldsymbol{w}, b, \{\xi_n\}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

subject to

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \geq 1 - \xi_n, \quad \forall n$$
$$\xi_n \geq 0, \quad \forall\ n$$

where C is a new hyperparameter.

This formulation is called the soft-margin SVM.

Optimization

$$\min_{\boldsymbol{w}, b, \{\xi_n\}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

subject to

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \geq 1 - \xi_n, \quad \forall n$$
$$\xi_n \geq 0, \quad \forall\ n$$

- It is a convex (**quadratic** in fact) problem
- we can apply any convex optimization algorithms, e.g. SGD
- there are **more specialized and efficient** algorithms
- but usually we apply *kernel trick*, which requires solving the *dual problem*

Hinge Loss

How does this formulation

$$\min_{\boldsymbol{w},b,\{\xi_n\}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

subject to

$$y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)+b) \geq 1-\xi_n, \quad \forall n$$
$$\xi_n \geq 0, \quad \forall n$$

is related to L2 regularized hinge loss?

Equivalent form

Formulation

$$\min_{\boldsymbol{w},b,\{\xi_n\}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

subject to

$$\xi_n \geq 1-y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)+b), \quad \forall n$$
$$\xi_n \geq 0, \quad \forall n$$

is equivalent to

$$\min_{\boldsymbol{w},b,\{\xi_n\}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

subject to

$$\xi_n = \max\left\{0, 1-y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)+b)\right\}, \quad \forall n$$

Equivalent form

Formulation

$$\min_{\boldsymbol{w},b,\{\xi_n\}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

subject to

$$\xi_n = \max\left\{0, 1-y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)+b)\right\}, \quad \forall n$$

is equivalent to

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \max\left\{0, 1-y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)+b)\right\}$$

This is exactly minimizing L2 regularized hinge loss!

Outline

Example Optimization Problem

Web server company wants to buy new servers.

Standard Model

- \$400
- 300W power
- Two shelves of rack
- Handles 1000 hits/min

Cutting-edge model

- \$1600
- 500W power
- One shelf
- 2000 hits/min

Budget:

- \$36,800
- 44 shelves of space
- 12,200W power

Goal: maximize the number of hits we serve per minute.

The approach: linear programming

- Introduce variables x_1 and x_2 (the number of servers of each model we buy)
- The number of hits per minute we get is:

$$1000x_1 + 2000x_2$$

- The budget places three limitations on us:
  - The financial budget:

$$400x_1 + 1600x_2 \leq 36800$$

  - The number of shelves available:

$$2x_1 + x_2 \leq 44$$

  - Power used collectively

$$300x_1 + 500x_2 \leq 12200$$

Summarize the optimization problem

$$\max_{x_1,x_2} \quad 1000x_1 + 2000x_2$$

subject to:

$$\begin{aligned} 400x_1 + 1600x_2 &\leq 36800 \\ 2x_1 + x_2 &\leq 44 \\ 300x_1 + 500x_2 &\leq 12200 \\ x_1, x_2 &\geq 0 \end{aligned}$$

Various algorithms exist to solve the problem

Applications

Maximum Flow as a Linear Program

- Given a flow network with source, sink, edge capacities
- Flow through an edge must be at most capacity of edge.
- Flow into a vertex must equal flow out (Exceptions: source, sink)

maximize:	$\sum_{e \in out(s)} f_e$	where s is the source.
subject to:	$0 \leq f_e \leq c_e$	for all edges e
	$\sum_{e \in in(v)} f_e = \sum_{e \in out(v)} f_e$	for all vertices v except the source and sink.

Standard form

A linear program is in **standard** form if it is in the following form:

$$\max_{x_n} \sum_n c_n\, x_n$$

subject to

$$\sum_n a_{mn}\, x_n \leq b_m, \quad \forall m$$

$$x_n \geq 0, \quad \forall n$$

We can write the standard form more compactly:

$$\max_{\boldsymbol{x}} \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x}$$

subject to

$$A\, \boldsymbol{x} \leq \boldsymbol{b} \text{ and } \boldsymbol{x} \geq 0$$

Duality

Primal (in x):

maximize:	$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$
subject to:	$A\boldsymbol{x} \leq \boldsymbol{b}$
	$\boldsymbol{x} \geq \boldsymbol{0}$

Dual (in y):

minimize:	$\boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}$
subject to:	$A^T\boldsymbol{y} \geq \boldsymbol{c}$
	$\boldsymbol{y} \geq \boldsymbol{0}$

Weak Duality

Weak Duality: Let $\boldsymbol{x}$ be any feasible solution to the primal and $\boldsymbol{y}$ be any feasible solution for the dual. Then, $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} \leq \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}$.

