SVM

Hendrik Santoso Sugiarto

IBDA2032 – *Artificial Intelligence*

CALVIN
INSTITUTE OF TECHNOLOGY
God's People for God's Glory

Capaian Pembelajaran

- Linear SVM
- Nonlinear SVM
- Multi-class SVM
- Regresi SVM

Linear SVM

Support Vector Machine (SVM)

- Paradigma machine learning yang menggunakan perspektif pemisahan data (jarak antar data)
- Dapat digunakan untuk regresi, klasifikasi, unsupervised
- Cocok untuk klasifikasi data dengan banyak fitur (high dimension) dan jumlah data kecil / menengah (kurang efisien untuk data besar)
- Non-probabilistic: tidak memiliki nilai peluang

Formulasi

- Setting

  $x_1, x_2, \dots$

  - Data latih: $\left((\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \dots, (\boldsymbol{x}^{(m)}, y^{(m)})\right)$
  - Untuk klasifikasi biner: $y^{(i)} \in \{+1, -1\}$
- Tujuan

  $w_1, w_2, \dots$

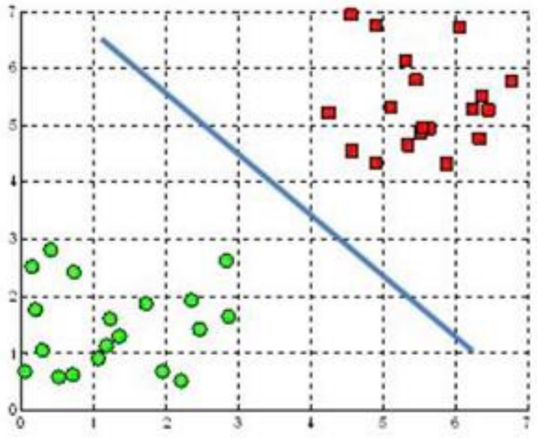
  - Menemukan hyperplane optimal yang dapat memisahkan data: $f(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{w}^T \boldsymbol{x} + b)$
  - Hyperplane: subspace dengan dimensi $n - 1$ dari dimensi data

$$0 = w_1 x_1 + w_2 x_2 + b$$

$$\frac{-w_1 x_1 - b}{w_2} = x_2$$

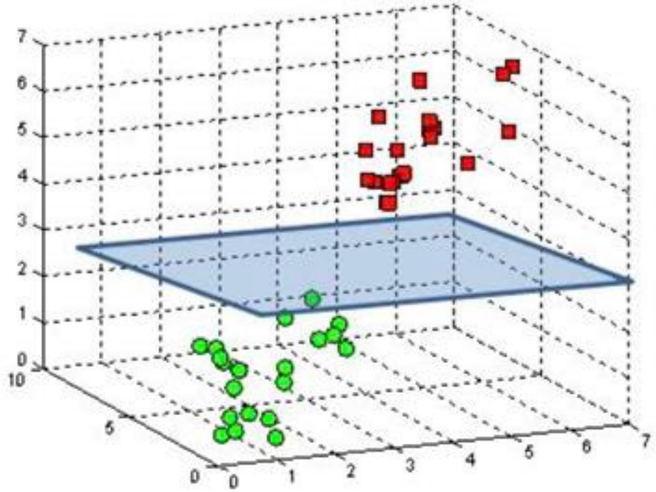
A hyperplane in $\mathbb{R}^2$ is a line

x_2



x_1

A hyperplane in $\mathbb{R}^3$ is a plane



CALVIN INSTITUTE OF TECHNOLOGY

Uji Pemahaman

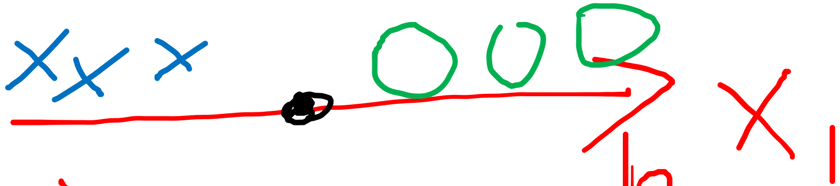
- Berapa dimensi hyperplane dari data dengan fitur berdimensi 1?
  $$\left(\boldsymbol{x}^{(i)} = \left\{x_1^{(i)}\right\}\right)$$ 0
- Berapa dimensi hyperplane dari data dengan fitur berdimensi 2?
  $$\left(\boldsymbol{x}^{(i)} = \left\{x_1^{(i)}, x_2^{(i)}\right\}\right)$$ 1
- Berapa dimensi hyperplane dari data dengan fitur berdimensi 3?
  $$\left(\boldsymbol{x}^{(i)} = \left\{x_1^{(i)}, x_2^{(i)}, x_3^{(i)}\right\}\right)$$ 2
- Berapa dimensi hyperplane dari data dengan fitur berdimensi 5?
  $$\left(\boldsymbol{x}^{(i)} = \left\{x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)}, x_5^{(i)}\right\}\right)$$ 4

Uji Pemahaman

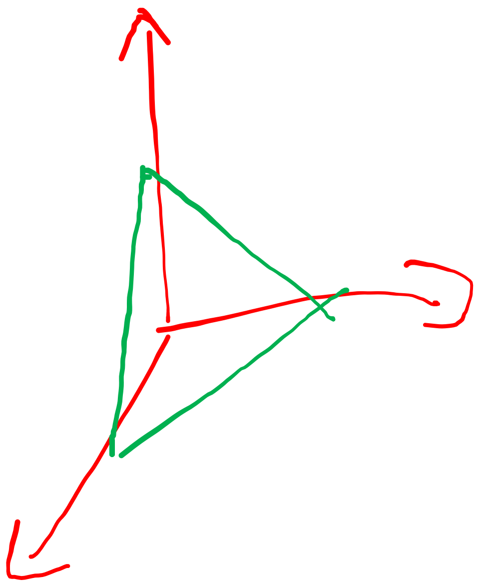
- Bagaimana bentuk persamaan hyperplane kasus sebelumnya?

$$0 = w^T x + b$$

1D :



$$w_1 x_1 + b = 0 \rightarrow x_1 = -\frac{b}{w_1}$$

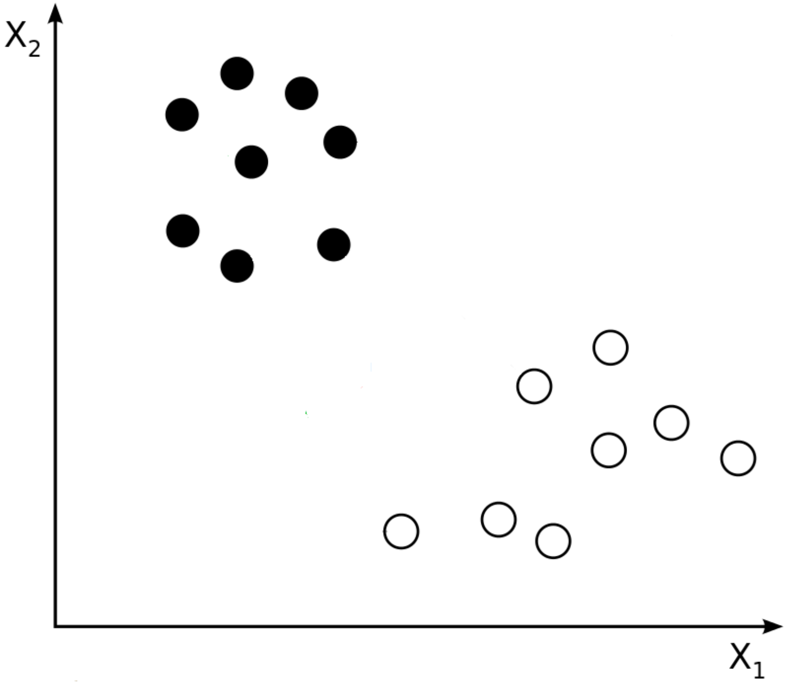


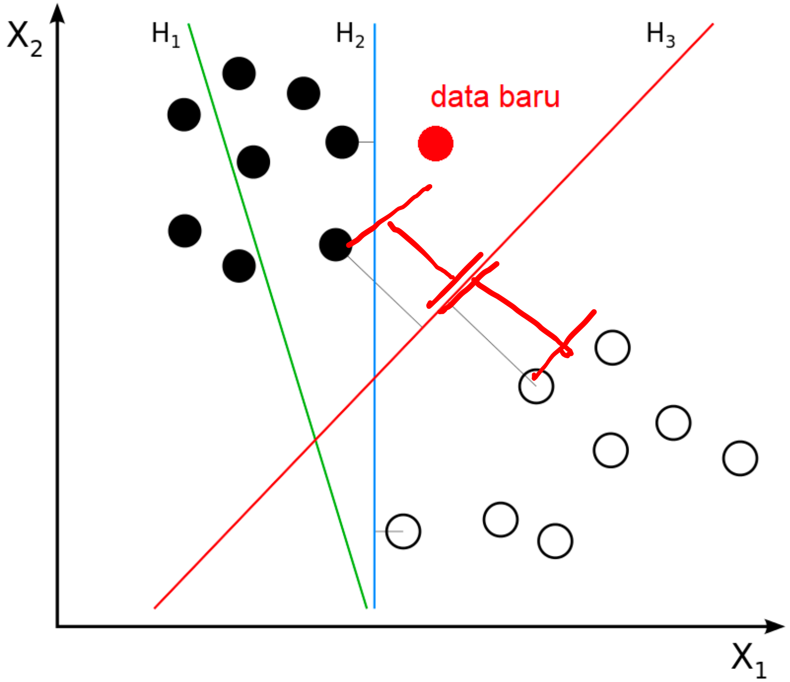
2D : $$x_1 = -\frac{w_2}{w_1} x_2 - \frac{b}{w_1}$$

