

# CSCI567 Machine Learning (Spring 2025)

Haipeng Luo

University of Southern California

Feb 21, 2025

1 / 45

## Outline

- 1 Review of last lecture
- 2 Support vector machines (primal formulation)
- 3 A detour of Lagrangian duality
- 4 Support vector machines (dual formulation)
- 5 A bit about Quiz One

3 / 45

## Administration

HW2 is due on Feb 27th and will be graded before Quiz 1 (Mar 7th).

2 / 45

Review of last lecture

## Outline

- 1 Review of last lecture
- 2 Support vector machines (primal formulation)
- 3 A detour of Lagrangian duality
- 4 Support vector machines (dual formulation)
- 5 A bit about Quiz One

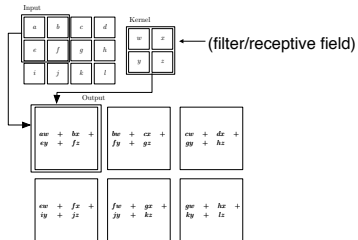
4 / 45

## Convolutional Neural Nets

### Typical architecture for CNNs:

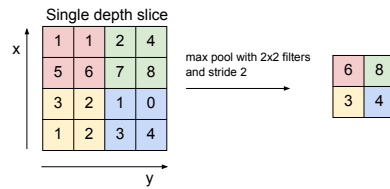
Input  $\rightarrow$  [[Conv  $\rightarrow$  ReLU]\*N  $\rightarrow$  Pool?]\*M  $\rightarrow$  [FC  $\rightarrow$  ReLU]\*Q  $\rightarrow$  FC

### 2D Convolution



©Stanford 2016

### MAX POOLING



Fei-Fei Li &amp; Justin Johnson &amp; Serena Yeung Lecture 5 - 73 April 18, 2017

5 / 45

## Kernel functions

**Definition:** a function  $k : \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}$  is called a *kernel function* if there exists a function  $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^M$  so that for any  $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^D$ ,

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$$

Examples we have seen

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}')^2$$

$$k(\mathbf{x}, \mathbf{x}') = \sum_{d=1}^D \frac{\sin(2\pi(x_d - x'_d))}{x_d - x'_d}$$

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^d$$

(polynomial kernel)

$$k(\mathbf{x}, \mathbf{x}') = e^{-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2\sigma^2}}$$

(Gaussian/RBF kernel)

6 / 45

## Kernelizing ML algorithms

Feasible as long as **only inner products are required**:

- regularized linear regression (dual formulation)

$$\phi(\mathbf{x})^T \mathbf{w}^* = \phi(\mathbf{x})^T \Phi^T (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y} \quad (\mathbf{K} = \Phi \Phi^T \text{ is kernel matrix})$$

- nearest neighbor, Perceptron, logistic regression, SVM, ...

7 / 45

## Outline

- Review of last lecture
- Support vector machines (primal formulation)
- A detour of Lagrangian duality
- Support vector machines (dual formulation)
- A bit about Quiz One

8 / 45

## Support vector machines (SVM)

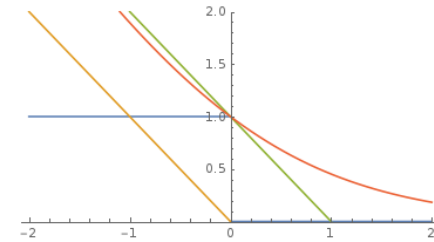
- most commonly used classification algorithms before deep learning
- works well with the kernel trick
- strong theoretical guarantees

We focus on **binary classification** here.

