Lecture 1: Linear Support Vector Machine

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)
Course History

NTU Version

- 15-17 weeks (2+ hours)
- highly-praised with English and blackboard teaching
Course Introduction

**Course History**

**NTU Version**
- 15-17 weeks (2+ hours)
- highly-praised with English and blackboard teaching

![Pie chart showing various fields]

**Coursera Version**
- 8 weeks of ‘foundations’ (previous course) + 8 weeks of ‘techniques’ (this course)
- Mandarin teaching to reach more audience in need
- slides teaching improved with Coursera’s quiz and homework mechanisms
**Course History**

**NTU Version**
- 15-17 weeks (2+ hours)
- highly-praised with **English and blackboard teaching**

**Coursera Version**
- 8 weeks of ‘foundations’ (previous course) + 8 weeks of ‘techniques’ (this course)
- **Mandarin teaching** to reach more audience in need
- **slides teaching** improved with Coursera’s quiz and homework mechanisms

**goal:** try making Coursera version even better than NTU version
Course Design

from Foundations to Techniques

- mixture of philosophical illustrations, key theory, core algorithms, usage in practice, and hopefully jokes :-)

Hsuan-Tien Lin (NTU CSIE)
Course Design

from Foundations to Techniques

- mixture of philosophical illustrations, key theory, core algorithms, usage in practice, and hopefully jokes :-) 
- three major techniques surrounding feature transforms:
Course Design

from Foundations to Techniques

- mixture of philosophical illustrations, key theory, core algorithms, usage in practice, and hopefully jokes :-)
- three major techniques surrounding feature transforms:
  - Embedding Numerous Features: how to exploit and regularize numerous features?
Course Design

from Foundations to Techniques

• mixture of philosophical illustrations, key theory, core algorithms, usage in practice, and hopefully jokes :-)
• three major techniques surrounding feature transforms:
  • Embedding Numerous Features: how to exploit and regularize numerous features?
    —inspires Support Vector Machine (SVM) model
Course Design

from Foundations to Techniques

- mixture of philosophical illustrations, key theory, core algorithms, usage in practice, and hopefully jokes :-) 
- three major techniques surrounding feature transforms:
  - Embedded Numerous Features: how to exploit and regularize numerous features?
    —inspires Support Vector Machine (SVM) model
  - Combining Predictive Features: how to construct and blend predictive features?
Course Design

from Foundations to Techniques

- mixture of philosophical illustrations, key theory, core algorithms, usage in practice, and hopefully jokes :-)
- three major techniques surrounding **feature transforms**:
  - Embedding Numerous Features: how to *exploit* and *regularize* numerous features?
    —inspires **Support Vector Machine** (SVM) model
  - Combining Predictive Features: how to *construct* and *blend* predictive features?
    —inspires **Adaptive Boosting** (AdaBoost) model
Course Design

from Foundations to Techniques

- mixture of philosophical illustrations, key theory, core algorithms, usage in practice, and hopefully jokes :-)  
- three major techniques surrounding feature transforms:
  - Embedding Numerous Features: how to exploit and regularize numerous features?  
    —inspires Support Vector Machine (SVM) model  
  - Combining Predictive Features: how to construct and blend predictive features?  
    —inspires Adaptive Boosting (AdaBoost) model  
  - Distilling Implicit Features: how to identify and learn implicit features?
Course Design

from Foundations to Techniques

- mixture of philosophical illustrations, key theory, core algorithms, usage in practice, and hopefully jokes :-)  
- three major techniques surrounding feature transforms:
  - Embedding Numerous Features: how to exploit and regularize numerous features?  
    —inspires Support Vector Machine (SVM) model  
  - Combining Predictive Features: how to construct and blend predictive features?  
    —inspires Adaptive Boosting (AdaBoost) model  
  - Distilling Implicit Features: how to identify and learn implicit features?  
    —inspires Deep Learning model
Course Design

from Foundations to Techniques

- mixture of philosophical illustrations, key theory, core algorithms, usage in practice, and hopefully jokes :-)  
- three major techniques surrounding feature transforms: 
  - Embedding Numerous Features: how to exploit and regularize numerous features?  
    — inspires Support Vector Machine (SVM) model 
  - Combining Predictive Features: how to construct and blend predictive features?  
    — inspires Adaptive Boosting (AdaBoost) model 
  - Distilling Implicit Features: how to identify and learn implicit features?  
    — inspires Deep Learning model 

allows students to use ML professionally
Fun Time

Which of the following description of this course is true?

1. the course will be taught in Taiwanese
2. the course will tell me the techniques that create the android Lieutenant Commander Data in Star Trek
3. the course will be 16 weeks long
4. the course will focus on three major techniques

Reference Answer:
1. no, my Taiwanese is unfortunately not good enough for teaching (yet)
2. no, although what we teach may serve as building blocks
3. no, unless you have also joined the previous course
4. yes, let's get started!
Fun Time

Which of the following description of this course is true?

1. the course will be taught in Taiwanese
2. the course will tell me the techniques that create the android Lieutenant Commander Data in Star Trek
3. the course will be 16 weeks long
4. the course will focus on three major techniques

Reference Answer: 4

1. no, my Taiwanese is unfortunately not good enough for teaching (yet)
2. no, although what we teach may serve as building blocks
3. no, unless you have also joined the previous course
4. yes, let’s get started!
Roadmap

1. Embedding Numerous Features: Kernel Models
   
   **Lecture 1: Linear Support Vector Machine**
   - Course Introduction
   - Large-Margin Separating Hyperplane
   - Standard Large-Margin Problem
   - Support Vector Machine
   - Reasons behind Large-Margin Hyperplane

2. Combining Predictive Features: Aggregation Models

3. Distilling Implicit Features: Extraction Models
Linear Classification Revisited

PLA/pocket

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

\[ s = \mathbf{w}^T \mathbf{x} \]

plausible error = 0/1
(small flipping noise)
minimize specially

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!

\[
E_{\text{out}}(w) \leq E_{\text{in}}(w) + \Omega(H_{\text{VC}}) = d + 1
\]
Which Line Is Best?

- PLA? depending on randomness

\[ E_{out}(w) \leq E_{in}(w) + \Omega(H) \]

\( VC = d + 1 \)
Which Line Is Best?

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

\[ E_{\text{out}}(w) \leq E_{\text{in}}(w) + \underbrace{\Omega(\mathcal{H})}_{d_{\text{VC}}=d+1} \]
Which Line Is Best?