3D : $$-\frac{b}{w_1} - \frac{w_2}{w_1} x_2 - \frac{w_3}{w_1} x_3 = x_1$$

SVM Linear

- Bagaimana menarik garis lurus yang memisahkan 2 kelas berikut?
- H1 (hijau)→tidak dapat digunakan untuk klasifikasi
- H2 (biru)→mampu klasifikasi, garis pemisah dekat data
- H3 (merah)→mampu klasifikasi, garis pemisah jauh dari data

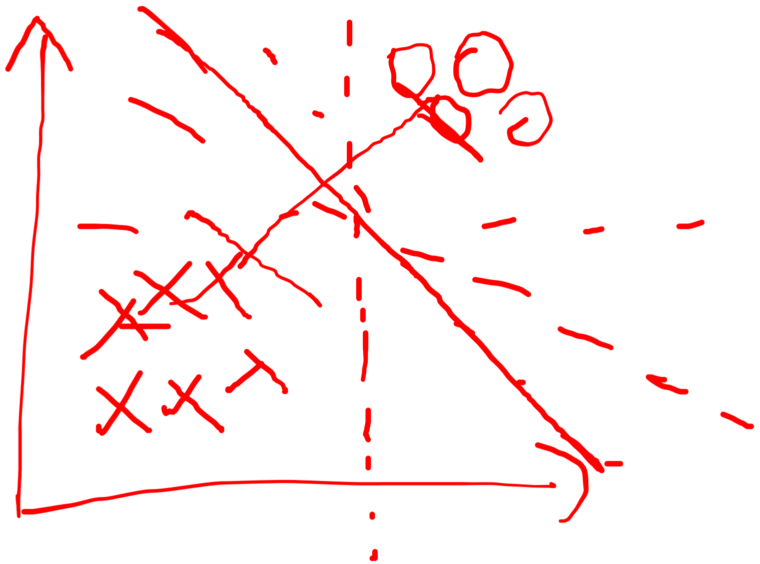




CALVIN INSTITUTE OF TECHNOLOGY

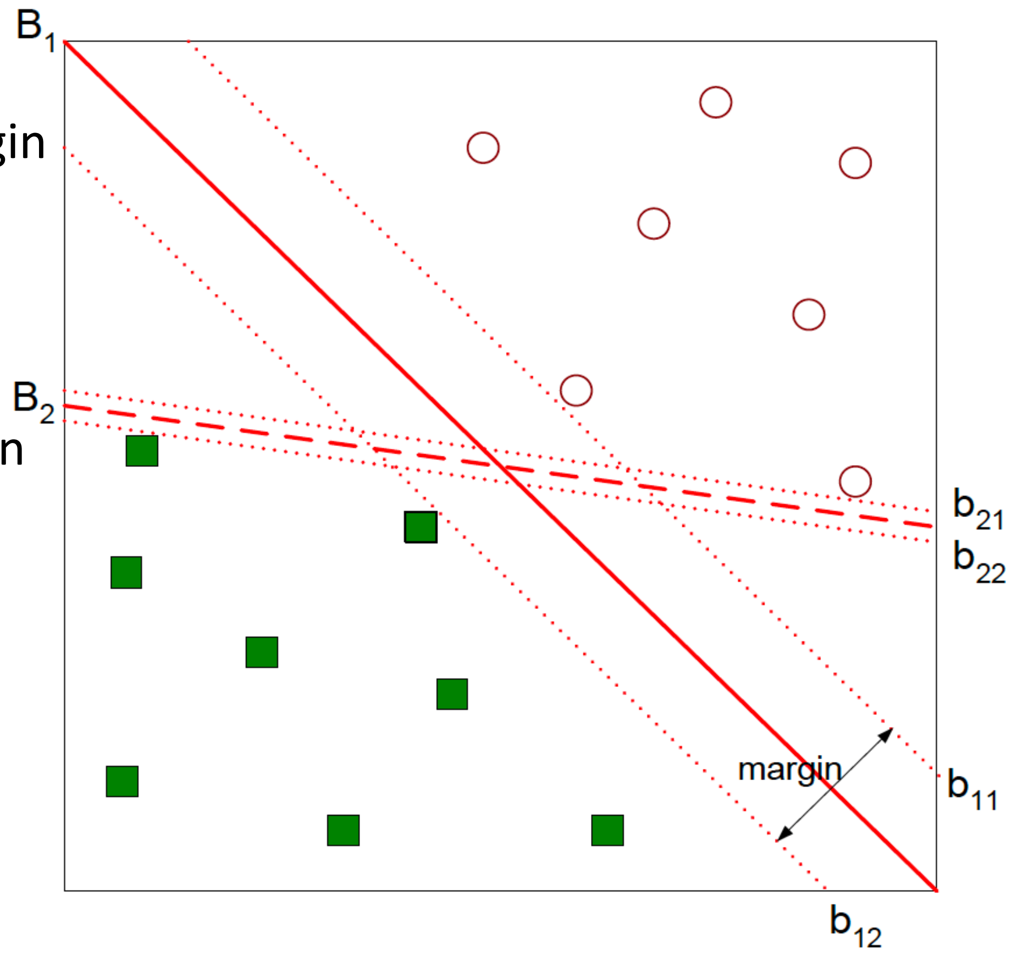
SVM Linear

- Manakah garis yang paling sesuai untuk memisahkan data? (bagaimana performa terhadap data baru?)
- Kelas yang berbeda dapat dengan mudah dipisahkan oleh lebih dari satu kemungkinan hyperplane (linearly separable)
- Metode SVM akan memilih garis yang berada sejauh mungkin dari data latih terdekat→large margin classifier



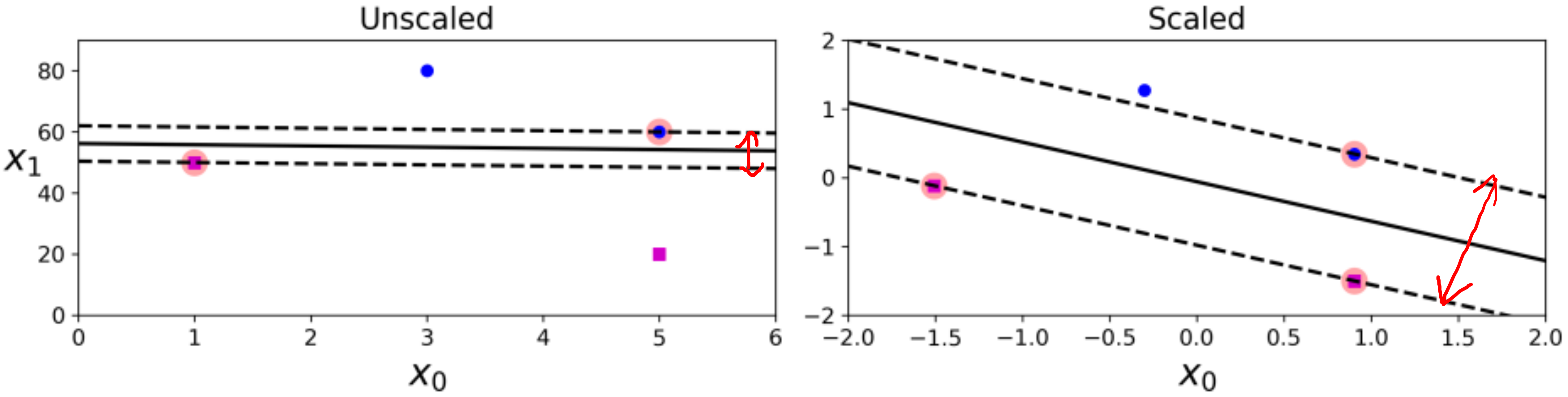
Intuisi: Margin Maksimum

- Intuisi sebuah margin:
  - Margin: lebar dari batas yang dapat diperbesar sebelum menyentuh sebuah data
- SVM:
  - Menemukan hyperplane yang dapat memaksimalkan margin
  - Batas keputusan harus sejauh mungkin dari data tiap kelas
- Support vector:
  - Data dari masing-masing kelas di tepi
  - Istilah: penambahan data latih lainnya berada diluar margin
    - →tidak mengubah Batasan keputusan
  - Batasan keputusan didukung (supported) oleh data di tepi