9 / 45

## Primal formulation

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



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

10 / 45

## Primal formulation

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

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

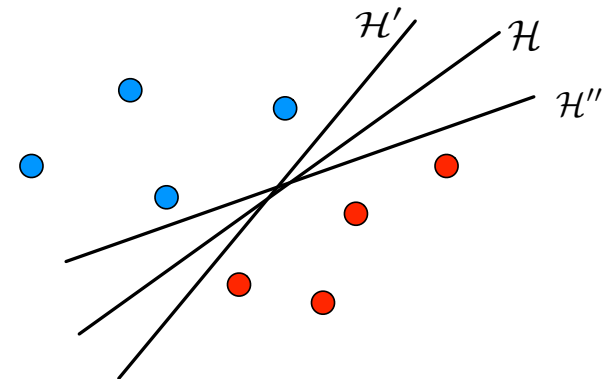
- recall  $y_n \in \{-1, +1\}$
- a nonlinear mapping  $\phi$  is applied
- the bias/intercept term  $b$  is used explicitly (think about why after this lecture)

*So why L2 regularized hinge loss?*

11 / 45

## Geometric motivation: separable case

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

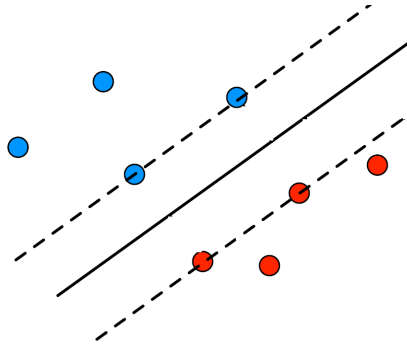


So which one should we choose?

12 / 45

## Intuition

The further away from data points the better.



How to formalize this intuition?

13 / 45

## Distance to hyperplane

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

Assume the **projection** is  $x - \ell \frac{w}{\|w\|_2}$ , then

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

and thus  $\ell = \frac{w^T x + b}{\|w\|_2}$ .

Therefore the distance is

$$\frac{|w^T x + b|}{\|w\|_2}$$

For a hyperplane that correctly classifies  $(x, y)$ , the distance becomes

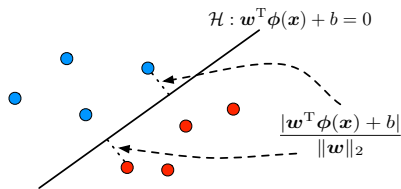
$$\frac{y(w^T x + b)}{\|w\|_2}$$

14 / 45

## Maximizing margin

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

$$\text{MARGIN OF } (w, b) = \min_n \frac{y_n(w^T \phi(x_n) + b)}{\|w\|_2}$$



The intuition “**the further away the better**” translates to solving

$$\max_{w, b} \min_n \frac{y_n(w^T \phi(x_n) + b)}{\|w\|_2} = \max_{w, b} \frac{1}{\|w\|_2} \min_n y_n(w^T \phi(x_n) + b)$$

15 / 45

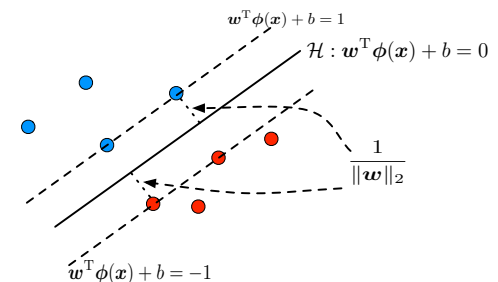
## Rescaling

**Note:** rescaling  $(w, b)$  does not change the hyperplane at all.

We can thus always scale  $(w, b)$  s.t.  $\min_n y_n(w^T \phi(x_n) + b) = 1$

The margin then becomes

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



16 / 45

## Summary for separable data

For a separable training set, we aim to solve

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

This is equivalent to

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1, \quad \forall n$$

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

17 / 45

## General non-separable case

If data is not linearly separable, the previous constraint

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

is obviously *not feasible*.

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

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

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

18 / 45

## SVM Primal formulation

We want  $\xi_n$  to be as small as possible too. The objective becomes

$$\min_{\mathbf{w}, b, \{\xi_n\}} \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_n \xi_n \\ \text{s.t.} \quad y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1 - \xi_n, \quad \forall n \\ \xi_n \geq 0, \quad \forall n$$

where  $C$  is a hyperparameter to balance the two goals.

