Machine Learning
(機器學習)

Lecture 10: Support Vector Machine (1)

Hsuan-Tien Lin (林軒田)

htlin@csie.ntu.edu.tw

Department of Computer Science
& Information Engineering

National Taiwan University
(國立台灣大學資訊工程系)

Roadmap

1. When Can Machines Learn?
2. Why Can Machines Learn?
3. How Can Machines Learn?
4. How Can Machines Learn Better?
5. Embedding Numerous Features: Kernel Models

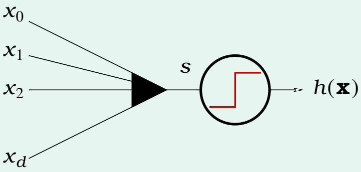
Lecture 10: Support Vector Machine (1)

- Large-Margin Separating Hyperplane
- Standard Large-Margin Problem
- Support Vector Machine
- Motivation of Dual SVM
- Lagrange Dual SVM
- Solving Dual SVM
- Messages behind Dual SVM

Linear Classification Revisited

PLA/pocket

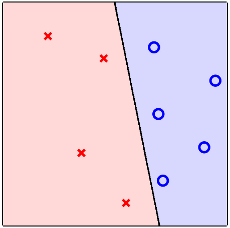
$$h(\mathbf{x}) = \text{sign}(s)$$



plausible err = 0/1

(small flipping noise)

minimize **specially**

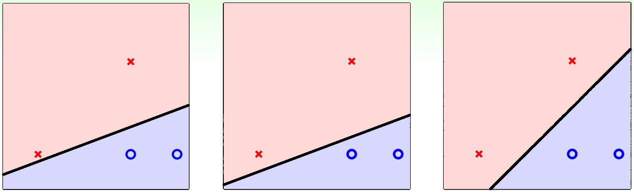


(linear separable)

linear (hyperplane) classifiers:

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x})$$

Which Line Is Best?

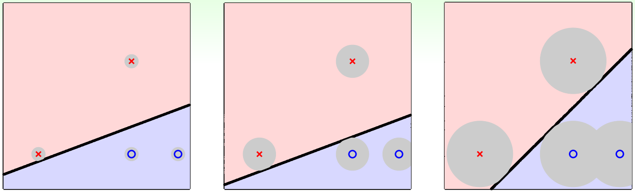


- PLA? depending on randomness
- VC bound? whichever you like!

$$E_{\text{out}}(\mathbf{w}) \leq \underbrace{E_{\text{in}}(\mathbf{w})}_{0} + \underbrace{\Omega(\mathcal{H})}_{d_{\text{VC}}=d+1}$$

You? **rightmost one, possibly :-)**

Why Rightmost Hyperplane?



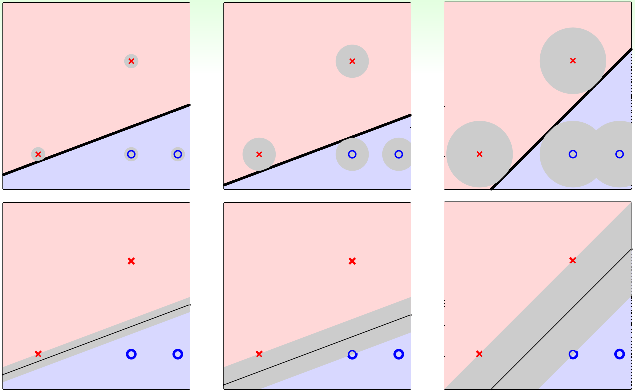
informal argument

if (Gaussian-like) noise on future $\mathbf{x} \approx \mathbf{x}_n$:

$\mathbf{x}_n$ further from hyperplane	distance to closest $\mathbf{x}_n$
$\Longleftrightarrow$ tolerate more noise	$\Longleftrightarrow$ amount of noise tolerance
$\Longleftrightarrow$ more robust to overfitting	$\Longleftrightarrow$ robustness of hyperplane

rightmost one: **more robust**
because of **larger distance to closest $\mathbf{x}_n$**