SVM sensitif terhadap scaling

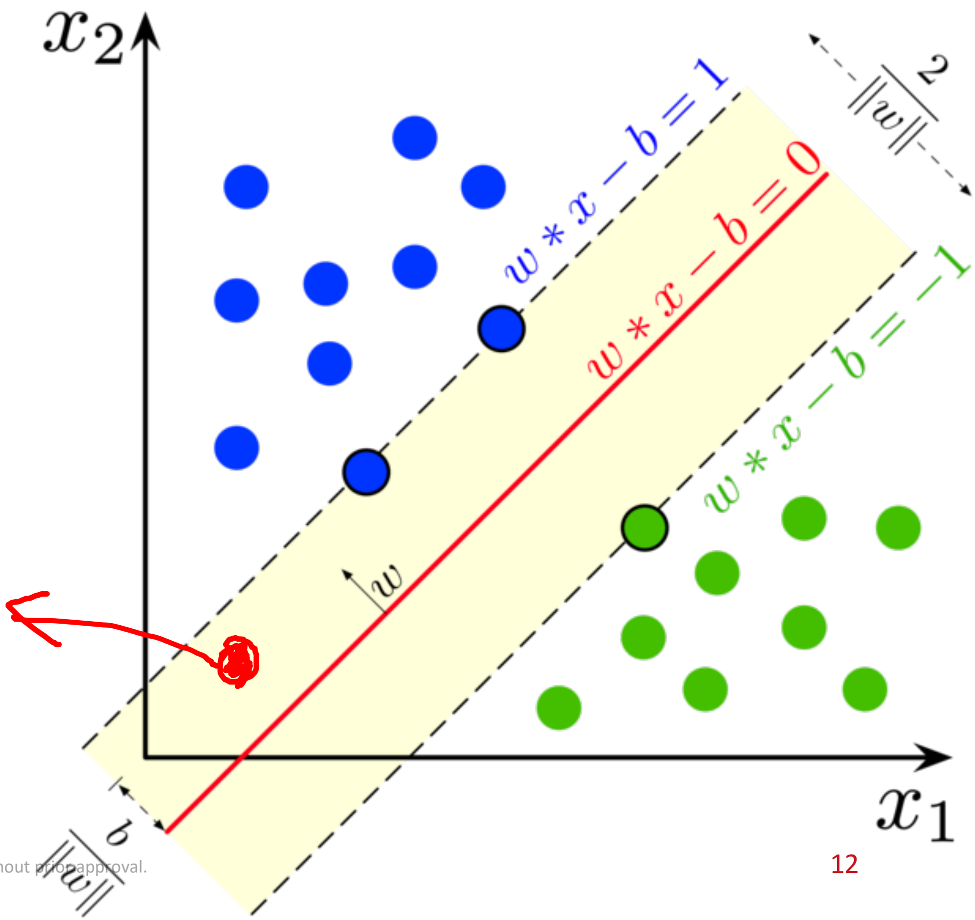
- Unscaled: skala vertical jauh lebih besar dari horizontal
- Scaled: Batasan keputusan nampak lebih jelas



Prediksi SVM

- Klasifikasi SVM linear →prediksi kelas dari data baru **x** dengan menghitung fungsi keputusan h: $h = \boldsymbol{w}^T \boldsymbol{x} + b = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$
- Prediksi: $\hat{y} = \begin{cases} 0, jika\ h < 0 \\ 1, jika\ h \geq 0 \end{cases}$ −ve +ve
- Batas keputusan: garis tegas dimana $h = 0$
- Batas margin: garis putus-putus dimana $h = \pm 1$

biru



Optimisasi

lagrange multiplier

- Dataset: $\{x_1, \ldots, x_n\}$, label: $y^{(i)} \in \{+1, -1\}$
- Batas keputusan harus sejauh mungkin dari data tiap kelas→memaksimalkan margin:

$$m = \frac{2}{||\boldsymbol{w}||} \rightarrow ||\boldsymbol{w}||^2 = \boldsymbol{w}^T\boldsymbol{w}$$

- Batasan keputusan harus meklasifikasi setiap data secara benar:

$$y^{(i)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b\right) \geq 1, \forall_i$$

- A linear constrained convex quadratic optimization problem (quadratic programming):

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

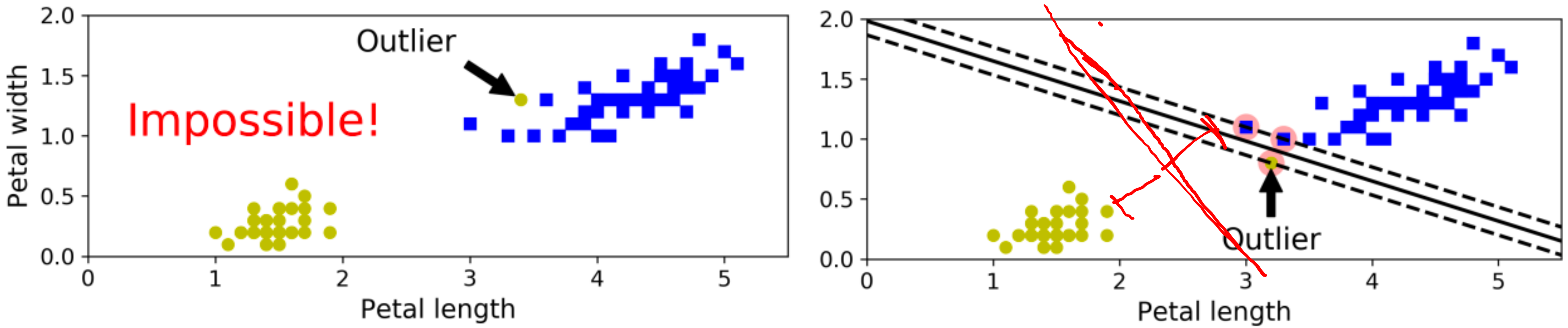
$$subject\ to\ \rightarrow \underline{y^{(i)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b\right)} \geq 1, \forall_i$$

↳ constraint

CALVIN INSTITUTE OF TECHNOLOGY

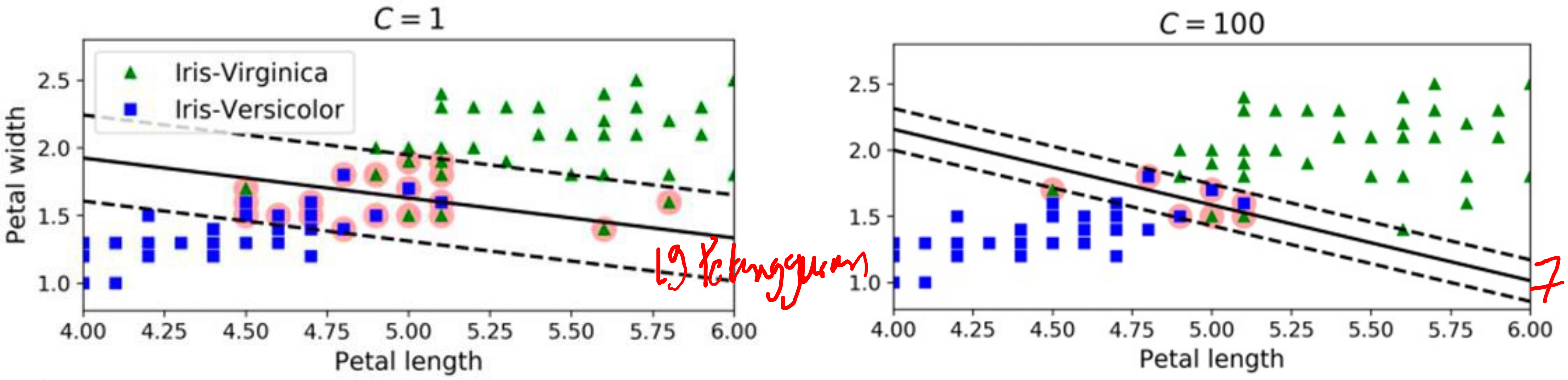
Hard Margin

- Hard margin→semua data harus di luar margin
- Masalah dengan hard margin:
  - Hanya bisa dilakukan jika data linearly separable
  - Sensitif terhadap pencilan: margin menjadi sempit