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

\[
E_{\text{out}}(w) \leq E_{\text{in}}(w) + \begin{cases} 
\Omega(H) \\
0 
\end{cases} \quad d_{\text{VC}} = d+1
\]

You? rightmost one, possibly :-(
informal argument

if (Gaussian-like) noise on future $x \approx x_n$:
informal argument

if (Gaussian-like) noise on future $x \approx x_n$:

$x_n$ further from hyperplane
informal argument

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

$\mathbf{x}_n$ further from hyperplane

$\iff$ tolerate more noise
informal argument

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

$\mathbf{x}_n$ further from hyperplane

$\iff$ tolerate more noise

$\iff$ more robust to overfitting
informal argument

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

- $\mathbf{x}_n$ further from hyperplane

$\iff$ tolerate more noise $\iff$ amount of noise tolerance

$\iff$ more robust to overfitting $\iff$ robustness of hyperplane

Why Rightmost Hyperplane?
informal argument

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

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

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

$\mathbf{x}_n$ further from hyperplane  $\iff$ distance to closest $\mathbf{x}_n$

tolerate more noise  $\iff$ amount of noise tolerance

more robust to overfitting  $\iff$ robustness of hyperplane

rightmost one: more robust
because of larger distance to closest $\mathbf{x}_n$
• **robust** separating hyperplane: **fat**
  —far from both sides of examples
Linear Support Vector Machine

**Fat Hyperplane**

- robust separating hyperplane: **fat**
  — far from both sides of examples
- robustness $\equiv$ **fatness**: distance to closest $x_n$
Fat Hyperplane

- **robust** separating hyperplane: **fat**
  — far from both sides of examples
- **robustness** $\equiv$ **fatness**: distance to closest $x_n$

**goal:** find **fattest** separating hyperplane
Linear Support Vector Machine

Large-Margin Separating Hyperplane

\[ \max_w \ fatness(w) \]

subject to \[ w \text{ classifies every } (x_n, y_n) \text{ correctly} \]

\[ \text{fatness}(w) = \min_{n=1,...,N} \text{distance}(x_n, w) \]
Large-Margin Separating Hyperplane

\[
\begin{align*}
\max_{\mathbf{w}} & \quad \text{margin}(\mathbf{w}) \\
\text{subject to} & \quad \mathbf{w} \text{ classifies every } (\mathbf{x}_n, y_n) \text{ correctly} \\
\text{margin}(\mathbf{w}) & = \min_{n=1,\ldots,N} \text{distance}(\mathbf{x}_n, \mathbf{w})
\end{align*}
\]

- fatness: formally called margin
Large-Margin Separating Hyperplane

\[
\max_w \quad \text{margin}(w) \\
\text{subject to} \quad \text{every } y_n w^T x_n > 0 \\
\text{margin}(w) = \min_{n=1,...,N} \text{distance}(x_n, w)
\]

- fatness: formally called \textit{margin}
- correctness: \(y_n = \text{sign}(w^T x_n)\)
**Large-Margin Separating Hyperplane**

$$\max_w \text{margin}(w)$$

subject to every $y_n w^T x_n > 0$

$$\text{margin}(w) = \min_{n=1, \ldots, N} \text{distance}(x_n, w)$$

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

**goal:** find **largest-margin separating** hyperplane
Consider two examples \((v, +1)\) and \((-v, -1)\) where \(v \in \mathbb{R}^2\) (without padding the \(v_0 = 1\)). Which of the following hyperplane is the largest-margin separating one for the two examples? You are highly encouraged to visualize by considering, for instance, \(v = (3, 2)\).

1. \(x_1 = 0\)
2. \(x_2 = 0\)
3. \(v_1 x_1 + v_2 x_2 = 0\)
4. \(v_2 x_1 + v_1 x_2 = 0\)
Consider two examples \((\mathbf{v}, +1)\) and \((-\mathbf{v}, -1)\) where \(\mathbf{v} \in \mathbb{R}^2\) (without padding the \(v_0 = 1\)). Which of the following hyperplane is the largest-margin separating one for the two examples? You are highly encouraged to visualize by considering, for instance, \(\mathbf{v} = (3, 2)\).

1. \(x_1 = 0\)
2. \(x_2 = 0\)
3. \(v_1 x_1 + v_2 x_2 = 0\)
4. \(v_2 x_1 + v_1 x_2 = 0\)

**Reference Answer:** 3

Here the largest-margin separating hyperplane (line) must be a perpendicular bisector of the line segment between \(\mathbf{v}\) and \(-\mathbf{v}\). Hence \(\mathbf{v}\) is a normal vector of the largest-margin line. The result can be extended to the more general case of \(\mathbf{v} \in \mathbb{R}^d\).
Distance to Hyperplane: Preliminary

\[
\max_w \text{ margin}(w) \\
\text{subject to } y_n w^T x_n > 0 \\
\text{margin}(w) = \min_{n=1,\ldots,N} \text{distance}(x_n, w)
\]
Distance to Hyperplane: Preliminary

\[
\begin{align*}
\max_w & \quad \text{margin}(w) \\
\text{subject to} & \quad \text{every } y_n w^T x_n > 0 \\
\text{margin}(w) & = \min_{n=1,\ldots,N} \text{distance}(x_n, w)
\end{align*}
\]

‘shorten’ \(x\) and \(w\)

distance needs \(w_0\) and \((w_1, \ldots, w_d)\) differently (to be derived)
Distance to Hyperplane: Preliminary

\[
\begin{align*}
\max_w & \quad \text{margin}(w) \\
\text{subject to} & \quad \text{every } y_n w^T x_n > 0 \\
\text{margin}(w) = & \quad \min_{n=1,\ldots,N} \text{distance}(x_n, w)
\end{align*}
\]

‘shorten’ \( x \) and \( w \)

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

\[
\begin{bmatrix}
\vdots \\
\vdots \\
w_d
\end{bmatrix} = \begin{bmatrix}
w_1 \\
\vdots \\
w_d
\end{bmatrix} ; \\
\begin{bmatrix}
x_0 \\
x_1 \\
x_d
\end{bmatrix} = \begin{bmatrix}
\vdots \\
\vdots \\
x_d
\end{bmatrix}
\]
Distance to Hyperplane: Preliminary

\[
\begin{align*}
\text{max}_w & \quad \text{margin}(w) \\
\text{subject to} & \quad \text{every } y_n w^T x_n > 0 \\
\text{margin}(w) & = \min_{n=1,\ldots,N} \text{distance}(x_n, w)
\end{align*}
\]

'shorten' \( x \) and \( w \)

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

\[
\begin{bmatrix}
\vdots \\
w \\
\vdots
\end{bmatrix} = 
\begin{bmatrix}
w_1 \\
\vdots \\
w_d
\end{bmatrix} ;
\begin{bmatrix}
x \\
\vdots
\end{bmatrix} = 
\begin{bmatrix}
x_1 \\
\vdots \\
x_d
\end{bmatrix}
\]

for this part: \( h(x) = \text{sign}(w^T x + b) \)
Distance to Hyperplane

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

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

consider $\mathbf{x}'$, $\mathbf{x}''$ on hyperplane
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 $\mathbf{w}^T \mathbf{x}' = \vdots$,

$\mathbf{w}^T \mathbf{x}'' = \vdots$
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 $\mathbf{w}^T \mathbf{x}' = -b$, $\mathbf{w}^T \mathbf{x''} = -b$
Distance to Hyperplane

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

consider $x'$, $x''$ on hyperplane

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

$$
\begin{pmatrix}
  w^T \\
  (x'' - x') \\
  \text{vector on hyperplane}
\end{pmatrix} = 0
$$
Distance to Hyperplane

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

consider $x', x''$ on hyperplane

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

$$\begin{pmatrix} w^T \\ (x'' - x') \end{pmatrix} = 0$$

(vector on hyperplane)
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:

$$\begin{pmatrix} \mathbf{w}^T \\ \mathbf{(x'' - x')} \end{pmatrix} = 0$$

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

**Standard Large-Margin Problem**

want: $\text{distance}(\mathbf{x}, b, \mathbf{w})$, 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:

$$\begin{pmatrix} \mathbf{w}^T \\ (\mathbf{x}'' - \mathbf{x}') \end{pmatrix} = 0$$

the vector on hyperplane

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

$$\text{distance}(\mathbf{x}, b, \mathbf{w}) = \left| (\mathbf{x} - \mathbf{x}') \right|$$
Distance to Hyperplane