19 / 45

## Equivalent form

### Formulation

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

is equivalent to

$$\min_{\mathbf{w}, b, \{\xi_n\}} C \sum_n \xi_n + \frac{1}{2} \|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad \max \{0, 1 - y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b)\} = \xi_n, \quad \forall n$$

20 / 45

## Equivalent form

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

$$\text{s.t.} \quad \max\{0, 1 - y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b)\} = \xi_n, \quad \forall n$$

is equivalent to

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

and

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

with  $\lambda = 1/C$ . *This is exactly minimizing L2 regularized hinge loss!*

21 / 45

## Outline

- 1 Review of last lecture
- 2 Support vector machines (primal formulation)
- 3 A detour of Lagrangian duality
- 4 Support vector machines (dual formulation)
- 5 A bit about Quiz One

23 / 45

## Optimization

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

$$\text{s.t.} \quad 1 - y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \leq \xi_n, \quad \forall n$$

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

- It is a convex (**quadratic** in fact) problem
- thus 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*

22 / 45

## Lagrangian duality

Extremely important and powerful tool in analyzing optimizations

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

- the derivation is not required for this course
- but the application of KKT conditions is required

Applying it to SVM reveals an important aspect of the algorithm

24 / 45

## Primal problem

Suppose we want to solve

$$\min_{\mathbf{w}} F(\mathbf{w}) \quad \text{s.t. } h_j(\mathbf{w}) \leq 0 \quad \forall j \in [J]$$

where functions  $h_1, \dots, h_J$  define  $J$  **constraints**.

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

$$F(\mathbf{w}, b, \{\xi_n\}) = C \sum_n \xi_n + \frac{1}{2} \|\mathbf{w}\|_2^2$$

$$h_n(\mathbf{w}, b, \{\xi_n\}) = 1 - y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \xi_n \quad \forall n \in [N]$$

$$h_{N+n}(\mathbf{w}, b, \{\xi_n\}) = -\xi_n \quad \forall n \in [N]$$

25 / 45

## Lagrangian

The **Lagrangian** of the previous problem is defined as:

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

where  $\lambda_1, \dots, \lambda_J \geq 0$  are called **Lagrange multipliers**.

Note that

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

and thus,

$$\min_{\mathbf{w}} \max_{\{\lambda_j\} \geq 0} L(\mathbf{w}, \{\lambda_j\}) \iff \min_{\mathbf{w}} F(\mathbf{w}) \quad \text{s.t. } h_j(\mathbf{w}) \leq 0 \quad \forall j \in [J]$$

26 / 45

## Duality

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

$$\max_{\{\lambda_j\} \geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \{\lambda_j\})$$

*How are the primal and dual connected?* Let  $\mathbf{w}^*$  and  $\{\lambda_j^*\}$  be the primal and dual solutions respectively, then

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

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

27 / 45

## Strong duality

When  $F, h_1, \dots, h_J$  are convex, under some mild conditions:

$$\min_{\mathbf{w}} \max_{\{\lambda_j\} \geq 0} L(\mathbf{w}, \{\lambda_j\}) = \max_{\{\lambda_j\} \geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \{\lambda_j\})$$

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

28 / 45

## Deriving the Karush-Kuhn-Tucker (KKT) conditions

Observe that if strong duality holds:

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

Implications:

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

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

29 / 45

## The Karush-Kuhn-Tucker (KKT) conditions

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

**Stationarity:**

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

**Complementary slackness:**

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

**Feasibility:**

$$h_j(\mathbf{w}^*) \leq 0 \quad \text{and} \quad \lambda_j^* \geq 0 \quad \text{for all } j \in [J]$$

These are *necessary conditions*. They are also *sufficient* when  $F$  is convex and  $h_1, \dots, h_J$  are continuously differentiable convex functions.

30 / 45

## Outline

- 1 Review of last lecture
- 2 Support vector machines (primal formulation)
- 3 A detour of Lagrangian duality
- 4 Support vector machines (dual formulation)
- 5 A bit about Quiz One

31 / 45

## Writing down the Lagrangian

Recall the primal formulation

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

**Lagrangian** is

$$\begin{aligned} L(\mathbf{w}, b, \{\xi_n\}, \{\alpha_n\}, \{\lambda_n\}) &= C \sum_n \xi_n + \frac{1}{2} \|\mathbf{w}\|_2^2 - \sum_n \lambda_n \xi_n \\ &\quad + \sum_n \alpha_n (1 - y_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \xi_n) \end{aligned}$$

where  $\alpha_1, \dots, \alpha_N \geq 0$  and  $\lambda_1, \dots, \lambda_N \geq 0$  are Lagrange multipliers.