Soft Margin

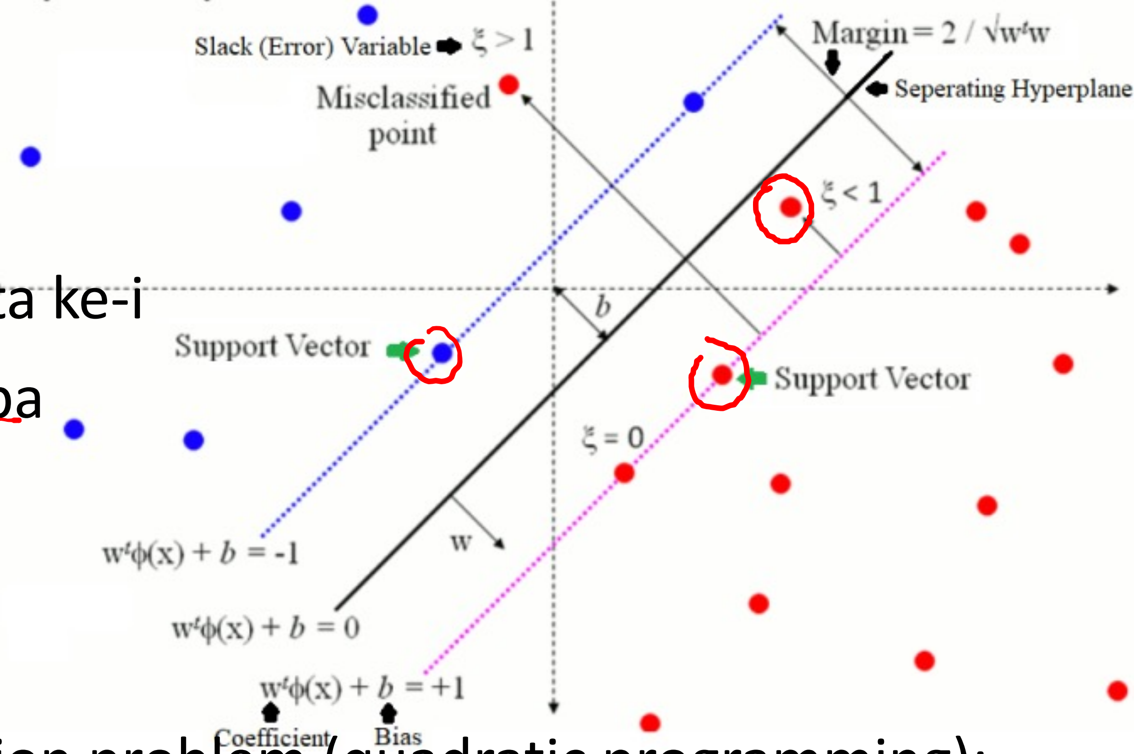
- Soft margin merupakan kompromi antara:
  - margin selebar mungkin (margin optimization)
  - membatasi pelanggaran batas (margin violation)
- Margin violation: data yang melanggar batas margin
- Kompromi ini diatur oleh hyperparameter C (nilai kecil artinya margin besar)
- Kasus C mana yang cenderung lebih baik generalisasinya?



19 pelanggaran

7 pelanggaran

Optimisasi Soft Margin



- Menggunakan slack variables $\zeta^{(i)} \geq 0$ untuk data ke-i
- $\zeta^{(i)}$ →data ke-i boleh melanggar batas sejauh apa
- 2 tujuan yang berseberangan:
  - $\zeta^{(i)}$ sekecil mungkin: menurunkan pelanggaran batas
  - $\frac{1}{2}||\boldsymbol{w}||^2$ sekecil mungkin: membesarkan margin
- A linear constrained convex quadratic optimization problem (quadratic programming):

margin

$$\boxed{\min_{\boldsymbol{w},b,\zeta} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}} + \boxed{C\sum_{i=1}^{m}\zeta^{(i)}}$$

pelanggaran

$$subject\ to\ \rightarrow y^{(i)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(i)} + b\right) \geq 1 - \zeta^{(i)}\ dan\ \zeta^{(i)} \geq 0, \forall_i$$

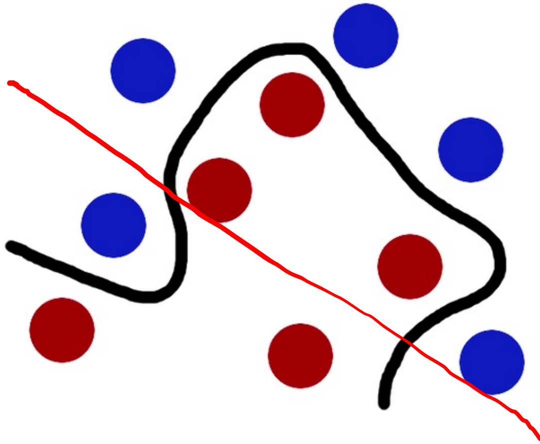
Hyperparameter

CALVIN INSTITUTE OF TECHNOLOGY

Non-Linear SVM

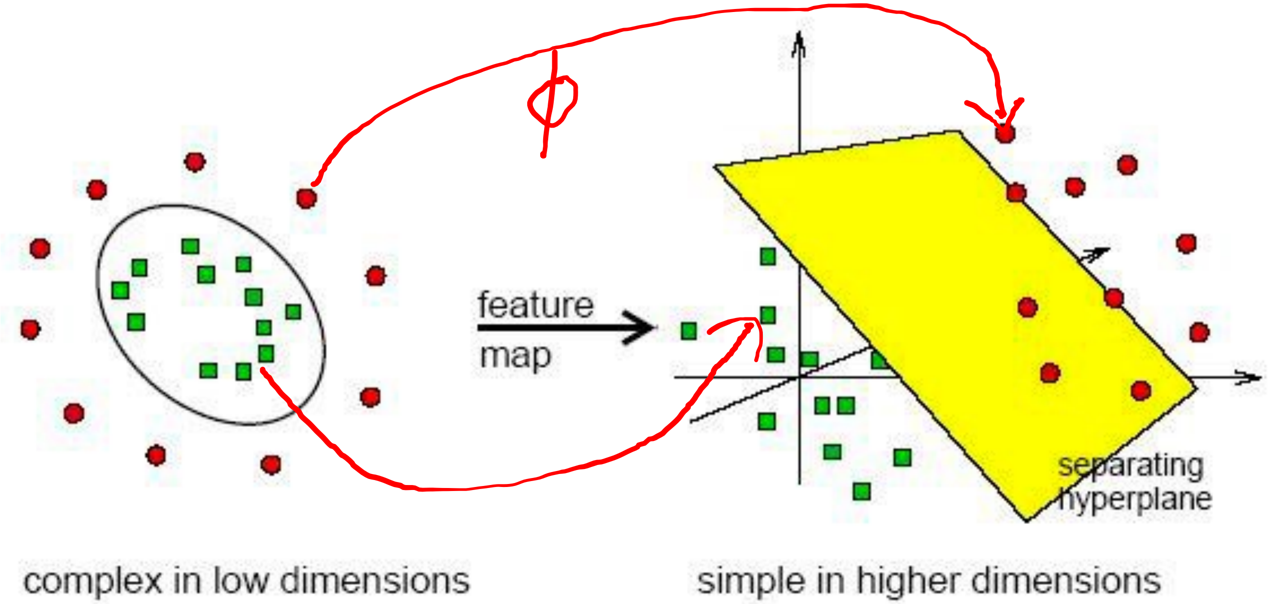
SVM NonLinear

$\phi(x)$



- Banyak data tidak terpisahkan secara linear
- Kernel trick: SVM linear untuk data yang non-linear
- Fungsi kernel→memetakan data non-linear ke dimensi yang lebih tinggi sehingga dapat terpisahkan secara linear
- Menambah fitur: misalnya dengan fitur polynomial lebih tinggi→mirip dengan regresi polynomial