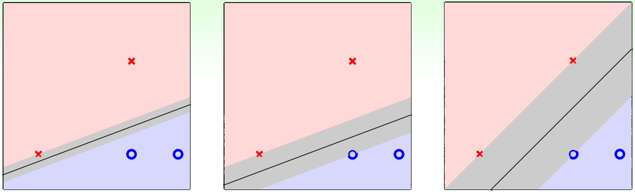
Fat Hyperplane



- robust separating hyperplane: **fat**
  —far from both sides of examples
- robustness $\equiv$ **fatness**: distance to closest $\mathbf{x}_n$

goal: find **fattest** separating hyperplane

Large-Margin Separating Hyperplane

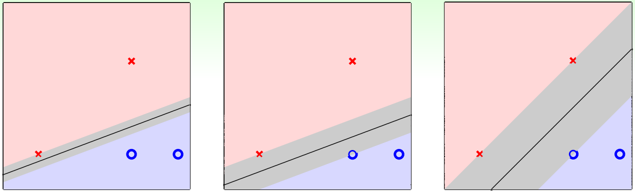


$$
\begin{aligned}
\max_{\mathbf{w}} \quad & \text{fatness}(\mathbf{w}) \\
\text{subject to} \quad & \mathbf{w} \text{ classifies every } (\mathbf{x}_n, y_n) \text{ correctly} \\
& \text{fatness}(\mathbf{w}) = \min_{n=1,\ldots,N} \text{distance}(\mathbf{x}_n, \mathbf{w})
\end{aligned}
$$

- fatness: formally called **margin**
- correctness: $y_n = \text{sign}(\mathbf{w}^T \mathbf{x}_n)$

goal: find **largest-margin** **separating** hyperplane

Large-Margin Separating Hyperplane



$$\max_{\mathbf{w}} \quad \text{margin}(\mathbf{w})$$
$$\text{subject to} \quad \text{every } y_n \mathbf{w}^T \mathbf{x}_n > 0$$
$$\text{margin}(\mathbf{w}) = \min_{n=1,\ldots,N} \text{distance}(\mathbf{x}_n, \mathbf{w})$$

- fatness: formally called **margin**
- correctness: $y_n = \text{sign}(\mathbf{w}^T \mathbf{x}_n)$

goal: find **largest-margin** **separating** hyperplane

Questions?

Distance to Hyperplane: Preliminary

$$\max_{\mathbf{w}} \quad \text{margin}(\mathbf{w})$$
$$\text{subject to} \quad \text{every } y_n \mathbf{w}^T \mathbf{x}_n > 0$$
$$\text{margin}(\mathbf{w}) = \min_{n=1,\ldots,N} \text{distance}(\mathbf{x}_n, \mathbf{w})$$

'shorten' x and w

distance needs w_0 and $(w_1, \ldots, w_d)$ differently (to be derived)

$$b = w_0 \qquad \cancel{x_0 = 1}$$

$$\begin{bmatrix} | \\ \mathbf{w} \\ | \end{bmatrix} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}; \quad \begin{bmatrix} | \\ \mathbf{x} \\ | \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$$

for this part: $h(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b)$

Distance to Hyperplane

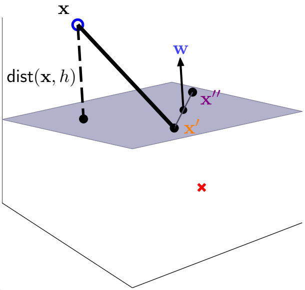
want: distance($\mathbf{x}$, b, $\mathbf{w}$), with hyperplane $\mathbf{w}^T\mathbf{x}' + b = 0$

consider $\mathbf{x}'$, $\mathbf{x}''$ on hyperplane

1. $\mathbf{w}^T\mathbf{x}' = -b$, $\mathbf{w}^T\mathbf{x}'' = -b$
2. $\mathbf{w} \perp$ hyperplane:

$$\left(\mathbf{w}^T \underbrace{(\mathbf{x}'' - \mathbf{x}')}_{\text{vector on hyperplane}}\right) = 0$$

3. distance = project $(\mathbf{x} - \mathbf{x}')$ to $\perp$ hyperplane



$$\text{distance}(\mathbf{x}, b, \mathbf{w}) = \left|\frac{\mathbf{w}^T}{\|\mathbf{w}\|}(\mathbf{x} - \mathbf{x}')\right| \overset{1}{=} \frac{1}{\|\mathbf{w}\|}|\mathbf{w}^T\mathbf{x} + b|$$