Proof.

$$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{c} \leq \boldsymbol{x}^{\mathrm{T}}\left(A^{\mathrm{T}}\boldsymbol{y}\right) = (A\boldsymbol{x})^{\mathrm{T}}\boldsymbol{y} \leq \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}$$

The first inequality follows from the fact that y is feasible solution,

the second inequality follows since x is feasible solution.

Recall a flow network: for any flow and any cut, $|f| \leq cap(A, B)$

Strong Duality

Strong Duality: Let $\boldsymbol{x}$ be any feasible solution to the primal and $\boldsymbol{y}$ be any feasible solution for the dual. Then, $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{b}^{\mathrm{T}}\boldsymbol{y}$.

Proof.

The proof of this theorem is beyond the scope of this lecture.

Recall the max-flow min-cut theorem from 570.

Outline

Lagrangian duality

Extremely important and powerful tool in analyzing optimizations

We will introduce basic concepts and derive the **KKT conditions**

Applying it to SVM reveals an important aspect of the algorithm

Primal problem

Suppose we want to solve

$$\min_{\boldsymbol{w}} F(\boldsymbol{w}) \quad \text{s.t. } h_j(\boldsymbol{w}) \le 0 \quad \forall\, j \in [\mathsf{J}]$$

where functions $h_1, \ldots, h_{\mathsf{J}}$ define J **constraints**.

SVM primal formulation is clearly of this form with $\mathsf{J} = 2\mathsf{N}$ constraints:

$$\begin{aligned} F(\boldsymbol{w}, b, \{\xi_n\}) &= C \sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ h_n(\boldsymbol{w}, b, \{\xi_n\}) &= 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n \quad \forall\, n \in [\mathsf{N}] \\ h_{\mathsf{N}+n}(\boldsymbol{w}, b, \{\xi_n\}) &= -\xi_n \quad \forall\, n \in [\mathsf{N}] \end{aligned}$$

Lagrangian

Let us define the **Lagrangian** of the previous problem as:

$$L(\boldsymbol{w}, \{\lambda_j\}) = F(\boldsymbol{w}) + \sum_{j=1}^{\mathsf{J}} \lambda_j h_j(\boldsymbol{w})$$

where $\lambda_1, \ldots, \lambda_{\mathsf{J}} \ge 0$ are new variables (called **Lagrangian multipliers**).

Note that

$$\max_{\{\lambda_j\} \ge 0} L(\boldsymbol{w}, \{\lambda_j\}) = \begin{cases} F(\boldsymbol{w}) & \text{if } h_j(\boldsymbol{w}) \le 0 \quad \forall\, j \in [\mathsf{J}] \\ +\infty & \text{else} \end{cases}$$

and thus,

$$\min_{\boldsymbol{w}} \max_{\{\lambda_j\} \ge 0} L(\boldsymbol{w}, \{\lambda_j\}) \iff \min_{\boldsymbol{w}} F(\boldsymbol{w}) \ \text{ s.t. } h_j(\boldsymbol{w}) \le 0 \quad \forall\, j \in [\mathsf{J}]$$

Duality

We call this the **primal problem**

$$\min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

We define the **dual problem** by swapping the min and max:

$$\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

How are the primal and dual connected?

We will establish "**weak duality**" and "**strong duality**" for a non-linear optimization.

Weak Duality

Let $\boldsymbol{w}^*$ and $\{\lambda_j^*\}$ be the primal and dual solutions respectively, then

$$\begin{aligned}\max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right) &= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \\ &\leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) \\ &\leq \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}^*, \{\lambda_j\}\right) \\ &= \min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right)\end{aligned}$$

This is called "**weak duality**".

Strong duality

When $F, h_1, \ldots, h_m$ are convex, under some conditions (KKT conditions):

$$\min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right)$$

This is called "**strong duality**".

We will derive those conditions in the next slides.