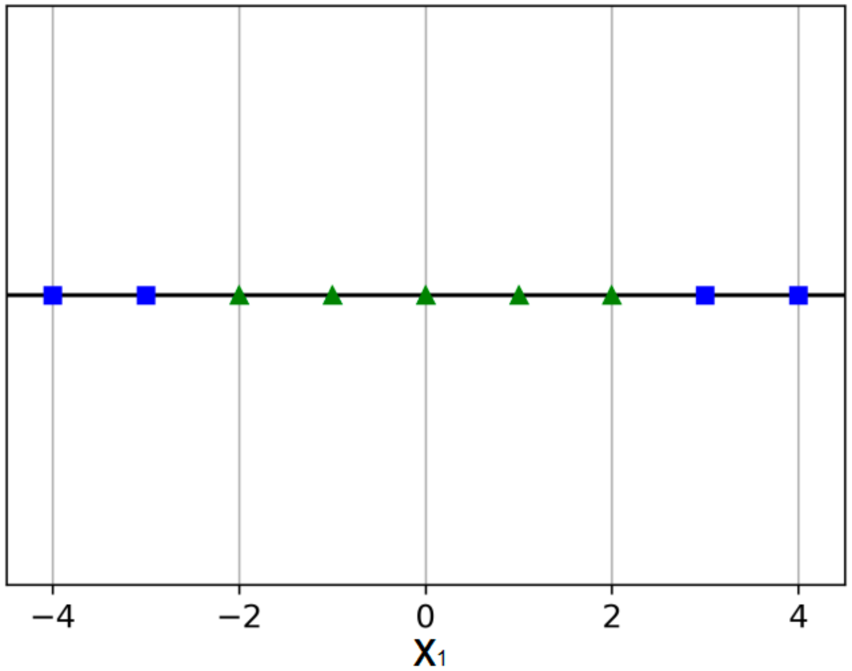
Separation may be easier in higher dimensions



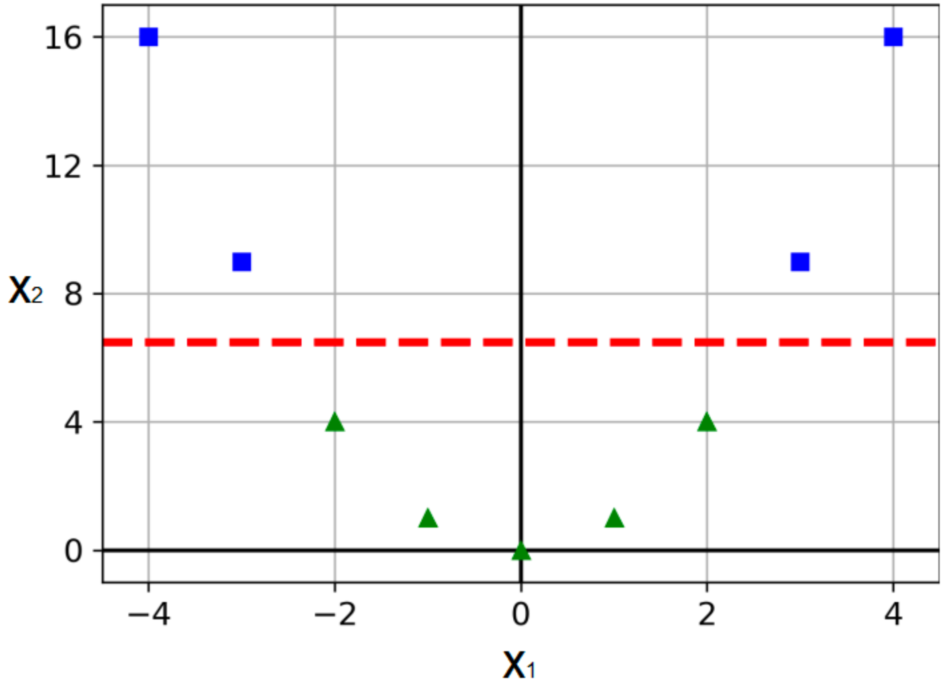
Feature Mapping

- Data dengan 1 fitur (x_1) → tidak terpisahkan linear
- Tambahakan 1 fitur polynomial derajat 2 ($x_2 = x_1^2$)→terpisahkan linear

1D $\xrightarrow{\phi}$ 2D

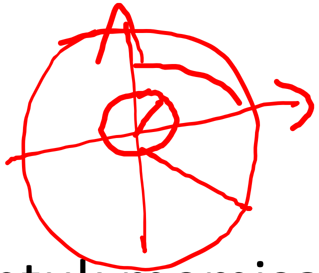


$x_1^2 =$



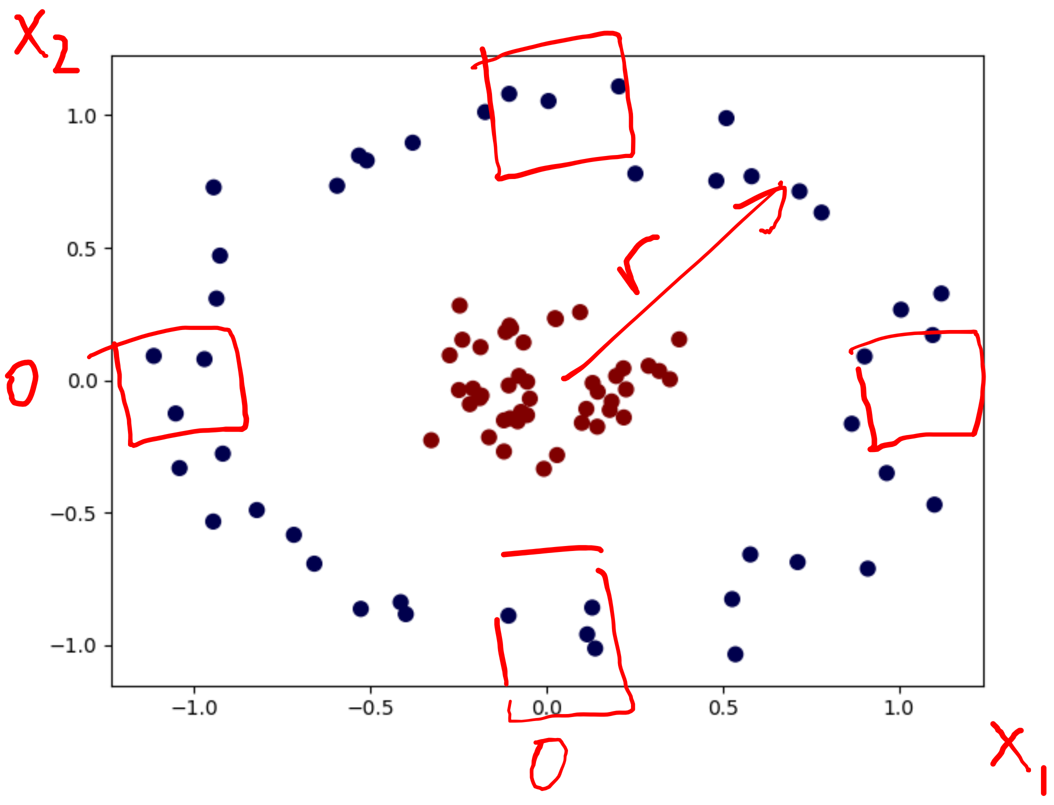
CALVIN INSTITUTE OF TECHNOLOGY

Uji Pemahaman

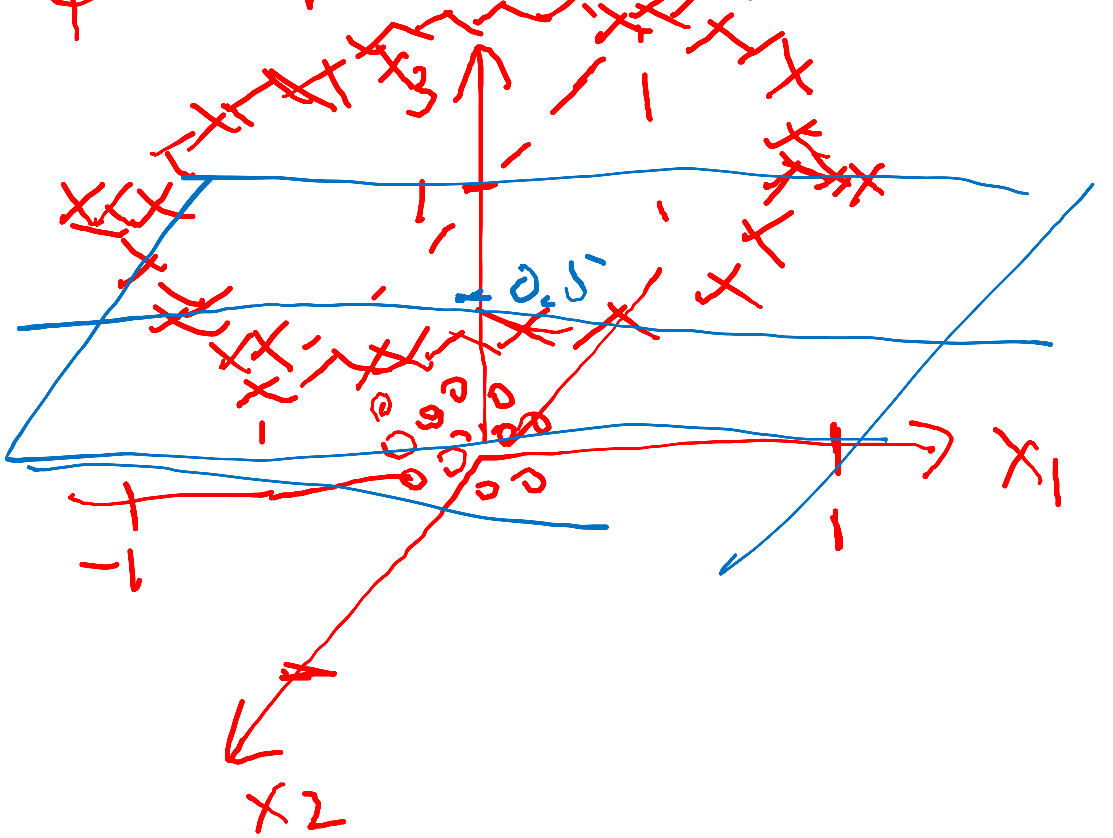


$x_1^2 + x_2^2 = r^2 = x_3$

- Gunakan polynomial mapping untuk memisahkan data berikut secara linear



$\Phi \rightarrow$ persamaan lingkaran



Optimisasi SVM Nonlinear

- Setelah menggunakan feature mapping:

data $\quad$ proyeksi data

$$\boldsymbol{x} \mapsto \Phi(\boldsymbol{x})$$

- Temukan hyperplane pada feature space

$$f(x) = \boldsymbol{w} \cdot \underline{\Phi(\boldsymbol{x})} + b$$

- Hampir sama dengan optimisasi SVM linear

$$\min_{\boldsymbol{w},b,\zeta} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{m}\zeta^{(i)}$$

$$subject\ to\ \rightarrow y^{(i)}\left(\boldsymbol{w}^T\phi(\boldsymbol{x}^{(i)}) + b\right) \geq 1 - \zeta^{(i)}\ dan\ \zeta^{(i)} \geq 0, \forall_i$$

Bagaimana cara memilih feature map

$$\boldsymbol{x} \mapsto \phi(\boldsymbol{x})$$

- Fitur polynomial:
  - Jika derajat terlalu rendah→tidak dapat terpisahkan linear
  - Jika derajat terlalu tinggi→jumlah fitur besar →model terlalu lambat (butuh banyak memori dan sulit menyelesaikan quadratic programming)
- Contoh:

$x_1, x_2, x_3 \longrightarrow \phi$ derajat 2

$$\boldsymbol{x} \in R^3, \phi(\boldsymbol{x}) \in R^{10}$$

$$\phi(\boldsymbol{x}) = (1, x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3)$$

CALVIN INSTITUTE OF TECHNOLOGY

Kernel Tricks

- SVM dengan fitur polynomial derajat tinggi dapat dipercepat dengan kernel trick
- Ide: ganti dot product dengan sebuah kernel

$$\kappa\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \left\langle \phi\left(\boldsymbol{x}^{(i)}\right), \phi\left(\boldsymbol{x}^{(j)}\right) \right\rangle$$

dot product

- Fungsi kernel:
  - Linear kernel

$$\kappa\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \left\langle \boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)} \right\rangle = {\boldsymbol{x}^{(i)}}^T \boldsymbol{x}^{(j)}$$

  - Polynomial kernel

$$\kappa\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \left\langle \phi\left(\boldsymbol{x}^{(i)}\right), \phi\left(\boldsymbol{x}^{(j)}\right) \right\rangle = \left({\boldsymbol{x}^{(i)}}^T \boldsymbol{x}^{(j)}\right)^d$$

  - Gaussian kernel / RBF kernel

$$\kappa\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \left\langle \phi\left(\boldsymbol{x}^{(i)}\right), \phi\left(\boldsymbol{x}^{(j)}\right) \right\rangle = \exp\left(-\frac{\left\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right\|^2}{2\sigma^2}\right)$$

$$\kappa\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \left\langle \phi\left(\boldsymbol{x}^{(i)}\right), \phi\left(\boldsymbol{x}^{(j)}\right) \right\rangle = \exp\left(-\gamma \left\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right\|^2\right)$$

CALVIN INSTITUTE OF TECHNOLOGY

Contoh

- Polinomial derajat 2 pada data latih 2 fitur / dimensi:

$$\phi(x_1, x_2) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$$

- Dot product hasil feature mapping = kernel dari dot product vektor asal

$$\phi(\boldsymbol{a})^T\phi(\boldsymbol{b}) = \begin{pmatrix} a_1^2 \\ \sqrt{2}a_1a_2 \\ a_2^2 \end{pmatrix}^T \begin{pmatrix} b_1^2 \\ \sqrt{2}b_1b_2 \\ b_2^2 \end{pmatrix} = a_1^2b_1^2 + 2a_1b_1a_2b_2 + a_2^2b_2^2$$

$$= (a_1b_1 + a_2b_2)^2 = \left(\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}^T \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}\right)^2 = (\boldsymbol{a}^T\boldsymbol{b})^2$$

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \cdot \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

CALVIN INSTITUTE OF TECHNOLOGY

Uji Pemahaman

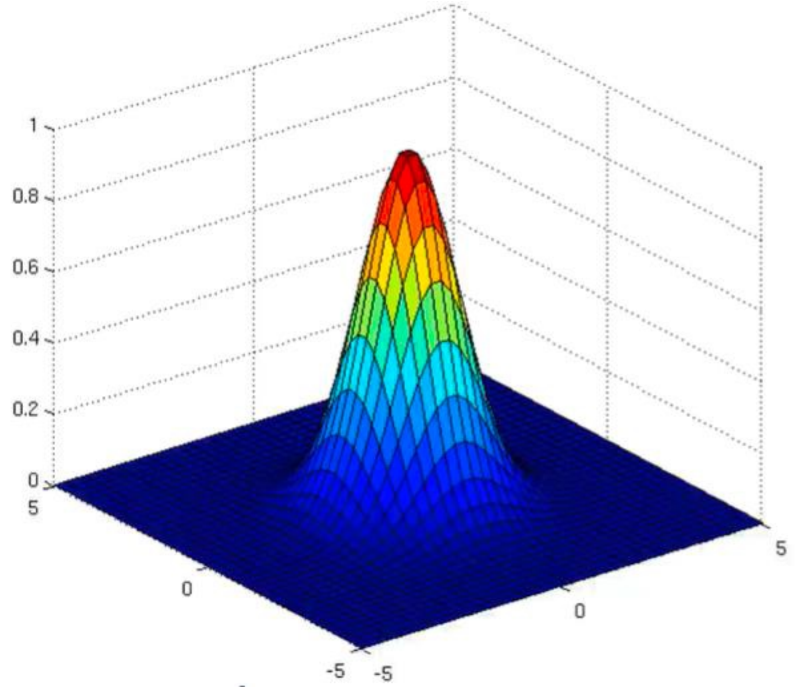
- $\boldsymbol{x}^{(1)} = \left(x_1^{(1)}, x_2^{(1)}, x_3^{(1)}\right) = (1,2,3), \quad \boldsymbol{x}^{(2)} = \left(x_1^{(2)}, x_2^{(2)}, x_3^{(2)}\right) = (4,5,6)$
- $\phi\left(\boldsymbol{x}^{(1)}\right) = \left(x_1^{(1)}x_1^{(1)}, x_1^{(1)}x_2^{(1)}, x_1^{(1)}x_3^{(1)}, x_2^{(1)}x_1^{(1)}, x_2^{(1)}x_2^{(1)}, x_2^{(1)}x_3^{(1)}, x_3^{(1)}x_1^{(1)}, x_3^{(1)}x_2^{(1)}, x_3^{(1)}x_3^{(1)}\right) = (1,2,3,2,4,6,3,6,9)$
- $\phi\left(\boldsymbol{x}^{(2)}\right) = \left(x_1^{(2)}x_1^{(2)}, x_1^{(2)}x_2^{(2)}, x_1^{(2)}x_3^{(2)}, x_2^{(2)}x_1^{(2)}, x_2^{(2)}x_2^{(2)}, x_2^{(2)}x_3^{(2)}, x_3^{(2)}x_1^{(2)}, x_3^{(2)}x_2^{(2)}, x_3^{(2)}x_3^{(2)}\right) = (16,20,24,20,25,30,24,30,36)$
- Hitung $\kappa\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right)$ secara manual, lalu bandingkan dengan kernel tricks