Distance to Separating Hyperplane

$$\text{distance}(\mathbf{x}, b, \mathbf{w}) = \frac{1}{\|\mathbf{w}\|}|\mathbf{w}^T\mathbf{x} + b|$$

- **separating** hyperplane: for every n

$$y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0$$

- distance to **separating** hyperplane:

$$\text{distance}(\mathbf{x}_n, b, \mathbf{w}) = \frac{1}{\|\mathbf{w}\|} y_n(\mathbf{w}^T\mathbf{x}_n + b)$$

$$\begin{aligned} \max_{b,\mathbf{w}} \quad & \text{margin}(b, \mathbf{w}) \\ \text{subject to} \quad & \text{every } y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0 \\ & \text{margin}(b, \mathbf{w}) = \min_{n=1,\ldots,N} \frac{1}{\|\mathbf{w}\|} y_n(\mathbf{w}^T\mathbf{x}_n + b) \end{aligned}$$

Margin of Special Separating Hyperplane

$$\max_{b,\mathbf{w}} \quad \text{margin}(b, \mathbf{w})$$
$$\text{subject to} \quad \text{every } y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0$$
$$\text{margin}(b, \mathbf{w}) = \min_{n=1,\dots,N} \frac{1}{\|\mathbf{w}\|} y_n(\mathbf{w}^T\mathbf{x}_n + b)$$

- $\mathbf{w}^T\mathbf{x} + b = 0$ same as $3\mathbf{w}^T\mathbf{x} + 3b = 0$: scaling does not matter
- **special** scaling: only consider separating $(b, \mathbf{w})$ such that

$$\min_{n=1,\dots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1 \Longrightarrow \text{margin}(b, \mathbf{w}) = \frac{1}{\|\mathbf{w}\|}$$

$$\max_{b,\mathbf{w}} \quad \frac{1}{\|\mathbf{w}\|}$$
$$\text{subject to} \quad \text{every } y_n(\mathbf{w}^T\mathbf{x}_n + b) > 0$$
$$\min_{n=1,\dots,N} y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$$

Standard Large-Margin Hyperplane Problem

$$\max_{b,\mathbf{w}} \quad \frac{1}{\|\mathbf{w}\|} \quad \text{subject to} \min_{n=1,\ldots,N} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$$

necessary constraints: $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$ for all n

original constraint: $\min_{n=1,\ldots,N} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$
want: optimal $(b, \mathbf{w})$ **here (inside)**

if optimal $(b, \mathbf{w})$ outside, e.g. $y_n(\mathbf{w}^T\mathbf{x}_n + b) > 1.126$ for all n
—can scale $(b, \mathbf{w})$ to "more optimal" $(\frac{b}{1.126}, \frac{\mathbf{w}}{1.126})$ **(contradiction!)**

final change: max $\Longrightarrow$ min, remove $\sqrt{}$, add $\frac{1}{2}$

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

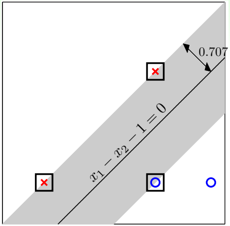
$$\text{subject to} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 \text{ for all } n$$

Questions?

Support Vector Machine (SVM)

optimal solution: $(w_1 = 1, w_2 = -1, b = -1)$

margin$(b, \mathbf{w})$ $= \frac{1}{\|\mathbf{w}\|} = \frac{1}{\sqrt{2}}$



- examples on boundary: 'locates' fattest hyperplane
  other examples: **not needed**
- call boundary example **support vector** (candidate)

support vector machine (SVM):
learn **fattest hyperplanes**
(with help of **support vectors**)

Solving General SVM

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$
$$\text{subject to} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 \text{ for all } n$$

- **not easy manually, of course :-)**
- gradient descent? **not easy with constraints**
- luckily:
  - (convex) quadratic objective function of $(b, \mathbf{w})$
  - linear constraints of $(b, \mathbf{w})$