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

counter $x', x''$ on hyperplane

1 $w^T x' = -b$, $w^T x'' = -b$

2 $w \perp$ hyperplane:

$$\begin{pmatrix}
   w^T \\
   (x'' - x') \\
   \text{vector on hyperplane}
\end{pmatrix} = 0$$

3 distance = project $(x - x')$ to $\perp$ hyperplane

$$\text{distance}(x, b, w) = \left| \frac{w^T}{\|w\|}(x - x') \right|$$
Distance to Hyperplane

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

consider $x'$, $x''$ on hyperplane

1. $w^T x' = -b$, $w^T x'' = -b$
2. $w \perp$ hyperplane:
   \[
   \begin{pmatrix}
   w^T \\
   (x'' - x')
   \end{pmatrix} = 0
   \]
   vector on hyperplane
3. distance = project $(x - x')$ to $\perp$ hyperplane

\[
distance(x, b, w) = \left| \frac{w^T}{||w||} (x - x') \right| \equiv \frac{1}{||w||} |w^T x|
\]
Distance to Hyperplane

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

consider \(x', x''\) on hyperplane

1. \(w^T x' = -b, w^T x'' = -b\)
2. \(w \perp\) hyperplane:
   \[
   \begin{pmatrix}
   w^T \\
   (x'' - x')
   \end{pmatrix} = 0
   \]
   (vector on hyperplane)
3. distance = project \((x - x')\) to \(\perp\) hyperplane

\[
\text{distance}(x, b, w) = \left| \frac{w^T}{\|w\|} (x - x') \right| \overset{1}{=} \frac{1}{\|w\|} |w^T x + b|
\]
**Distance to **Separating **Hyperplane**

\[ \text{distance}(x, b, w) = \frac{1}{\|w\|} |w^T x + b| \]
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 \( n \)
  \[
y_n(\mathbf{w}^T \mathbf{x}_n + b) > 0
\]

\[
\max_{b, \mathbf{w}} \quad \text{margin}(b, \mathbf{w})
\]

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

\[
\text{margin}(b, \mathbf{w}) = \min_{n=1,...,N} \text{distance}(\mathbf{x}_n, b, \mathbf{w})
\]
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:
  \[
  \text{distance}(\mathbf{x}_n, b, \mathbf{w}) = \frac{1}{\|\mathbf{w}\|} y_n(\mathbf{w}^T \mathbf{x}_n + b)
  \]

\[
\max_{b, \mathbf{w}} \text{margin}(b, \mathbf{w})
\]
subject to
\[
every \ y_n(\mathbf{w}^T \mathbf{x}_n + b) > 0
\]
\[
\text{margin}(b, \mathbf{w}) = \min_{n=1,...,N} \text{distance}(\mathbf{x}_n, b, \mathbf{w})
\]
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:
  
  \[
  \text{distance}(\mathbf{x}_n, b, \mathbf{w}) = \frac{1}{\|\mathbf{w}\|} y_n(\mathbf{w}^T \mathbf{x}_n + b)
  \]

\[
\max_{\mathbf{b}, \mathbf{w}} \text{margin}(\mathbf{b}, \mathbf{w})
\]

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

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

Hsuan-Tien Lin (NTU CSIE)
Margin of **Special** Separating Hyperplane

\[
\max_{b, w} \text{margin}(b, w) \\
\text{subject to } \forall n \ (y_n(w^T x_n + b) > 0) \\
\text{margin}(b, w) = \min_{n=1, \ldots, N} \frac{1}{\|w\|} y_n(w^T x_n + b)
\]
Margin of **Special** Separating Hyperplane

\[
\begin{align*}
\max_{b, w} \quad & \text{margin}(b, w) \\
\text{subject to} \quad & \text{every } y_n(w^T x_n + b) > 0 \\
\text{margin}(b, w) = \min_{n=1, \ldots, N} \frac{1}{\|w\|} y_n(w^T x_n + b)
\end{align*}
\]

- \(w^T x + b = 0\) same as \(3w^T x + 3b = 0\): scaling does not matter
Margin of **Special** Separating Hyperplane

\[
\begin{align*}
\text{max}_{b, \mathbf{w}} \quad & \text{margin}(b, \mathbf{w}) \\
\text{subject to} \quad & \text{every } y_n(\mathbf{w}^T \mathbf{x}_n + b) > 0 \\
\text{margin}(b, \mathbf{w}) = \min_{n=1, \ldots, N} \frac{1}{\|\mathbf{w}\|} y_n(\mathbf{w}^T \mathbf{x}_n + b)
\end{align*}
\]

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

\[
\min_{n=1, \ldots, N} y_n(\mathbf{w}^T \mathbf{x}_n + b) = 1 \implies
\]
Margin of **Special** Separating Hyperplane

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

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

\[
\min_{n=1, \ldots, N} y_n(\mathbf{w}^T \mathbf{x}_n + b) = 1 \implies \text{margin}(b, \mathbf{w}) = \frac{1}{\|\mathbf{w}\|}
\]
Margin of \textbf{Special} Separating Hyperplane

\[
\begin{align*}
\max_{b,w} & \quad \text{margin}(b, w) \\
\text{subject to} & \quad \text{every } y_n(w^T x_n + b) > 0 \\
\text{margin}(b, w) & = \min_{n=1,\ldots,N} \frac{1}{\|w\|} y_n(w^T x_n + b)
\end{align*}
\]

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

\[
\min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1 \implies \text{margin}(b, w) = \frac{1}{\|w\|}
\]

\[
\begin{align*}
\max_{b,w} & \quad \frac{1}{\|w\|} \\
\text{subject to} & \quad \text{every } y_n(w^T x_n + b) > 0 \\
\min_{n=1,\ldots,N} & \quad y_n(w^T x_n + b) = 1
\end{align*}
\]
Margin of **Special** Separating Hyperplane

\[
\begin{align*}
\text{max}_{b, w} & \quad \text{margin}(b, w) \\
\text{subject to} & \quad \text{every } y_n(w^T x_n + b) > 0 \\
\text{margin}(b, w) & = \min_{n=1, \ldots, N} \frac{1}{\|w\|} y_n(w^T x_n + b)
\end{align*}
\]

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

\[
\min_{n=1, \ldots, N} y_n(w^T x_n + b) = 1 \implies \text{margin}(b, w) = \frac{1}{\|w\|}
\]

\[
\begin{align*}
\text{max}_{b, w} & \quad \frac{1}{\|w\|} \\
\text{subject to} & \quad \text{every } y_n(w^T x_n + b) > 0 \\
\min_{n=1, \ldots, N} & \quad y_n(w^T x_n + b) = 1
\end{align*}
\]
Linear Support Vector Machine

Standard Large-Margin Problem

Standard Large-Margin Hyperplane Problem

\[
\begin{align*}
\max_{b,w} & \quad \frac{1}{\|w\|} \\
\text{subject to} & \quad \min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1
\end{align*}
\]

original constraint: \( \min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1 \)
Linear Support Vector Machine