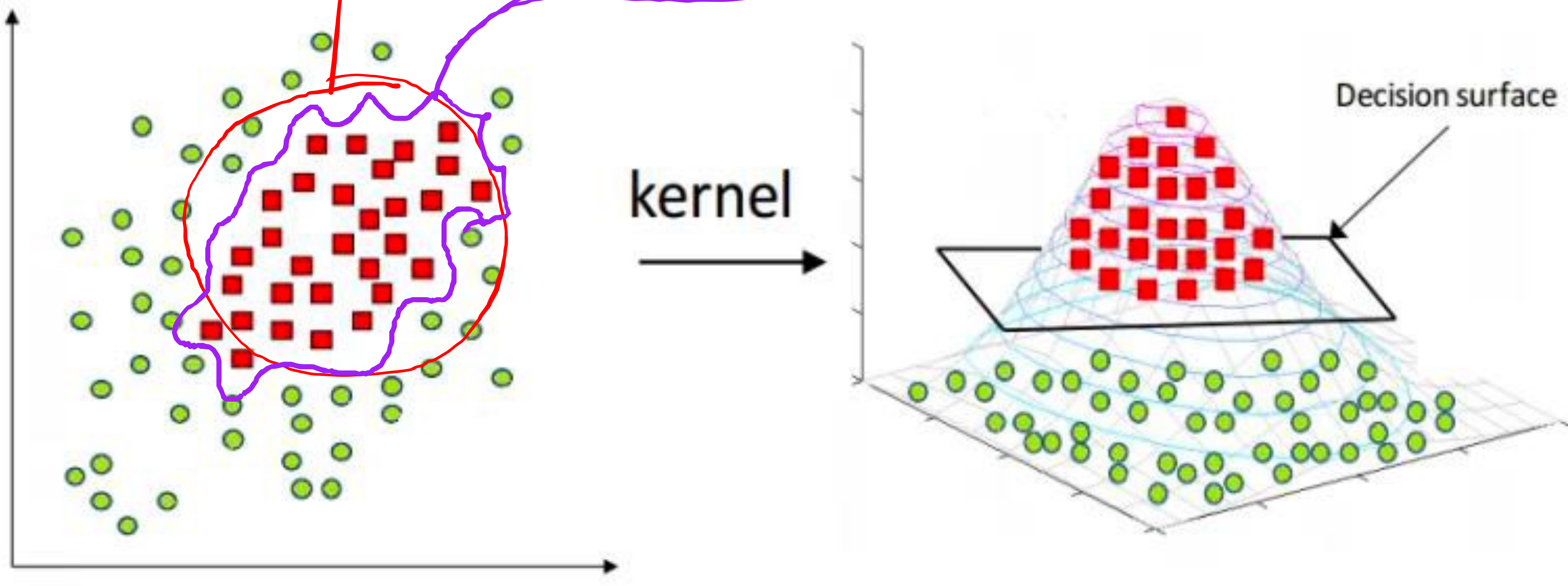
$\phi(x^{(1)}) \cdot \phi(x^{(2)}) = 16 + 40 + 72 + 40 + 100 + 180 + 72 + 180 + 324 = 1024$

$\phi(x^{(1)}, x^{(2)}) = \phi(4, 10, 18) = 16 + 100 + 324 + 2 \times 40 + 2 \times 180 + 2 \times 72 = 1024$

RBF Kernel

- $\kappa\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) = \left\langle \phi\left(\boldsymbol{x}^{(i)}\right), \phi\left(\boldsymbol{x}^{(j)}\right) \right\rangle = \exp\left(-\gamma \left|\left|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right|\right|^2\right)$
- γ →hyperparameter: γ terlalu kecil→underfitting, γ terlalu besar→overfitting





Optimisasi dengan Kernel

- Proses optimisasi dapat dilihat sebagai primal maupun dual problem (https://en.wikipedia.org/wiki/Duality_(optimization))
- Dalam optimisasi SVM secara dual, hipotesis akhir dapat ditulis sebagai:

kernel

$$w = \sum_{i=1}^{N} \alpha_i y^{(i)} \Phi\left(\boldsymbol{x}^{(i)}\right)$$

$$f(x) = \boldsymbol{w}^T \Phi(\boldsymbol{x}) + b = \sum_{i=1}^{N} \alpha_i y^{(i)} \Phi\left(\boldsymbol{x}^{(i)}\right) \Phi(\boldsymbol{x}) + b = \boxed{\sum_{i=1}^{N} \alpha_i y^{(i)} \kappa(\boldsymbol{x}_i, \boldsymbol{x}) + b}$$

Kernel Learning

- Manakah kernel terbaik dari banyak alternatif kernel?
- Multiple kernel learning (MKL): mencari kombinasi optimal dari beberapa kernel

$$\kappa(\boldsymbol{a}, \boldsymbol{b}, ; \gamma) = \sum_{k=1}^{m} \gamma_k \kappa_k(\boldsymbol{a}, \boldsymbol{b})$$

- MKL framework→konstruksi fungsi kernel:
  - Berbagai macam fitur
  - Berbagai tipe kernel → RBF, Poly 2, Poly 3, ...
  - Berbagai parameter kernel γ, d

Curse of kernelization

- Melatih kernel classifier sangat computationally expensive
- SVM linear butuh linear time $O(N)$
- SVM kernel dengan QP solvers butuh $O(N^3)$
- Bagaimana cara melatih pada large-scale dataset? Gunakan aproksimasi kernel

Aproksimasi Kernel

- Tujuan: buat representasi baru $\mathbf{z}(\boldsymbol{x}) \in \mathbb{R}^D$ sehingga $\kappa\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}^{(j)}\right) \approx \mathbf{z}\left(\boldsymbol{x}^{(i)}\right)^T \mathbf{z}\left(\boldsymbol{x}^{(j)}\right)$
- Model linear:
  - Hipotesis dapat ditulis ulang sebagai:

$$f(x) = \sum_{i=1}^{M} \alpha_i \kappa\left(\boldsymbol{x}^{(i)}, \boldsymbol{x}\right) \approx \sum_{i=1}^{M} \alpha_i \mathbf{z}\left(\boldsymbol{x}^{(i)}\right)^T \mathbf{z}(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{z}(\boldsymbol{x})$$

$$dimana\ \boldsymbol{w}^T = \sum_{i=1}^{M} \alpha_i \mathbf{z}\left(\boldsymbol{x}^{(i)}\right)$$

  - Gunakan linear classifier pada representasi baru $\mathbf{z}$
- 2 metode:
  - Aproksimasi fungsional: metode Fourier
  - Aproksimasi matriks: metode Nystrom

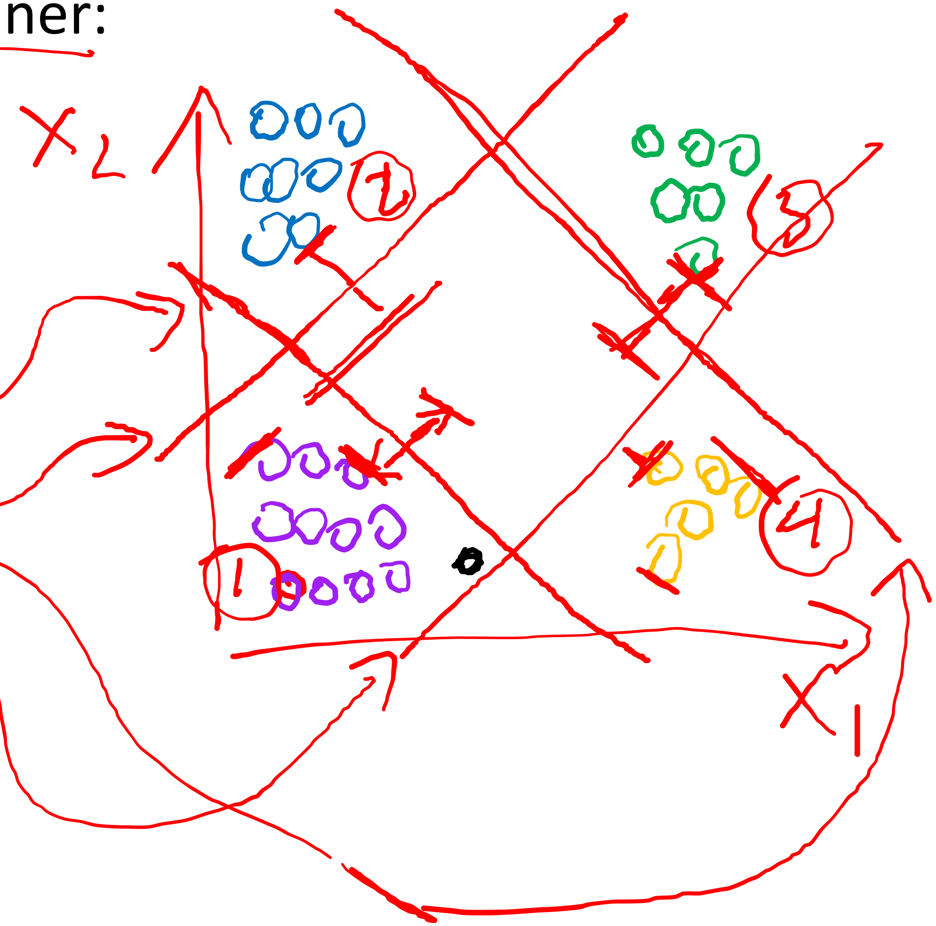
SVM Multi Kelas

Klasifikasi multi kelas

OvR

- Misalkan terdapat K kelas
- Gunakan one versus rest (OvR): latih sebanyak K SVM biner:
  - 1^{st} class vs (2-k)th class
  - 2^{nd} class vs (1,3-k)th class
  - …
- K decision functions:

$$
\begin{gathered}
(\boldsymbol{w}_1)^T\phi(\boldsymbol{x}) + b_1 \\
(\boldsymbol{w}_2)^T\phi(\boldsymbol{x}) + b_2 \\
\vdots \\
(\boldsymbol{w}_K)^T\phi(\boldsymbol{x}) + b_K
\end{gathered}
$$