32 / 45



## Applying the stationarity condition

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

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

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_n y_n \alpha_n \phi(\mathbf{x}_n) = \mathbf{0} \implies \mathbf{w} = \sum_n y_n \alpha_n \phi(\mathbf{x}_n)$$

$$\frac{\partial L}{\partial b} = - \sum_n \alpha_n y_n = 0 \quad \text{and} \quad \frac{\partial L}{\partial \xi_n} = C - \lambda_n - \alpha_n = 0, \quad \forall n$$

33 / 45

## Rewrite the Lagrangian in terms of dual variables

Replacing  $\mathbf{w}$  by  $\sum_n y_n \alpha_n \phi(\mathbf{x}_n)$  in the Lagrangian gives

$$\begin{aligned} L &= C \sum_n \xi_n + \frac{1}{2} \|\mathbf{w}\|_2^2 - \sum_n \lambda_n \xi_n + \sum_n \alpha_n (1 - y_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) - \xi_n) \\ &= C \sum_n \xi_n + \frac{1}{2} \left\| \sum_n y_n \alpha_n \phi(\mathbf{x}_n) \right\|_2^2 - \sum_n \lambda_n \xi_n + \\ &\quad \sum_n \alpha_n \left( 1 - y_n \left( \left( \sum_m y_m \alpha_m \phi(\mathbf{x}_m) \right)^T \phi(\mathbf{x}_n) + b \right) - \xi_n \right) \\ &= \sum_n \alpha_n + \frac{1}{2} \left\| \sum_n y_n \alpha_n \phi(\mathbf{x}_n) \right\|_2^2 - \sum_{m,n} \alpha_n \alpha_m y_m y_n \phi(\mathbf{x}_m)^T \phi(\mathbf{x}_n) \\ &\quad (\sum_n \alpha_n y_n = 0 \text{ and } C = \lambda_n + \alpha_n) \\ &= \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} \alpha_n \alpha_m y_m y_n \phi(\mathbf{x}_m)^T \phi(\mathbf{x}_n) \end{aligned}$$

34 / 45

## The dual formulation

To find the dual solutions, it amounts to solving

$$\begin{aligned} \max_{\{\alpha_n\}, \{\lambda_n\}} \quad & \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n \phi(\mathbf{x}_m)^T \phi(\mathbf{x}_n) \\ \text{s.t.} \quad & \sum_n \alpha_n y_n = 0 \\ & C - \lambda_n - \alpha_n = 0, \quad \alpha_n \geq 0, \quad \lambda_n \geq 0, \quad \forall n \end{aligned}$$

Note the last three constraints can be written as  $0 \leq \alpha_n \leq C$  for all  $n$ . So the final **dual formulation of SVM** is:

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

35 / 45

## Kernelizing SVM

Now it is clear that with a **kernel function**  $k$  for the mapping  $\phi$ , we can kernelize SVM as:

$$\begin{aligned} \max_{\{\alpha_n\}} \quad & \sum_n \alpha_n - \frac{1}{2} \sum_{m,n} y_m y_n \alpha_m \alpha_n k(\mathbf{x}_m, \mathbf{x}_n) \\ \text{s.t.} \quad & \sum_n \alpha_n y_n = 0 \quad \text{and} \quad 0 \leq \alpha_n \leq C, \quad \forall n \end{aligned}$$

Again, no need to compute  $\phi(\mathbf{x})$ . It is a **quadratic program** and many efficient optimization algorithms exist.