Deriving the Karush-Kuhn-Tucker (KKT) conditions

Observe that if strong duality holds:

$$\begin{aligned}F(\boldsymbol{w}^*) &= \min_{\boldsymbol{w}} \max_{\{\lambda_j\}\geq 0} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \max_{\{\lambda_j\}\geq 0} \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j\}\right) = \\ &= \min_{\boldsymbol{w}} L\left(\boldsymbol{w}, \{\lambda_j^*\}\right) \leq L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* h_j(\boldsymbol{w}^*) \leq \\ &\leq F(\boldsymbol{w}^*)\end{aligned}$$

Implications:

- *all inequalities above have to be equalities!*
- last equality implies $\lambda_j^* h_j(\boldsymbol{w}^*) = 0$ for all $j \in [\mathsf{J}]$
- equality $\min_{\boldsymbol{w}} L(\boldsymbol{w}, \{\lambda_j^*\}) = L(\boldsymbol{w}^*, \{\lambda_j^*\})$ implies $\boldsymbol{w}^*$ is a **minimizer** of $L(\boldsymbol{w}, \{\lambda_j^*\})$ and thus has **zero gradient**.

The Karush-Kuhn-Tucker (KKT) conditions

If $\boldsymbol{w}^*$ and $\{\lambda_j^*\}$ are the primal and dual solution respectively, then:

Stationarity:

$$\nabla_{\boldsymbol{w}} L\left(\boldsymbol{w}^*, \{\lambda_j^*\}\right) = \nabla F(\boldsymbol{w}^*) + \sum_{j=1}^{\mathsf{J}} \lambda_j^* \nabla h_j(\boldsymbol{w}^*) = \mathbf{0}$$

Complementary slackness:

$$\lambda_j^* h_j(\boldsymbol{w}^*) = 0 \quad \text{for all } j \in [\mathsf{J}]$$

Feasibility:

$$h_j(\boldsymbol{w}^*) \le 0 \quad \text{and} \quad \lambda_j^* \ge 0 \quad \text{for all } j \in [\mathsf{J}]$$

These are *necessary conditions*. They are also *sufficient* when F is convex and $h_1, \ldots, h_{\mathsf{J}}$ are continuously differentiable convex functions.

Outline

Writing down the Lagrangian

Recall the primal formulation

$$\min_{\boldsymbol{w}, b, \{\xi_n\}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

subject to

$$1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n \le 0, \quad \forall n$$
$$-\xi_n \le 0, \quad \forall n$$

Lagrangian is

$$L\left(\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}\right) = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right)$$

where $\alpha_1, \ldots, \alpha_{\mathsf{N}} \ge 0$ and $\lambda_1, \ldots, \lambda_{\mathsf{N}} \ge 0$ are Lagrangian multipliers.

Applying the stationarity condition

$$L = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right)$$

$\exists$ primal and dual variables $\boldsymbol{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}$ s.t. $\nabla_{\boldsymbol{w}, b, \{\xi_n\}} L = \mathbf{0}$, which means

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_n \alpha_n y_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \mathbf{0}$$
$$\frac{\partial L}{\partial b} = -\sum_n \alpha_n y_n = 0$$
$$\frac{\partial L}{\partial \xi_n} = C - \lambda_n - \alpha_n = 0, \quad \forall n$$

Rewrite the Lagrangian in terms of dual variables

Replacing $\boldsymbol{w}$ by $\sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)$, after some simplification, we have

$$L = \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \sum_n C\,\xi_n - \sum_n \lambda_n \xi_n + \sum_n \alpha_n \left(1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) - \xi_n\right)$$
$$= \frac{1}{2}\|\sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 + \sum_n C\,\xi_n - \sum_n \lambda_n \xi_n + \sum_n \alpha_n - \sum_n \alpha_n \xi_n -$$
$$\sum_n \alpha_n y_n \left(\left(\sum_m y_m \alpha_m \boldsymbol{\phi}(\boldsymbol{x}_m) \right)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b \right)$$

Rewrite the Lagrangian in terms of dual variables

Since $C = \lambda_n + \alpha_n$ (see slide 40) , we get

$$L = \frac{1}{2}\|\sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 + \sum_n \alpha_n$$
$$- \sum_n \alpha_n y_n \left(\left(\sum_m y_m \alpha_m \boldsymbol{\phi}(\boldsymbol{x}_m) \right)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n) + b \right)$$