Standard Large-Margin Problem

Standard Large-Margin Hyperplane Problem

\[
\begin{align*}
\text{max} & \quad \frac{1}{\|w\|} \\
\text{subject to} & \quad \min_{n=1,...,N} \quad y_n(w^T x_n + b) = 1
\end{align*}
\]

necessary constraints: \( y_n(w^T x_n + b) \geq 1 \) for all \( n \)

original constraint: \( \min_{n=1,...,N} \quad y_n(w^T x_n + b) = 1 \)
Standard Large-Margin Hyperplane Problem

\[
\begin{aligned}
\text{max} & \quad \frac{1}{\|w\|} \\
\text{subject to} & \quad \min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1 \\
\end{aligned}
\]

necessary constraints: \( y_n(w^T x_n + b) \geq 1 \) for all \( n \)

original constraint: \( \min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1 \)

want: optimal \((b, w)\) here (inside)
Linear Support Vector Machine

Standard Large-Margin Problem

Standard Large-Margin Hyperplane Problem

\[
\begin{align*}
\max_{b, w} & \quad \frac{1}{\|w\|} \\
\text{subject to} & \quad \min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1
\end{align*}
\]

necessary constraints: \( y_n(w^T x_n + b) \geq 1 \) for all \( n \)

original constraint: \( \min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1 \)

want: optimal \((b, w)\) here (inside)

if optimal \((b, w)\) outside,
Standard Large-Margin Hyperplane Problem

\[
\begin{align*}
\max_{b, w} & \quad \frac{1}{\|w\|} \\
\text{subject to} & \quad \min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1
\end{align*}
\]

necessary constraints: \( y_n(w^T x_n + b) \geq 1 \) for all \( n \)

original constraint: \( \min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1 \)

want: optimal \((b, w)\) here (inside)

if optimal \((b, w)\) outside, e.g. \( y_n(w^T x_n + b) > 1 \) for all \( n \)
Linear Support Vector Machine

Standard Large-Margin Problem

Standard Large-Margin Hyperplane Problem

\[
\max_{b, w} \frac{1}{\|w\|} \quad \text{subject to} \quad \min_{n=1, \ldots, N} y_n(w^T x_n + b) = 1
\]

necessary constraints: \( y_n(w^T x_n + b) \geq 1 \) for all \( n \)

original constraint: \( \min_{n=1, \ldots, N} y_n(w^T x_n + b) = 1 \)

want: optimal \((b, w)\) here (inside)

if optimal \((b, w)\) outside, e.g. \( y_n(w^T x_n + b) > 1.126 \) for all \( n \)
Standard Large-Margin Hyperplane Problem

\[
\max_{b, w} \quad \frac{1}{\|w\|} \quad \text{subject to} \quad \min_{n=1,...,N} \quad y_n(w^T x_n + b) = 1
\]

necessary constraints: \( y_n(w^T x_n + b) \geq 1 \) for all \( n \)

original constraint: \( \min_{n=1,...,N} \quad y_n(w^T x_n + b) = 1 \)

want: optimal \((b, w)\) here (inside)

if optimal \((b, w)\) outside, e.g. \( y_n(w^T x_n + b) > 1.126 \) for all \( n \)
—can scale \((b, w)\) to “more optimal” \((\frac{b}{1.126}, \frac{w}{1.126})\)
Standard Large-Margin Hyperplane Problem

\[
\begin{align*}
\max_{b,w} & \quad \frac{1}{\|w\|} \\
\text{subject to } & \quad \min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1
\end{align*}
\]

necessary constraints: \( y_n(w^T x_n + b) \geq 1 \) for all \( n \)

original constraint: \( \min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1 \)

want: optimal \((b, w)\) here (inside)

if optimal \((b, w)\) outside, e.g. \( y_n(w^T x_n + b) > 1.126 \) for all \( n \)
—can scale \((b, w)\) to “more optimal” \((b, w) = \left(\frac{b}{1.126}, \frac{w}{1.126}\right)\) (contradiction!)
Standard Large-Margin Hyperplane Problem

\[
\max_{b,w} \quad \frac{1}{\|w\|} \\
\text{subject to} \quad \min_{n=1,...,N} \quad y_n(w^T x_n + b) = 1
\]

necessary constraints: \( y_n(w^T x_n + b) \geq 1 \) for all \( n \)

original constraint: \( \min_{n=1,...,N} \quad y_n(w^T x_n + b) = 1 \)

want: optimal \((b, w)\) here (inside)

if optimal \((b, w)\) outside, e.g. \( y_n(w^T x_n + b) > 1.126 \) for all \( n \)
—can scale \((b, w)\) to “more optimal” \((\frac{b}{1.126}, \frac{w}{1.126})\) (contradiction!)

\[
\max_{b,w} \quad \frac{1}{\|w\|} \\
\text{subject to} \quad y_n(w^T x_n + b) \geq 1 \text{ for all } n
\]
Linear Support Vector Machine

Standard Large-Margin Problem

Standard Large-Margin Hyperplane Problem

\[
\begin{align*}
\max_{b, w} & \quad \frac{1}{\|w\|} \\
\text{subject to} & \quad \min_{n=1, \ldots, N} y_n(w^T x_n + b) = 1
\end{align*}
\]

necessary constraints: \(y_n(w^T x_n + b) \geq 1\) for all \(n\)

original constraint: \(\min_{n=1, \ldots, N} y_n(w^T x_n + b) = 1\)

want: optimal \((b, w)\) here (inside)

if optimal \((b, w)\) outside, e.g. \(y_n(w^T x_n + b) > 1.126\) for all \(n\)
—can scale \((b, w)\) to “more optimal” \((\frac{b}{1.126}, \frac{w}{1.126})\) (contradiction!)

final change: \(\max \implies \min\), remove \(\sqrt{\cdot}\)

\[
\begin{align*}
\max_{b, w} & \quad \frac{1}{\|w\|} \\
\text{subject to} & \quad y_n(w^T x_n + b) \geq 1 \text{ for all } n
\end{align*}
\]
Standard Large-Margin Hyperplane Problem

\[
\begin{align*}
\text{max } & \quad \frac{1}{\|w\|} \\
\text{subject to } & \quad \min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1
\end{align*}
\]

necessary constraints: \( y_n(w^T x_n + b) \geq 1 \) for all \( n \)

original constraint: \( \min_{n=1,\ldots,N} y_n(w^T x_n + b) = 1 \)
want: optimal \((b, w)\) here (inside)

if optimal \((b, w)\) outside, e.g. \( y_n(w^T x_n + b) > 1.126 \) for all \( n \)
—can scale \((b, w)\) to “more optimal” \( (\frac{b}{1.126}, \frac{w}{1.126}) \) (contradiction!)

final change: \( \text{max } \Rightarrow \text{min}, \text{ remove } \sqrt{\cdot} \)

\[
\begin{align*}
\text{min } & \quad w^T w \\
\text{subject to } & \quad y_n(w^T x_n + b) \geq 1 \text{ for all } n
\end{align*}
\]
Standard Large-Margin Hyperplane Problem

\[
\begin{align*}
\max_{b, w} \quad & \frac{1}{\|w\|} \\
\text{subject to} \quad & \min_{n=1, \ldots, N} y_n(w^T x_n + b) = 1
\end{align*}
\]

necessary constraints: \( y_n(w^T x_n + b) \geq 1 \) for all \( n \)

original constraint: \( \min_{n=1, \ldots, N} y_n(w^T x_n + b) = 1 \)

want: optimal \( (b, w) \) here (inside)

if optimal \( (b, w) \) outside, e.g. \( y_n(w^T x_n + b) > 1.126 \) for all \( n \)
—can scale \( (b, w) \) to “more optimal” \( \left( \frac{b}{1.126}, \frac{w}{1.126} \right) \) (contradiction!)

final change: \( \max \iff \min \), remove \( \sqrt{\quad} \), add \( \frac{1}{2} \)

\[
\begin{align*}
\min_{b, w} \quad & \frac{1}{2} w^T w \\
\text{subject to} \quad & y_n(w^T x_n + b) \geq 1 \text{ for all } n
\end{align*}
\]
Consider three examples \((x_1, +1), (x_2, +1), (x_3, -1)\), where \(x_1 = (3, 0)\), \(x_2 = (0, 4)\), \(x_3 = (0, 0)\). In addition, consider a hyperplane \(x_1 + x_2 = 1\). Which of the following is not true?