—**quadratic programming**

quadratic programming (QP):
'easy' optimization problem

Quadratic Programming

optimal $(b, \mathbf{w}) = ?$

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{subject to} \quad y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, \quad \text{for } n = 1, 2, \ldots, N$$

optimal $\mathbf{u} \leftarrow \text{QP}(\mathrm{Q}, \mathbf{p}, \mathrm{A}, \mathbf{c})$

$$\min_{\mathbf{u}} \quad \frac{1}{2}\mathbf{u}^T\mathrm{Q}\mathbf{u} + \mathbf{p}^T\mathbf{u}$$

$$\text{subject to} \quad \mathbf{a}_m^T\mathbf{u} \geq c_m, \quad \text{for } m = 1, 2, \ldots, M$$

objective function: $\mathbf{u} = \begin{bmatrix} b \\ \mathbf{w} \end{bmatrix}; \mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}; \mathbf{p} = \mathbf{0}_{d+1}$

constraints: $\mathbf{a}_n^T = y_n \begin{bmatrix} 1 & \mathbf{x}_n^T \end{bmatrix}; c_n = 1; M = N$

SVM with general QP solver:
easy **if you've read the manual :-)**

SVM with QP Solver

Linear Hard-Margin SVM Algorithm

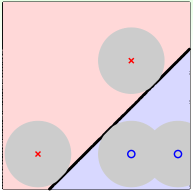
1. $\mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_d^T \\ \mathbf{0}_d & \mathrm{I}_d \end{bmatrix}$; $\mathbf{p} = \mathbf{0}_{d+1}$; $\mathbf{a}_n^T = y_n \begin{bmatrix} 1 & \mathbf{x}_n^T \end{bmatrix}$; $c_n = 1$
2. $\begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \leftarrow \mathrm{QP}(\mathrm{Q}, \mathbf{p}, \mathrm{A}, \mathbf{c})$
3. return b & $\mathbf{w}$ as g_{SVM}

- **hard-margin**: nothing violate 'fat boundary'
- **linear**: $\mathbf{x}_n$

want **non-linear**?
$\mathbf{z}_n = \mathbf{\Phi}(\mathbf{x}_n)$—**remember? :-)**

Why Large-Margin Hyperplane?

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$
$$\text{subject to} \quad y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1 \text{ for all } n$$



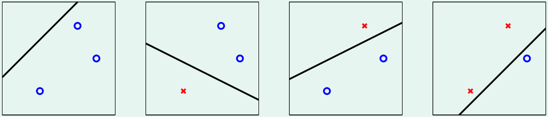
	minimize	constraint
regularization	E_{in}	$\mathbf{w}^T\mathbf{w} \leq C$
SVM	$\mathbf{w}^T\mathbf{w}$	$E_{\text{in}} = 0$ [and more]

SVM (large-margin hyperplane):
'weight-decay regularization' within $E_{\text{in}} = 0$

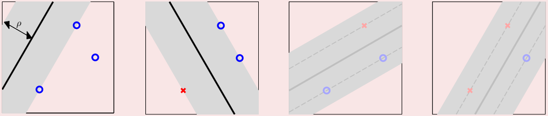
Large-Margin Restricts Dichotomies

consider 'large-margin algorithm' $\mathcal{A}_\rho$:
either **returns g with margin$(g) \geq \rho$ (if exists)**, or 0 otherwise

$\mathcal{A}_0$: like PLA $\Longrightarrow$ shatter 'general' 3 inputs



$\mathcal{A}_{1.126}$: more strict than SVM $\Longrightarrow$ cannot shatter any 3 inputs



fewer dichotomies $\Longrightarrow$ smaller 'VC dim.' $\Longrightarrow$ **better generalization**

VC Dimension of Large-Margin Algorithm

fewer dichotomies $\Longrightarrow$ smaller **'VC dim.'**
considers $d_{\mathrm{VC}}(\mathcal{A}_\rho)$ [data-dependent, need more than VC]
instead of $d_{\mathrm{VC}}(\mathcal{H})$ [data-independent, covered by VC]

generally, when $\mathcal{X}$ in radius-R hyperball:

$$d_{\mathrm{VC}}(\mathcal{A}_\rho) \le \min\left(\frac{R^2}{\rho^2}, d\right) + 1 \le \underbrace{d+1}_{d_{\mathrm{VC}}(\text{perceptrons})}$$

Benefits of Large-Margin Hyperplanes

	large-margin hyperplanes	hyperplanes	hyperplanes + feature transform $\mathbf{\Phi}$
#	even fewer	**not many**	many
boundary	simple	simple	**sophisticated**

- **not many** good, for d_{VC} and generalization
- **sophisticated** good, for possibly better E_{in}

a new possibility: non-linear SVM

	large-margin hyperplanes + numerous feature transform $\mathbf{\Phi}$
#	**not many**
boundary	**sophisticated**

Questions?