Rewrite the Lagrangian in terms of dual variables

Since $\sum_n \alpha_n y_n = 0$ (see slide 40) , we have

$$L = \frac{1}{2}\|\sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 + \sum_n \alpha_n$$
$$- \sum_n \alpha_n y_n \left(\sum_m y_m \alpha_m \boldsymbol{\phi}(\boldsymbol{x}_m) \right)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n)$$

which could be further simplified

$$L = \sum_n \alpha_n + \frac{1}{2}\|\sum_n y_n \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)\|_2^2 - \sum_{m,n} \alpha_n \alpha_m y_m y_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n)$$
$$= \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} \alpha_n \alpha_m y_m y_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n)$$

The dual formulation

So the **dual formulation of SVM** is:

$$\max_{\{\alpha_n\},\{\lambda_n\}} \quad \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_n)$$

subject to (see slide 40)

$$\sum_n \alpha_n y_n = 0,$$
$$C - \lambda_n - \alpha_n = 0,$$
$$\alpha_n \geq 0, \quad \forall\ n$$
$$\lambda_n \geq 0, \quad \forall\ n$$

Now it is clear that with a **kernel function** for the mapping ϕ, we can kernelize SVM. That is the reason why we need the dual SVM.

The dual formulation

The last three constraints can be simplified, therefore the **dual formulation of SVM** can be written as

$$\max_{\{\alpha_n\}} \quad \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

subject to

$$\sum_n \alpha_n y_n = 0,$$
$$0 \le \alpha_n \le C, \quad \forall\, n$$

wheher $k(x, x')$ is a kernel.

Recover the primal solution

But how do we predict given the dual solution $\{\alpha_n^*\}$? Need to figure out the primal solution $\boldsymbol{w}^*$ and b^*.

Based on previous observation (see slide 40,

$$\boldsymbol{w}^* = \sum_n \alpha_n^* y_n \boldsymbol{\phi}(\boldsymbol{x}_n) = \sum_{n:\alpha_n>0} \alpha_n^* y_n \boldsymbol{\phi}(\boldsymbol{x}_n)$$

A point with $\alpha_n^* > 0$ is called a "**support vector**". Hence the name SVM.

To identify b^*, we need to apply complementary slackness.

Applying complementary slackness

Recall the SVM primal formulation

$$\min_{\boldsymbol{w},b,\{\xi_n\}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

subject to

$$1 - \xi_n - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b) \le 0, \quad \forall n$$
$$-\xi_n \le 0, \quad \forall n$$

Recall complementary slackness (slide 37):

$$\lambda_j^* h_j(\boldsymbol{w}^*) = 0 \quad \text{for all } j \in [\mathsf{J}]$$

Therefore, for all n we have

$$\lambda_n^*\xi_n^* = 0, \quad \alpha_n^*\left(1 - \xi_n^* - y_n({\boldsymbol{w}^*}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)\right) = 0$$

Applying complementary slackness

Complementary slackness:

$$\lambda_n^*\xi_n^* = 0, \quad \alpha_n^*\left(1 - \xi_n^* - y_n({\boldsymbol{w}^*}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)\right) = 0$$

For some support vector $\boldsymbol{\phi}(\boldsymbol{x}_n)$ if we have $0 < \alpha_n^* < C$, then

$$\lambda_n^* = C - \alpha_n^* > 0$$

With the first condition we know $\xi_n^* = 0$.

With the second condition we know $1 = y_n({\boldsymbol{w}^*}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)$ and thus

$$b^* = y_n - {\boldsymbol{w}^*}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) = y_n - \sum_m y_m \alpha_m^* k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$

Having both $\boldsymbol{w}^*$ and b^* we can do prediction on a new point $\boldsymbol{x}$:

$$\mathrm{SGN}\left({\boldsymbol{w}^*}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}) + b^*\right) = \mathrm{SGN}\left(\sum_m y_m \alpha_m^* k(\boldsymbol{x}_m, \boldsymbol{x}) + b^*\right)$$

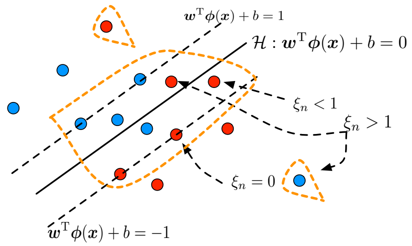
Geometric interpretation of support vectors

A support vector satisfies $\alpha_n^* \neq 0$ and

$$1 - \xi_n^* \leq y_n({\boldsymbol{w}^*}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*)$$

When

- $\xi_n^* = 0$, $y_n({\boldsymbol{w}^*}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b^*) = 1$ and thus the point is $1/\|\boldsymbol{w}^*\|_2$ away from the hyperplane.
- $\xi_n^* < 1$, the point is classified correctly but does not satisfy the large margin constraint.
- $\xi_n^* > 1$, the point is misclassified.