1. the hyperplane is a separating one for the three examples
2. the distance from the hyperplane to \(x_1\) is 2
3. the distance from the hyperplane to \(x_3\) is \(\frac{1}{\sqrt{2}}\)
4. the example that is closest to the hyperplane is \(x_3\)
Consider three examples \((x_1, +1), (x_2, +1), (x_3, -1)\), where \(x_1 = (3, 0), x_2 = (0, 4), x_3 = (0, 0)\). In addition, consider a hyperplane \(x_1 + x_2 = 1\). Which of the following is not true?

1. the hyperplane is a separating one for the three examples
2. the distance from the hyperplane to \(x_1\) is 2
3. the distance from the hyperplane to \(x_3\) is \(\frac{1}{\sqrt{2}}\)
4. the example that is closest to the hyperplane is \(x_3\)

**Reference Answer:** 2

The distance from the hyperplane to \(x_1\) is

\[
\frac{1}{\sqrt{2}}(3 + 0 - 1) = \sqrt{2}.
\]
Solving a Particular Standard Problem

\[ \text{min} \quad \frac{1}{2} w^T w \]

subject to \[ y_n (w^T x_n + b) \geq 1 \text{ for all } n \]
Solving a Particular Standard Problem

\[ \begin{aligned} \min_{b,w} & \quad \frac{1}{2} w^T w \\ \text{subject to} & \quad y_n (w^T x_n + b) \geq 1 \text{ for all } n \end{aligned} \]

\[ \begin{aligned} X = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad & \quad y = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix} \end{aligned} \]

\[ \begin{aligned} -b \geq 1 & \quad (i) \\ -2w_1 - 2w_2 - b \geq 1 & \quad (ii) \\ 2w_1 + b \geq 1 & \quad (iii) \\ 3w_1 + b \geq 1 & \quad (iv) \end{aligned} \]
Linear Support Vector Machine

Support Vector Machine

Solving a Particular Standard Problem

\[
\min_{b, w} \frac{1}{2} w^T w \\
\text{subject to } y_n(w^T x_n + b) \geq 1 \text{ for all } n
\]

\[
X = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix}, \quad y = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}
\]

\[
-2w_1 - 2w_2 - b \geq 1 \quad (i) \\
2w_1 + b \geq 1 \quad (ii) \\
3w_1 + b \geq 1 \quad (iii) \\
- b \geq 1
\]

\[
\left\{ (i) \quad \& \quad (iii) \right\} \implies w_1 \geq +1
\]
Solving a Particular Standard Problem

\[
\begin{align*}
\min_{b,w} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_n(w^T x_n + b) \geq 1 \text{ for all } n
\end{align*}
\]

\[
X = \begin{bmatrix}
0 & 0 \\
2 & 2 \\
2 & 0 \\
3 & 0 \\
\end{bmatrix} \quad y = \begin{bmatrix}
-1 \\
-1 \\
+1 \\
+1 \\
\end{bmatrix} \quad \begin{align*}
-2w_1 - 2w_2 - b & \geq 1 \quad (i) \\
2w_1 & + b \geq 1 \quad (ii) \\
3w_1 & + b \geq 1 \quad (iii) \\
\end{align*}
\]

\[
\begin{align*}
\{ (i) \& (iii) \implies w_1 & \geq +1 \\
(ii) \& (iii) \implies w_2 & \leq -1 \}\implies &
\end{align*}
\]
Solving a Particular Standard Problem

\[
\begin{align*}
\text{min} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_n(w^T x_n + b) \geq 1 \text{ for all } n
\end{align*}
\]

\[
X = \begin{bmatrix}
0 & 0 \\
2 & 2 \\
2 & 0 \\
3 & 0
\end{bmatrix} \quad y = \begin{bmatrix}
-1 \\
-1 \\
+1 \\
+1
\end{bmatrix}
\]

\[
\begin{align*}
-2w_1 - 2w_2 - b & \geq 1 \quad (i) \\
2w_1 + b & \geq 1 \quad (ii) \\
3w_1 + b & \geq 1 \quad (iii) \\
- b & \geq 1 \quad (iv)
\end{align*}
\]

\[
\begin{align*}
\{ (i) \text{ } & \& \text{ } (iii) \Rightarrow w_1 \geq +1 \\
(ii) \text{ } & \& \text{ } (iii) \Rightarrow w_2 \leq -1 \}
\end{align*}
\]

\[
\frac{1}{2} w^T w \geq 1
\]
Solving a Particular Standard Problem

\[
\begin{align*}
\min_{b,\mathbf{w}} & \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} \\
\text{subject to} & \quad y_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1 \text{ for all } n
\end{align*}
\]

\[
\begin{array}{cccc}
X = \begin{bmatrix}
0 & 0 \\
2 & 2 \\
2 & 0 \\
3 & 0 \\
\end{bmatrix} & \quad y = \begin{bmatrix}
-1 \\
-1 \\
+1 \\
+1 \\
\end{bmatrix} & \quad \begin{align*}
-2w_1 - 2w_2 & \geq -b \geq 1 \quad (i) \\
2w_1 & \geq +b \geq 1 \quad (ii) \\
3w_1 & \geq +b \geq 1 \quad (iii) \\
\end{align*}
\end{array}
\]

\[ \begin{align*}
\{ (i) & \quad \text{&} \quad (iii) \quad \Rightarrow \quad w_1 \geq +1 \} \quad \Rightarrow \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} \geq 1 \\
\{ (ii) & \quad \text{&} \quad (iii) \quad \Rightarrow \quad w_2 \leq -1 \} \quad \Rightarrow \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} \geq 1
\end{align*} \]

\[ \text{• } (w_1 = 1, w_2 = -1, b = -1) \text{ at lower bound and satisfies } (i) - (iv) \]
Linear Support Vector Machine

Support Vector Machine

Solving a Particular Standard Problem

\[
\begin{align*}
\text{min} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_n (w^T x_n + b) \geq 1 \text{ for all } n
\end{align*}
\]

\[
X = \begin{bmatrix}
0 & 0 \\
2 & 2 \\
2 & 0 \\
3 & 0
\end{bmatrix}, \quad
y = \begin{bmatrix}
-1 \\
-1 \\
+1 \\
+1
\end{bmatrix}
\]

\[
\begin{align*}
-2w_1 - 2w_2 - b & \geq 1 \quad (i) \\
2w_1 & + b \geq 1 \quad (ii) \\
3w_1 & + b \geq 1 \quad (iii) \\
\end{align*}
\]

\[
\begin{align*}
\begin{cases}
(i) & \& (iii) \implies w_1 \geq +1 \\
(ii) & \& (iii) \implies w_2 \leq -1
\end{cases} & \implies \frac{1}{2} w^T w \geq 1
\end{align*}
\]

\[
\begin{align*}
(w_1 = 1, w_2 = -1, b = -1) \text{ at lower bound and satisfies } (i) \text{ – } (iv)
\end{align*}
\]

\[
g_{\text{SVM}}(x) = \text{sign}(x_1 - x_2 - 1): 
\]
Solving a Particular Standard Problem