Non-Linear Support Vector Machine Revisited

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{s. t.} \quad y_n(\mathbf{w}^T \underbrace{\mathbf{z}_n}_{\mathbf{\Phi}(\mathbf{x}_n)} + b) \geq 1,$$

$$\text{for } n = 1, 2, \ldots, N$$

Non-Linear Hard-Margin SVM

1. $\mathrm{Q} = \begin{bmatrix} 0 & \mathbf{0}_{\tilde{d}}^T \\ \mathbf{0}_{\tilde{d}} & \mathrm{I}_{\tilde{d}} \end{bmatrix}$; $\mathbf{p} = \mathbf{0}_{\tilde{d}+1}$;
   $\mathbf{a}_n^T = y_n \begin{bmatrix} 1 & \mathbf{z}_n^T \end{bmatrix}$; $c_n = 1$
2. $\begin{bmatrix} b \\ \mathbf{w} \end{bmatrix} \leftarrow \mathrm{QP}(\mathrm{Q}, \mathbf{p}, \mathrm{A}, \mathbf{c})$
3. return $b \in \mathbb{R}$ & $\mathbf{w} \in \mathbb{R}^{\tilde{d}}$ with
   $g_{\text{SVM}}(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{\Phi}(\mathbf{x}) + b)$

- demanded: **not many** (large-margin), but **sophisticated** boundary (feature transform)
- QP with $\tilde{d} + 1$ variables and N constraints —challenging if $\tilde{d}$ large, **or infinite?! :-)**

goal: SVM **without dependence on** $\tilde{d}$

Todo: SVM 'without' $\tilde{d}$

Original SVM

(convex) QP of

- $\tilde{d} + 1$ variables
- N constraints

'Equivalent' SVM

(convex) QP of

- N variables
- $N + 1$ constraints

Warning: Heavy Math!!!!!!

- introduce some necessary math without rigor to help **understand SVM deeper**
- **'claim' some results** if details unnecessary
  —like how we 'claimed' Hoeffding

'Equivalent' SVM: based on some **dual problem** of Original SVM

Key Tool: Lagrange Multipliers

Regularization by Constrained-Minimizing E_{in}

$$\min_{\mathbf{w}} E_{\text{in}}(\mathbf{w}) \text{ s.t. } \mathbf{w}^T\mathbf{w} \le C$$

$\Longleftrightarrow$

Regularization by Minimizing E_{aug}

$$\min_{\mathbf{w}} E_{\text{aug}}(\mathbf{w}) = E_{\text{in}}(\mathbf{w}) + \frac{\lambda}{N}\mathbf{w}^T\mathbf{w}$$

- C equivalent to some $\lambda \ge 0$ by checking **optimality condition**

$$\nabla E_{\text{in}}(\mathbf{w}) + \frac{2\lambda}{N}\mathbf{w} = \mathbf{0}$$

- regularization: view λ as **given parameter instead of C**, and solve 'easily'
- dual SVM: view λ's as unknown given the constraints, and **solve them as variables instead**

how many λ's as variables?

N—one per constraint

Starting Point: Constrained to 'Unconstrained'

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{s.t.} \quad y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1,$$

$$\text{for } n = 1, 2, \ldots, N$$

Lagrange Function

with Lagrange multipliers $\cancel{\lambda_n}\ \alpha_n$,

$$\mathcal{L}(b, \mathbf{w}, \boldsymbol{\alpha}) = \underbrace{\frac{1}{2}\mathbf{w}^T\mathbf{w}}_{\text{objective}} + \sum_{n=1}^{N} \alpha_n \underbrace{(1 - y_n(\mathbf{w}^T\mathbf{z}_n + b))}_{\text{constraint}}$$

Claim

$$\text{SVM} \equiv \min_{b,\mathbf{w}} \left(\max_{\text{all } \alpha_n \geq 0} \mathcal{L}(b, \mathbf{w}, \boldsymbol{\alpha}) \right) = \min_{b,\mathbf{w}} \left(\infty \text{ if violate} ;\ \frac{1}{2}\mathbf{w}^T\mathbf{w} \text{ if feasible} \right)$$

- any 'violating' $(b, \mathbf{w})$: $\max\limits_{\text{all } \alpha_n \geq 0} \left(\square + \sum_n \alpha_n (\text{some positive}) \right) \to \infty$
- any 'feasible' $(b, \mathbf{w})$: $\max\limits_{\text{all } \alpha_n \geq 0} \left(\square + \sum_n \alpha_n (\text{all non-positive}) \right) = \square$

constraints now **hidden in** max

Questions?