36 / 45

## Recover the primal solution

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

Based on previous observation,

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

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

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

37 / 45

## Applying complementary slackness

For all  $n$  we should have

$$\lambda_n^* \xi_n^* = 0, \quad \alpha_n^* (1 - \xi_n^* - y_n(w^{*\top} \phi(x_n) + b^*)) = 0$$

For any support vector  $\phi(x_n)$  with  $0 < \alpha_n^* < C$ ,  $\lambda_n^* = C - \alpha_n^* > 0$  holds.

- first condition implies  $\xi_n^* = 0$ .
- second condition implies  $1 = y_n(w^{*\top} \phi(x_n) + b^*)$  and thus

$$b^* = y_n - w^{*\top} \phi(x_n) = y_n - \sum_m \alpha_m^* y_m k(x_m, x_n)$$

Usually *average* over all  $n$  with  $0 < \alpha_n^* < C$  to stabilize computation.

The prediction on a new point  $x$  is therefore

$$\text{SGN}(w^{*\top} \phi(x) + b^*) = \text{SGN}\left(\sum_m \alpha_m^* y_m k(x_m, x) + b^*\right)$$

38 / 45

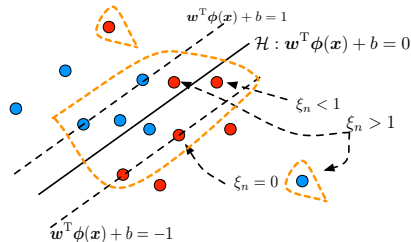
## Geometric interpretation of support vectors

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

$$1 - \xi_n^* - y_n(w^{*\top} \phi(x_n) + b^*) = 0$$

When

- $\xi_n^* = 0$ ,  $y_n(w^{*\top} \phi(x_n) + b^*) = 1$  and thus the point is  $1/\|w^*\|_2$  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.



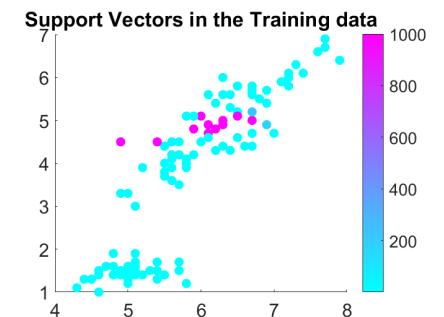
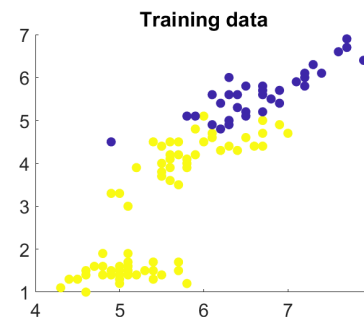
Support vectors (circled with the orange line) are *the only points that matter!*

39 / 45

## An example

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

For SVM, very often  $\# \text{support vectors} \ll N$



40 / 45

## Summary

SVM: **max-margin linear classifier**

**Primal** (equivalent to minimizing L2 regularized hinge loss):

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

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

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

41 / 45

A bit about Quiz One

## Outline

- 1 Review of last lecture
- 2 Support vector machines (primal formulation)
- 3 A detour of Lagrangian duality
- 4 Support vector machines (dual formulation)
- 5 A bit about Quiz One

43 / 45

## 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

42 / 45

A bit about Quiz One

## Coverage of Quiz 1

**Coverage:** mostly Lec 1-6, some multiple-choice questions from Lec 7. Will provide necessary formulas.

**Five problems** in total

- one problem of 15 multiple-choice **multiple-answer** questions
  - 0.5 point for selecting (not selecting) each correct (incorrect) answer
  - “which of the following is correct?” **does not imply one correct answer**
- four other homework-like problems, each has a couple sub-problems
  - linear regression, linear classifiers, backpropagation, kernel, SVM

**Tips:** expect to see variants of questions from discussion/homework

44 / 45

## Sample Quizzes

Two samples from 2021 and 2024 (available on course website):

- 2021 is slightly harder (especially Problem 5) and some formulas are not provided, because it was open-book/internet (due to covid)
- will work on **Problems 1 and 3(c-e) from 2021** now, and discuss solutions
- work on the rest in your own time, will keep discussing and release all solutions next week.