\[
\begin{align*}
\min_{b,w} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_n (w^T x_n + b) \geq 1 \text{ for all } n
\end{align*}
\]

\[
X = \begin{bmatrix} 0 & 0 \\ 2 & 2 \\ 2 & 0 \\ 3 & 0 \end{bmatrix} \quad y = \begin{bmatrix} -1 \\ -1 \\ +1 \\ +1 \end{bmatrix}
\]

\[
\begin{align*}
-2w_1 - 2w_2 - b & \geq 1 \quad (i) \\
2w_1 + b & \geq 1 \quad (ii) \\
3w_1 + b & \geq 1 \quad (iii) \\
- b & \geq 1 \quad (iv)
\end{align*}
\]

\[
\begin{align*}
\{ (i) & \land (iii) \Rightarrow w_1 \geq +1 \\
(ii) & \land (iii) \Rightarrow w_2 \leq -1 \} \Rightarrow \frac{1}{2} w^T w \geq 1
\end{align*}
\]

\[
\begin{align*}
(w_1 = 1, w_2 = -1, b = -1) \text{ at lower bound and satisfies (i) – (iv)}
\end{align*}
\]

\[
g_{\text{SVM}}(x) = \text{sign}(x_1 - x_2 - 1): \text{ SVM? :-)}
\]
optimal solution: \((w_1 = 1, \ w_2 = -1, \ b = -1)\)

\[
\text{margin}(b, w) = \frac{1}{\|w\|} = \frac{1}{\sqrt{2}}
\]
Support Vector Machine (SVM)

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

\[
\text{margin}(b, w) = \frac{1}{\|w\|} = \frac{1}{\sqrt{2}}
\]

• examples on boundary: ‘locates’ fattest hyperplane
  other examples: not needed
optimal solution: \((w_1 = 1, w_2 = -1, b = -1)\)

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

• examples on boundary: ‘locates’ fattest hyperplane
  other examples: not needed
• call boundary example support vector (candidate)
Support Vector Machine (SVM)

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

margin \((b, w)\)  
\[ = \frac{1}{\|w\|} = \frac{1}{\sqrt{2}} \]

- 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

\[
\begin{align*}
\min_{b,w} & \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} \\
\text{subject to} & \quad y_n (\mathbf{w}^T \mathbf{x}_n + b) \geq 1 \text{ for all } n
\end{align*}
\]
Solving General SVM

\[
\min_{b,w} \quad \frac{1}{2} w^T w \\
\text{subject to} \quad y_n(w^T x_n + b) \geq 1 \text{ for all } n
\]

- not easy manually, of course :-)
- gradient descent? not easy with constraints
Solving General SVM

\[
\begin{align*}
\min_{b, w} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_n(w^T x_n + b) \geq 1 \text{ for all } n
\end{align*}
\]

- not easy manually, of course :-)
- gradient descent? not easy with constraints
- luckily:
  - (convex) quadratic objective function of \((b, w)\)
\[
\begin{align*}
\min_{b,\mathbf{w}} & \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} \\
\text{subject to} & \quad y_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1 \text{ for all } n
\end{align*}
\]

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

\[
\begin{align*}
\min_{b,w} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_n (w^T x_n + b) \geq 1 \text{ for all } n
\end{align*}
\]

- not easy manually, of course :-)
- gradient descent? not easy with constraints
- luckily:
  - (convex) quadratic objective function of \((b, w)\)
  - linear constraints of \((b, w)\)
  —quadratic programming
Solving General SVM

\[
\begin{align*}
\min_{b, w} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_n (w^T x_n + b) \geq 1 \text{ for all } n
\end{align*}
\]

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

— quadratic programming

quadratic programming (QP): 
‘easy’ optimization problem
optimal \((b, w) = ?\)

\[
\min_{b, w} \quad \frac{1}{2} w^T w
\]

subject to
\[
y_n(w^T x_n + b) \geq 1, \quad \text{for } n = 1, 2, \ldots, N
\]
optimal \((b, w) = ?\)

\[
\begin{aligned}
\min_{b, w} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_n(w^T x_n + b) \geq 1, \\
& \text{for } n = 1, 2, \ldots, N
\end{aligned}
\]

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

\[
\begin{aligned}
\min_u & \quad \frac{1}{2} u^T Qu + p^T u \\
\text{subject to} & \quad a_m^T u \geq c_m, \\
& \text{for } m = 1, 2, \ldots, M
\end{aligned}
\]
Linear Support Vector Machine

Support Vector Machine

Quadratic Programming

optimal \((b, w) = ?\)  

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_n(w^T x_n + b) \geq 1, \\
& \quad \text{for } n = 1, 2, \ldots, N
\end{align*}
\]

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

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} u^T Qu + p^T u \\
\text{subject to} & \quad a_m^T u \geq c_m, \\
& \quad \text{for } m = 1, 2, \ldots, M
\end{align*}
\]

objective function: \(u = \)  

\(; Q = \)  

\(; p = \)

constraints: \(a_n^T = \)  

\(; c_n = \)  

\(; M = \)
optimal \((b, w) = ?\)

\[
\min_{b, w} \quad \frac{1}{2} w^T w \\
\text{subject to} \quad y_n(w^T x_n + b) \geq 1, \\
\text{for } n = 1, 2, \ldots, N
\]

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

\[
\min_u \quad \frac{1}{2} u^T Qu + p^T u \\
\text{subject to} \quad a_m^T u \geq c_m, \\
\text{for } m = 1, 2, \ldots, M
\]

objective function:

\[
u = \begin{bmatrix} b \\ w \end{bmatrix}; \quad Q = \begin{bmatrix} 0 & 0^T_d \\ 0_d & I_d \end{bmatrix}; \quad p = 0_{d+1}
\]

constraints:

\[
a_n^T = y_n \begin{bmatrix} 1 \\ x_n^T \end{bmatrix}; \quad c_n = 1; \quad M = N
\]
Linear Support Vector Machine

Support Vector Machine

Quadratic Programming

optimal \((b, w) = ?\)

\[
\begin{align*}
\min_{b, w} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_n(w^T x_n + b) \geq 1, \\
& \quad \text{for } n = 1, 2, \ldots, N
\end{align*}
\]

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

\[
\begin{align*}
\min_u & \quad \frac{1}{2} u^T Qu + p^T u \\
\text{subject to} & \quad a_m^T u \geq c_m, \\
& \quad \text{for } m = 1, 2, \ldots, M
\end{align*}
\]

objective function:

\[
u = \begin{bmatrix} b \\ w \end{bmatrix}; \quad Q = \begin{bmatrix} 0 & 0^T_d \\ 0_d & I_d \end{bmatrix}; \quad p = 0_{d+1}
\]

constraints:

\[
a_n^T = y_n \begin{bmatrix} 1 \\ x_n^T \end{bmatrix}; \quad c_n = 1; \quad M = N
\]

SVM with general QP solver:

easy if you’ve read the manual :-)
Linear Hard-Margin SVM Algorithm

1. \( Q = \begin{bmatrix} 0 & 0^T_d \\ 0_d & I_d \end{bmatrix} \); \( p = 0_{d+1} \); \( a_n^T = y_n \begin{bmatrix} 1 \\ x_n^T \end{bmatrix} \); \( c_n = 1 \)
Linear Hard-Margin SVM Algorithm

1. \[ Q = \begin{bmatrix} 0 & 0^T_d \\ 0_d & I_d \end{bmatrix}; \quad p = 0_{d+1}; \quad a_n^T = y_n \begin{bmatrix} 1 & x_n^T \end{bmatrix}; \quad c_n = 1 \]