Strong Duality of Quadratic Programming

$$\underbrace{\min_{b,\mathbf{w}} \left(\max_{\text{all } \alpha_n \geq 0} \mathcal{L}(b, \mathbf{w}, \boldsymbol{\alpha}) \right)}_{\text{equiv. to original (primal) SVM}} = \underbrace{\max_{\text{all } \alpha_n \geq 0} \left(\min_{b,\mathbf{w}} \mathcal{L}(b, \mathbf{w}, \boldsymbol{\alpha}) \right)}_{\text{Lagrange dual}}$$

- '$=$': **strong duality**, true for QP if
  - convex primal
  - feasible primal (true if $\boldsymbol{\Phi}$-separable)
  - linear constraints

—called constraint qualification

exists primal-dual optimal solution $(b, \mathbf{w}, \boldsymbol{\alpha})$ for both sides

Solving Lagrange Dual: Simplifications (1/2)

$$\max_{\text{all } \alpha_n \geq 0} \left(\min_{b,\mathbf{w}} \underbrace{\frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \alpha_n(1 - y_n(\mathbf{w}^T\mathbf{z}_n + b))}_{\mathcal{L}(b,\mathbf{w},\boldsymbol{\alpha})} \right)$$

- inner problem 'unconstrained', at optimal:
  $\frac{\partial \mathcal{L}(b,\mathbf{w},\boldsymbol{\alpha})}{\partial b} = 0 = -\sum_{n=1}^{N} \alpha_n y_n$
- no loss of optimality if solving with constraint $\sum_{n=1}^{N} \alpha_n y_n = 0$

but wait, *b* **can be removed**

$$\max_{\text{all } \alpha_n \geq 0, \sum y_n \alpha_n = 0} \left(\min_{b,\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \alpha_n(1 - y_n(\mathbf{w}^T\mathbf{z}_n)) - \cancel{\sum_{n=1}^{N} \alpha_n y_n \cdot b} \right)$$

Solving Lagrange Dual: Simplifications (2/2)

$$\max_{\text{all } \alpha_n \geq 0, \sum y_n \alpha_n = 0} \left(\min_{b,\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \alpha_n (1 - y_n(\mathbf{w}^T\mathbf{z}_n)) \right)$$

- inner problem 'unconstrained', at optimal: $\frac{\partial \mathcal{L}(b,\mathbf{w},\boldsymbol{\alpha})}{\partial w_i} = 0 = w_i - \sum_{n=1}^{N} \alpha_n y_n z_{n,i}$
- no loss of optimality if solving with constraint $\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{z}_n$

but wait!

$$\max_{\text{all } \alpha_n \geq 0, \sum y_n \alpha_n = 0, \mathbf{w} = \sum \alpha_n y_n \mathbf{z}_n} \left(\min_{b,\mathbf{w}} \frac{1}{2}\mathbf{w}^T\mathbf{w} + \sum_{n=1}^{N} \alpha_n - \mathbf{w}^T\mathbf{w} \right)$$

$$\Longleftrightarrow \max_{\text{all } \alpha_n \geq 0, \sum y_n \alpha_n = 0, \mathbf{w} = \sum \alpha_n y_n \mathbf{z}_n} -\frac{1}{2} \left\| \sum_{n=1}^{N} \alpha_n y_n \mathbf{z}_n \right\|^2 + \sum_{n=1}^{N} \alpha_n$$

KKT Optimality Conditions

$$\max_{\text{all } \alpha_n \geq 0, \sum y_n \alpha_n = 0, \mathbf{w} = \sum \alpha_n y_n \mathbf{z}_n} -\frac{1}{2} \| \sum_{n=1}^{N} \alpha_n y_n \mathbf{z}_n \|^2 + \sum_{n=1}^{N} \alpha_n$$

if primal-dual optimal $(b, \mathbf{w}, \boldsymbol{\alpha})$,

- primal feasible: $y_n(\mathbf{w}^T \mathbf{z}_n + b) \geq 1$
- dual feasible: $\alpha_n \geq 0$
- dual-inner optimal: $\sum y_n \alpha_n = 0$; $\mathbf{w} = \sum \alpha_n y_n \mathbf{z}_n$
- primal-inner optimal (at optimal all 'Lagrange terms' disappear):