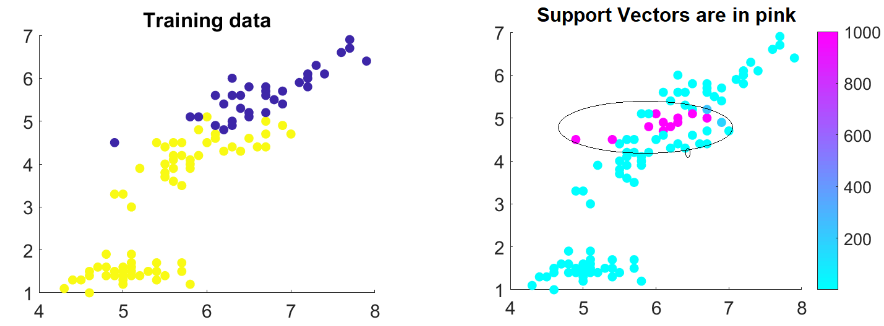


Support vectors (circled with the orange line) are *the only points that matter!*

An example

One drawback of kernel method: **non-parametric**, need to keep all training points potentially

However, for SVM, very often #support vectors $\ll$ N



Summary

Interpretation: maximize the margin

For separable data

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1, \quad \forall \ n \end{aligned}$$

For non-separable data

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \geq 1 - \xi_n, \quad \forall \ n \\ & \xi_n \geq 0, \quad \forall \ n \end{aligned}$$

where C is a hyperparameter and ξ_n are slack variables.

Summary

Interpretation: minimize loss

Minimize loss on all data

$$\min_{\boldsymbol{w}, b} \sum_n \max(0, 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b]) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

equivalently

$$\begin{aligned} \min_{\boldsymbol{w}, b, \{\xi_n\}} \quad & C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & 1 - y_n[\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b] \leq \xi_n, \quad \forall \ n \\ & \xi_n \geq 0, \quad \forall \ n \end{aligned}$$

Summary

SVM: **max-margin linear classifier**

Primal (equivalent to minimizing L2 regularized hinge loss):

$$\min_{\boldsymbol{w},b,\{\xi_n\}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

subject to

$$\xi_n \geq 1 - y_n(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n) + b), \quad \forall n$$
$$\xi_n \geq 0, \quad \forall n$$

Dual (kernelizable, reveals what training points are support vectors):

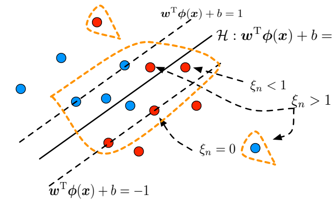
$$\max_{\{\alpha_n\}} \quad \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_m)^{\mathrm{T}}\boldsymbol{\phi}(\boldsymbol{x}_n)$$
$$\text{s.t.} \quad \sum_n \alpha_n y_n = 0 \quad \text{and} \quad 0 \leq \alpha_n \leq C, \quad \forall\, n$$

Geometric interpretation of support vectors

Some α_n will become zero

$$\min_{\boldsymbol{\alpha}} \quad \sum_n \alpha_n - \frac{1}{2}\sum_{m,n} y_m y_n \alpha_m \alpha_n k(\boldsymbol{x}_m, \boldsymbol{x}_n)$$
$$\text{s.t.} \quad 0 \leq \alpha_n \leq C, \quad \forall\, n$$
$$\sum_n \alpha_n y_n = 0$$

Nonzero α_n is called support vector



Support vectors are those being circled with the orange line. Removing them will change the solution.

Summary

Typical steps of applying Lagrangian duality

- start with a primal problem
- write down the Lagrangian (one dual variable per constraint)
- apply KKT conditions to find the connections between primal and dual solutions
- eliminate primal variables and arrive at the dual formulation
- maximize the Lagrangian with respect to dual variables
- recover the primal solutions from the dual solutions