2. \[ \begin{bmatrix} b \\ w \end{bmatrix} \leftarrow \text{QP}(Q, p, A, c) \]
Linear Hard-Margin SVM Algorithm

1. \( Q = \begin{bmatrix} 0 & 0^T \\ 0_d & I_d \end{bmatrix} \); \( p = 0_{d+1} \); \( a_n^T = y_n \begin{bmatrix} 1 & x_n^T \end{bmatrix} \); \( c_n = 1 \)

2. \( \begin{bmatrix} b \\ w \end{bmatrix} \leftarrow \text{QP}(Q, p, A, c) \)

3. return \( b \) & \( w \) as \( g_{\text{SVM}} \)
Linear Support Vector Machine

SVM with QP Solver

Linear Hard-Margin SVM Algorithm

1. \( Q = \begin{bmatrix} 0 & 0^T_d \\ 0_d & I_d \end{bmatrix}; \ p = 0_{d+1}; \ a_n^T = y_n \begin{bmatrix} 1 & x_n^T \end{bmatrix}; \ c_n = 1 \)

2. \( \begin{bmatrix} b \\ w \end{bmatrix} \leftarrow \text{QP}(Q, p, A, c) \)

3. return \( b \) & \( w \) as \( g_{\text{SVM}} \)

- **hard-margin**: nothing violate ‘fat boundary’
- **linear**: \( x_n \)
Linear Support Vector Machine

SVM with QP Solver

Linear Hard-Margin SVM Algorithm

1. \[ Q = \begin{bmatrix} 0 & 0^T_d \\ 0_d & I_d \end{bmatrix}; \quad p = 0_{d+1}; \quad a_n^T = y_n \begin{bmatrix} 1 \\ x_n^T \end{bmatrix}; \quad c_n = 1 \]

2. \[ \begin{bmatrix} b \\ w \end{bmatrix} \leftarrow \text{QP}(Q, p, A, c) \]

3. return \( b \) & \( w \) as \( g_{\text{SVM}} \)

- **hard-margin**: nothing violate ‘fat boundary’
- **linear**: \( x_n \)

want non-linear?

\[ z_n = \Phi(x_n) \text{—remember? :-) } \]
Consider two negative examples with \( x_1 = (0, 0) \) and \( x_2 = (2, 2) \); two positive examples with \( x_3 = (2, 0) \) and \( x_4 = (3, 0) \), as shown on page 17 of the slides. Define \( u, Q, p, c_n \) as those listed on page 20 of the slides. What are \( a_n^T \) that need to be fed into the QP solver?

1. \( a_1^T = [-1, 0, 0] \), \( a_2^T = [-1, 2, 2] \), \( a_3^T = [-1, 2, 0] \), \( a_4^T = [-1, 3, 0] \)
2. \( a_1^T = [1, 0, 0] \), \( a_2^T = [1, -2, -2] \), \( a_3^T = [-1, 2, 0] \), \( a_4^T = [-1, 3, 0] \)
3. \( a_1^T = [1, 0, 0] \), \( a_2^T = [1, 2, 2] \), \( a_3^T = [1, 2, 0] \), \( a_4^T = [1, 3, 0] \)
4. \( a_1^T = [-1, 0, 0] \), \( a_2^T = [-1, -2, -2] \), \( a_3^T = [1, 2, 0] \), \( a_4^T = [1, 3, 0] \)
Consider two negative examples with \( \mathbf{x}_1 = (0, 0) \) and \( \mathbf{x}_2 = (2, 2) \); two positive examples with \( \mathbf{x}_3 = (2, 0) \) and \( \mathbf{x}_4 = (3, 0) \), as shown on page 17 of the slides. Define \( \mathbf{u}, \mathbf{Q}, \mathbf{p}, c_n \) as those listed on page 20 of the slides. What are \( \mathbf{a}_n^T \) that need to be fed into the QP solver?

1. \( \mathbf{a}_1^T = [-1, 0, 0] \), \( \mathbf{a}_2^T = [-1, 2, 2] \), \( \mathbf{a}_3^T = [-1, 2, 0] \), \( \mathbf{a}_4^T = [-1, 3, 0] \)
2. \( \mathbf{a}_1^T = [1, 0, 0] \), \( \mathbf{a}_2^T = [1, -2, -2] \), \( \mathbf{a}_3^T = [-1, 2, 0] \), \( \mathbf{a}_4^T = [-1, 3, 0] \)
3. \( \mathbf{a}_1^T = [1, 0, 0] \), \( \mathbf{a}_2^T = [1, 2, 2] \), \( \mathbf{a}_3^T = [1, 2, 0] \), \( \mathbf{a}_4^T = [1, 3, 0] \)
4. \( \mathbf{a}_1^T = [-1, 0, 0] \), \( \mathbf{a}_2^T = [-1, -2, -2] \), \( \mathbf{a}_3^T = [1, 2, 0] \), \( \mathbf{a}_4^T = [1, 3, 0] \)

Reference Answer: 4

We need \( \mathbf{a}_n^T = \mathbf{y}_n [1 \ \mathbf{x}_n^T] \).
Why Large-Margin Hyperplane?

\[
\begin{align*}
\min_{b, w} & \quad \frac{1}{2} w^T w \\
\text{subject to} & \quad y_n(w^T z_n + b) \geq 1 \text{ for all } n
\end{align*}
\]
Why Large-Margin Hyperplane?

\[
\begin{align*}
\min_{b, \mathbf{w}} & \quad \frac{1}{2} \mathbf{w}^T \mathbf{w} \\
\text{subject to} & \quad y_n(\mathbf{w}^T \mathbf{z}_n + b) \geq 1 \text{ for all } n
\end{align*}
\]

<table>
<thead>
<tr>
<th>regularization</th>
<th>minimize</th>
<th>constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>( E_{\text{in}} )</td>
<td>( \mathbf{w}^T \mathbf{w} \leq C )</td>
</tr>
<tr>
<td></td>
<td>( \mathbf{w}^T \mathbf{w} )</td>
<td>( E_{\text{in}} = 0 ) [and more]</td>
</tr>
</tbody>
</table>
Why Large-Margin Hyperplane?

\[
\min_{b,w} \frac{1}{2} w^T w
\]
subject to \( y_n(w^T z_n + b) \geq 1 \) for all \( n \)

---

<table>
<thead>
<tr>
<th>Regularization</th>
<th>Minimize ( E_{in} )</th>
<th>Constraint ( w^T w \leq C )</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>( w^T w )</td>
<td>( E_{in} = 0 ) [and more]</td>
</tr>
</tbody>
</table>

SVM (large-margin hyperplane):
‘weight-decay regularization’ within \( E_{in} = 0 \)
Large-Margin Restricts Dichotomies

Consider ‘large-margin algorithm’ \( A_\rho \):
- either returns \( g \) with \( \text{margin}(g) \geq \rho \) (if exists), or 0 otherwise.
Large-Margin Restricts Dichotomies

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

$\mathcal{A}_0$: like PLA $\iff$ shatter ‘general’ 3 inputs
consider ‘large-margin algorithm’ $A_\rho$:

either \textbf{returns $g$ with margin}(g) \geq \rho \text{ (if exists)}, \text{ or 0 otherwise}

$A_0$: like PLA $\iff$ shatter ‘general’ 3 inputs

$A_{1.126}$: more strict than SVM $\iff$ cannot shatter any 3 inputs
Large-Margin Restricts Dichotomies