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

—called **Karush-Kuhn-Tucker (KKT) conditions**, necessary for optimality [& sufficient here]

will use **KKT** to 'solve' $(b, \mathbf{w})$ from optimal $\boldsymbol{\alpha}$

Questions?

Dual Formulation of Support Vector Machine

$$\max_{\text{all } \alpha_n \geq 0, \sum y_n \alpha_n = 0, \mathbf{w} = \sum \alpha_n y_n \mathbf{z}_n} \quad -\frac{1}{2} \left\| \sum_{n=1}^{N} \alpha_n y_n \mathbf{z}_n \right\|^2 + \sum_{n=1}^{N} \alpha_n$$

standard hard-margin SVM dual

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \mathbf{z}_n^T \mathbf{z}_m - \sum_{n=1}^{N} \alpha_n$$

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

$$\alpha_n \geq 0, \text{for } n = 1, 2, \ldots, N$$

(convex) QP of N variables & $N + 1$ constraints, as promised

how to solve? **yeah, we know QP! :-)**

Dual SVM with QP Solver

optimal $\boldsymbol{\alpha} = ?$

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \mathbf{z}_n^T \mathbf{z}_m - \sum_{n=1}^{N} \alpha_n$$

$$\text{subject to} \quad \sum_{n=1}^{N} y_n \alpha_n = 0; \quad \alpha_n \ge 0, \text{ for } n = 1, 2, \ldots, N$$

optimal $\boldsymbol{\alpha} \leftarrow \text{QP}(\mathrm{Q}, \mathbf{p}, \mathrm{A}, \mathbf{c})$

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\boldsymbol{\alpha}^T \mathrm{Q} \boldsymbol{\alpha} + \mathbf{p}^T \boldsymbol{\alpha}$$

$$\text{subject to} \quad \mathbf{a}_i^T \boldsymbol{\alpha} \ge c_i, \text{ for } i = 1, 2, \ldots$$

- $q_{n,m} = y_n y_m \mathbf{z}_n^T \mathbf{z}_m$
- $\mathbf{p} = -\mathbf{1}_N$
- $\mathbf{a}_{\ge} = \mathbf{y}$, $\mathbf{a}_{\le} = -\mathbf{y}$; $\mathbf{a}_n^T = n$-th unit direction
- $c_{\ge} = 0$, $c_{\le} = 0$; $c_n = 0$

note: many solvers treat **equality ($\mathbf{a}_{\ge}$, $\mathbf{a}_{\le}$)** & **bound ($\mathbf{a}_n$)** constraints specially for **numerical stability**

Dual SVM with Special QP Solver

optimal $\boldsymbol{\alpha} \leftarrow \text{QP}(\boxed{\mathrm{Q_D}}, \mathbf{p}, \mathrm{A}, \mathbf{c})$

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\boldsymbol{\alpha}^T \boxed{\mathrm{Q_D}} \boldsymbol{\alpha} + \mathbf{p}^T\boldsymbol{\alpha}$$

subject to special equality and bound constraints

- $\boxed{q}_{n,m} = y_n y_m \mathbf{z}_n^T \mathbf{z}_m$, often non-zero
- if $N = 30,000$, dense $\mathrm{Q_D}$ (N by N symmetric) takes > 3G RAM
- need **special solver** for
    - not storing whole $\mathrm{Q_D}$
    - utilizing special constraints properly

  to scale up to large N

usually better to use **special solver** in practice

Optimal $(b, \mathbf{w})$

KKT conditions

if primal-dual optimal $(b, \mathbf{w}, \boldsymbol{\alpha})$,

- primal feasible: $y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1$
- dual feasible: $\alpha_n \geq 0$
- dual-inner optimal: $\sum y_n\alpha_n = 0$; $\mathbf{w} = \sum \alpha_n y_n \mathbf{z}_n$
- primal-inner optimal (at optimal all 'Lagrange terms' disappear):

$$\alpha_n(1 - y_n(\mathbf{w}^T\mathbf{z}_n + b)) = 0 \text{ (\textbf{complementary slackness})}$$