Consider ‘large-margin algorithm’ $A_\rho$:

- either returns $g$ with margin($g$) $\geq \rho$ (if exists), or 0 otherwise

$A_0$: like PLA $\iff$ shatter ‘general’ 3 inputs

$A_{1.126}$: more strict than SVM $\iff$ cannot shatter any 3 inputs

fewer dichotomies $\iff$ smaller ‘VC dim.’ $\iff$ better generalization
VC Dimension of Large-Margin Algorithm

fewer dichotomies $\implies$ smaller ‘VC dim.’
VC Dimension of Large-Margin Algorithm

fewer dichotomies \(\implies\) smaller ‘VC dim.’

considers \(d_{VC}(\mathcal{A}_\rho)\) [data-dependent, need more than VC]
VC Dimension of Large-Margin Algorithm

fewer dichotomies $\implies$ smaller ‘VC dim.’

considers $d_{VC}(A_\rho)$ [data-dependent, need more than VC]

instead of $d_{VC}(\mathcal{H})$ [data-independent, covered by VC]
VC Dimension of Large-Margin Algorithm

fewer dichotomies $\implies$ smaller ‘VC dim.’

- considers $d_{VC}(A_\rho)$ [data-dependent, need more than VC]
  instead of $d_{VC}(\mathcal{H})$ [data-independent, covered by VC]

$d_{VC}(A_\rho)$ when $\mathcal{X}$ = unit circle in $\mathbb{R}^2$

- $\rho = 0$: just perceptrons ($d_{VC} = 3$)
VC Dimension of Large-Margin Algorithm

fewer dichotomies $\implies$ smaller ‘VC dim.’
considers $d_{VC}(A_\rho)$ [data-dependent, need more than VC]
instead of $d_{VC}(\mathcal{H})$ [data-independent, covered by VC]

$d_{VC}(A_\rho)$ when $\mathcal{X}$ = unit circle in $\mathbb{R}^2$

- $\rho = 0$: just perceptrons ($d_{VC} = 3$)
- $\rho > \sqrt{3}/2$: cannot shatter any 3 inputs ($d_{VC} < 3$)
  ---some inputs must be of distance $\leq \sqrt{3}$
Linear Support Vector Machine

Reasons behind Large-Margin Hyperplane

VC Dimension of Large-Margin Algorithm

fewer dichotomies \(\implies\) smaller ‘VC dim.’

considers \(d_{VC}(A_\rho)\) [data-dependent, need more than VC]

instead of \(d_{VC}(\mathcal{H})\) [data-independent, covered by VC]

d_{VC}(A_\rho) when \(\mathcal{X} = \) unit circle in \(\mathbb{R}^2\)

- \(\rho = 0\): just perceptrons (\(d_{VC} = 3\))
- \(\rho > \frac{\sqrt{3}}{2}\): cannot shatter any 3 inputs (\(d_{VC} < 3\))
  —some inputs must be of distance \(\leq \sqrt{3}\)

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

\[
d_{VC}(A_\rho) \leq \min \left( \frac{R^2}{\rho^2}, d \right) + 1 \leq d + 1
\]

\(d_{VC}\) (perceptrons)
### Benefits of Large-Margin Hyperplanes

<table>
<thead>
<tr>
<th></th>
<th>large-margin hyperplanes</th>
<th>hyperplanes</th>
<th>hyperplanes + feature transform $\Phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>even fewer</td>
<td><strong>not many</strong></td>
<td><strong>many</strong></td>
</tr>
<tr>
<td>boundary</td>
<td>simple</td>
<td>simple</td>
<td><strong>sophisticated</strong></td>
</tr>
</tbody>
</table>

- Linear Support Vector Machine
- Reasons behind Large-Margin Hyperplane
## Benefits of Large-Margin Hyperplanes

<table>
<thead>
<tr>
<th></th>
<th>large-margin hyperplanes</th>
<th>hyperplanes</th>
<th>hyperplanes + feature transform $\Phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>even fewer</td>
<td>not many</td>
<td>many</td>
</tr>
<tr>
<td>boundary</td>
<td>simple</td>
<td>simple</td>
<td>sophisticated</td>
</tr>
</tbody>
</table>

- **not many** good, for $d_{vc}$ and generalization
## Benefits of Large-Margin Hyperplanes

<table>
<thead>
<tr>
<th></th>
<th>large-margin hyperplanes</th>
<th>hyperplanes</th>
<th>hyperplanes + feature transform $\Phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>even fewer</td>
<td>not many</td>
<td>many</td>
</tr>
<tr>
<td>boundary</td>
<td>simple</td>
<td>simple</td>
<td>sophisticated</td>
</tr>
</tbody>
</table>

- **not many** good, for $d_{vc}$ and generalization
- **sophisticated** good, for possibly better $E_{in}$
## Benefits of Large-Margin Hyperplanes

<table>
<thead>
<tr>
<th></th>
<th>Large-Margin Hyperplanes</th>
<th>Hyperplanes</th>
<th>Hyperplanes + Feature Transform $\Phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td># boundary</td>
<td>even fewer</td>
<td>not many</td>
<td>many</td>
</tr>
<tr>
<td></td>
<td>simple</td>
<td>simple</td>
<td>sophisticated</td>
</tr>
</tbody>
</table>

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

A new possibility: non-linear SVM

<table>
<thead>
<tr>
<th></th>
<th>Large-Margin Hyperplanes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+ Numerous Feature Transform $\Phi$</td>
</tr>
<tr>
<td># boundary</td>
<td>not many</td>
</tr>
<tr>
<td></td>
<td>sophisticated</td>
</tr>
</tbody>
</table>
Consider running the ‘large-margin algorithm’ \( \mathcal{A}_\rho \) with \( \rho = \frac{1}{4} \) on a \( \mathcal{Z} \)-space such that \( \mathbf{z} = \Phi(\mathbf{x}) \) is of 1126 dimensions (excluding \( z_0 \)) and \( \| \mathbf{z} \| \leq 1 \). What is the upper bound of \( d_{\text{VC}}(\mathcal{A}_\rho) \) when calculated by 
\[
\min \left( \frac{R^2}{\rho^2}, d \right) + 1
\]

1. 5
2. 17
3. 1126
4. 1127
Consider running the ‘large-margin algorithm’ $\mathcal{A}_\rho$ with $\rho = \frac{1}{4}$ on a $\mathcal{Z}$-space such that $z = \Phi(x)$ is of 1126 dimensions (excluding $z_0$) and $\|z\| \leq 1$. What is the upper bound of $d_{vc}(\mathcal{A}_\rho)$ when calculated by $\min\left(\frac{R^2}{\rho^2}, d\right) + 1$?

1. 5
2. 17
3. 1126
4. 1127

Reference Answer: 2

By the description, $d = 1126$ and $R = 1$. So the upper bound is simply 17.
Summary

1 Embedding Numerous Features: Kernel Models

Lecture 1: Linear Support Vector Machine

- Course Introduction
  from foundations to techniques
- Large-Margin Separating Hyperplane
  intuitively more robust against noise
- Standard Large-Margin Problem
  minimize ‘length of w’ at special separating scale
- Support Vector Machine
  ‘easy’ via quadratic programming
- Reasons behind Large-Margin Hyperplane
  fewer dichotomies and better generalization

• next: solving non-linear Support Vector Machine

2 Combining Predictive Features: Aggregation Models

3 Distilling Implicit Features: Extraction Models