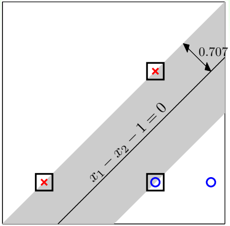
- optimal $\boldsymbol{\alpha} \Longrightarrow$ optimal $\mathbf{w}$? easy above!
- optimal $\boldsymbol{\alpha} \Longrightarrow$ optimal b? a range from primal feasible & equality from **comp. slackness** if one $\alpha_n > 0 \Rightarrow b = y_n - \mathbf{w}^T\mathbf{z}_n$

comp. slackness:
$\alpha_n > 0 \Rightarrow$ **on fat boundary (SV!)**

Questions?

Support Vectors Revisited

- on boundary: 'locates' fattest hyperplane; others: **not needed**
- examples with $\alpha_n > 0$: on boundary
- call $\alpha_n > 0$ examples $(\mathbf{z}_n, y_n)$ **support vectors** ~~(candidates)~~
- SV (positive α_n) $\subseteq$ SV candidates (on boundary)



- only SV needed to compute $\mathbf{w}$: $\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{z}_n = \sum_{\text{SV}} \alpha_n y_n \mathbf{z}_n$
- only SV needed to compute b: $b = y_n - \mathbf{w}^T \mathbf{z}_n$ with any SV $(\mathbf{z}_n, y_n)$

SVM: learn **fattest hyperplane**
by identifying **support vectors**
with **dual** optimal solution

Summary: Two Forms of Hard-Margin SVM

Primal Hard-Margin SVM

$$\min_{b,\mathbf{w}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{sub. to} \quad y_n(\mathbf{w}^T\mathbf{z}_n + b) \geq 1,$$

$$\text{for } n = 1, 2, \ldots, N$$

- $\tilde{d} + 1$ variables, N constraints —suitable when $\tilde{d} + 1$ small
- physical meaning: locate **specially-scaled** $(b, \mathbf{w})$

Dual Hard-Margin SVM

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\boldsymbol{\alpha}^T \mathrm{Q_D} \boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha}$$

$$\text{s.t.} \quad \mathbf{y}^T\boldsymbol{\alpha} = 0;$$

$$\alpha_n \geq 0 \text{ for } n = 1, \ldots, N$$

- N variables, $N + 1$ simple constraints —suitable when N small
- physical meaning: locate **SVs** $(\mathbf{z}_n, y_n)$ & their α_n

both eventually result in optimal $(b, \mathbf{w})$ for fattest hyperplane

$$g_{\text{SVM}}(\mathbf{x}) = \text{sign}(\mathbf{w}^T\boldsymbol{\Phi}(\mathbf{x}) + b)$$

Are We Done Yet?

goal: SVM **without dependence on $\tilde{d}$**

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T \mathrm{Q_D} \boldsymbol{\alpha} - \mathbf{1}^T\boldsymbol{\alpha} \\ \text{subject to} \quad & \mathbf{y}^T\boldsymbol{\alpha} = 0; \\ & \alpha_n \geq 0, \text{for } n = 1, 2, \ldots, N \end{aligned}$$

- N variables, $N+1$ constraints: **no dependence on $\tilde{d}$?**
- $q_{n,m} = y_n y_m \mathbf{z}_n^T \mathbf{z}_m$: inner product in $\mathbb{R}^{\tilde{d}}$ —$O(\tilde{d})$ via naïve computation!

no dependence only if avoiding naïve computation **(next lecture :-))**

Questions?

Summary

1. Embedding Numerous Features: Kernel Models

Lecture 10: Support Vector Machine (1)

- Large-Margin Separating Hyperplane
  intuitively more robust against noise
- Standard Large-Margin Problem
  minimize 'length of $\mathbf{w}$' at special separating scale
- Support Vector Machine
  'easy' via quadratic programming
- Motivation of Dual SVM
  want to remove dependence on $\tilde{d}$
- Lagrange Dual SVM
  KKT conditions link primal/dual
- Solving Dual SVM
  another QP, better solved with special solver
- Messages behind Dual SVM
  SVs represent fattest hyperplane