CALVIN INSTITUTE OF TECHNOLOGY

OvR

- Prediksi

$$\arg\max_{k} (\boldsymbol{w}_k)^T \phi(\boldsymbol{x}) + b_k$$

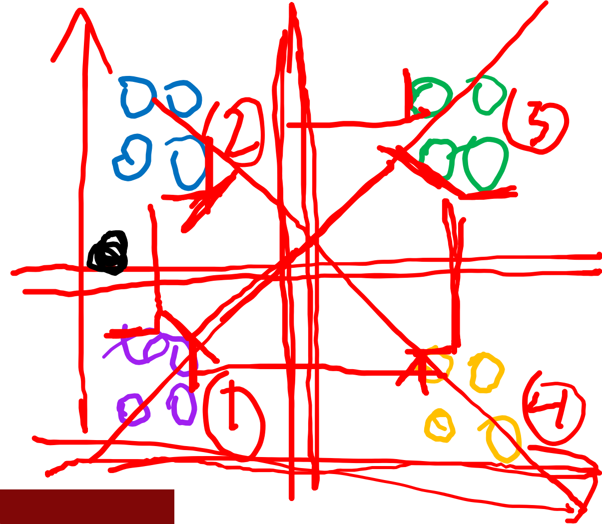
- Jika prediksinya adalah 1st class:

$$\begin{gathered}(\boldsymbol{w}_1)^T \phi(\boldsymbol{x}) + b_1 \geq +1 \\ (\boldsymbol{w}_2)^T \phi(\boldsymbol{x}) + b_2 \leq -1 \\ \vdots \\ (\boldsymbol{w}_K)^T \phi(\boldsymbol{x}) + b_K \leq -1\end{gathered}$$

CALVIN INSTITUTE OF TECHNOLOGY

OvO

$\frac{4(4-1)}{2} = 6$



- One versus one: latih K(K-1)/2 SVM biner:
  $$(1,2), (1,3), \dots, (1,k), (2,3), (2,4), \dots, (k-1,k)$$
- Contoh: jika terdapat 4 kelas→butuh 6 SVM biner

$y^{(i)} = +1$	$y^{(i)} = -1$	Fungsi keputusan
Kelas 1	Kelas 2	$f_{12}(\boldsymbol{x}) = (\boldsymbol{w}_{12})^T \boldsymbol{x} + b_{12}$
Kelas 1	Kelas 3	$f_{13}(\boldsymbol{x}) = (\boldsymbol{w}_{13})^T \boldsymbol{x} + b_{13}$
Kelas 1	Kelas 4	$f_{14}(\boldsymbol{x}) = (\boldsymbol{w}_{14})^T \boldsymbol{x} + b_{14}$
Kelas 2	Kelas 3	$f_{23}(\boldsymbol{x}) = (\boldsymbol{w}_{23})^T \boldsymbol{x} + b_{23}$
Kelas 2	Kelas 4	$f_{24}(\boldsymbol{x}) = (\boldsymbol{w}_{24})^T \boldsymbol{x} + b_{24}$
Kelas 3	Kelas 4	$f_{34}(\boldsymbol{x}) = (\boldsymbol{w}_{34})^T \boldsymbol{x} + b_{34}$

- Pilih kelas dengan voting terbanyak

OvR vs OvO

$\to$ QP $\to$ d=3

- Asumsikan optimisasi SVM dengan ukuran n adalah $O(n^d)$
- OvR: terdapat K model, masing-masing n data→$KO(n^d)$
- OvO: terdapat K(K-1)/2 model, masing-masing 2n/K data→$\frac{K(K-1)}{2}O((2n/K)^d)$

OvR : $10 \times O(1M^3)$

OvO : $45 \times O(200K^3)$

Regresi SVM

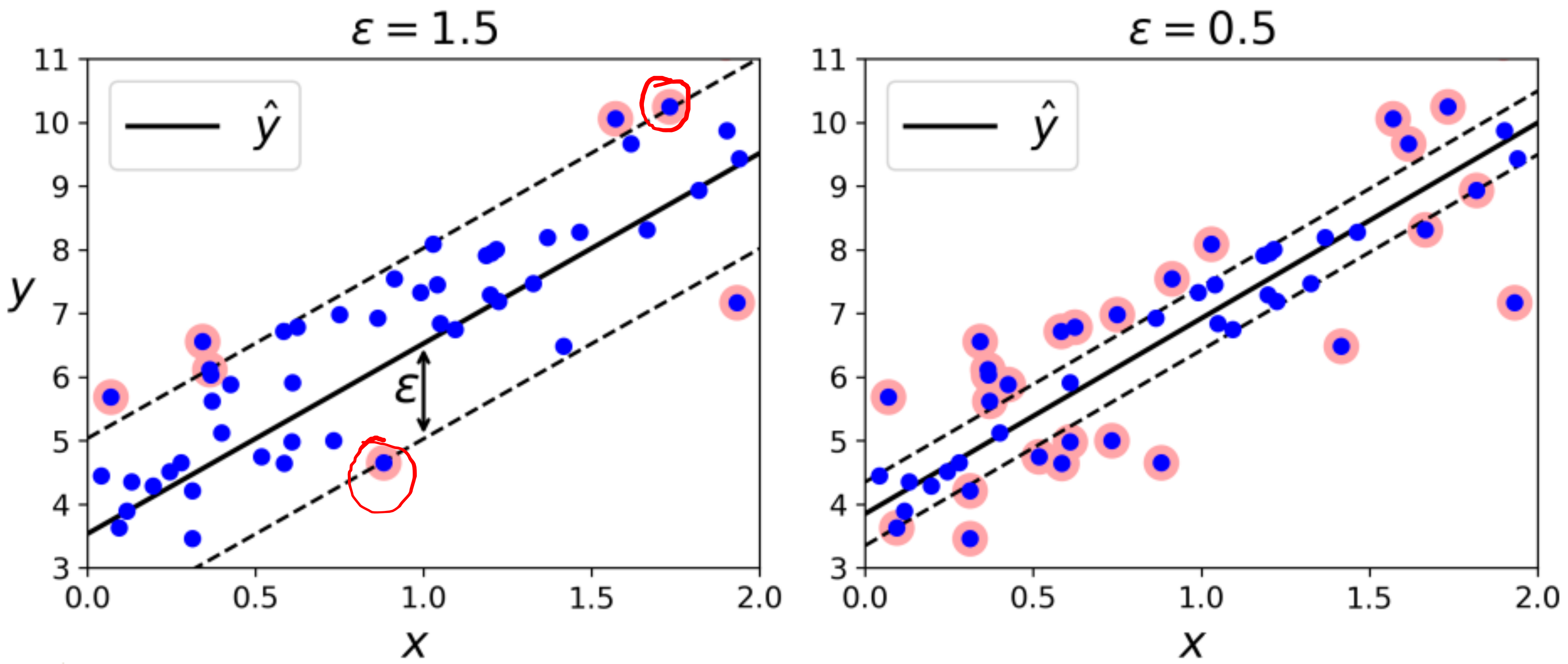
God's People for God's Glory
CALVIN
INSTITUTE OF TECHNOLOGY

Regresi dengan SVM

- SVM dapat digunakan untuk regresi
- Klasifikasi:
  - membuat margin selebar mungkin di antara dua kelas
  - Meminimalkan margin violation: sesedikit mungkin data yang melewati margin
- Regresi:
  - Membuat sebanyak mungkin data masuk dalam margin
  - Meminimlkan margin violation: sesedikit mungkin data melewati margin

Regresi SVM

- Lebar margin dikontrol oleh hyperparameter ϵ
- Apa akibat penambahan data di dalam margin?
- Tidak berdampak pada prediksi model→ϵ-insensitive (linear)



Optimisasi Regresi SVM

- Tujuan: memperoleh model $\hat{f}(x)$ dengan prediksi $\hat{y}$ yang berbeda dengan target y maksimal sejauh ϵ
- Formulasi permasalahan optimisasi dengan soft constraint:

Violation

margin

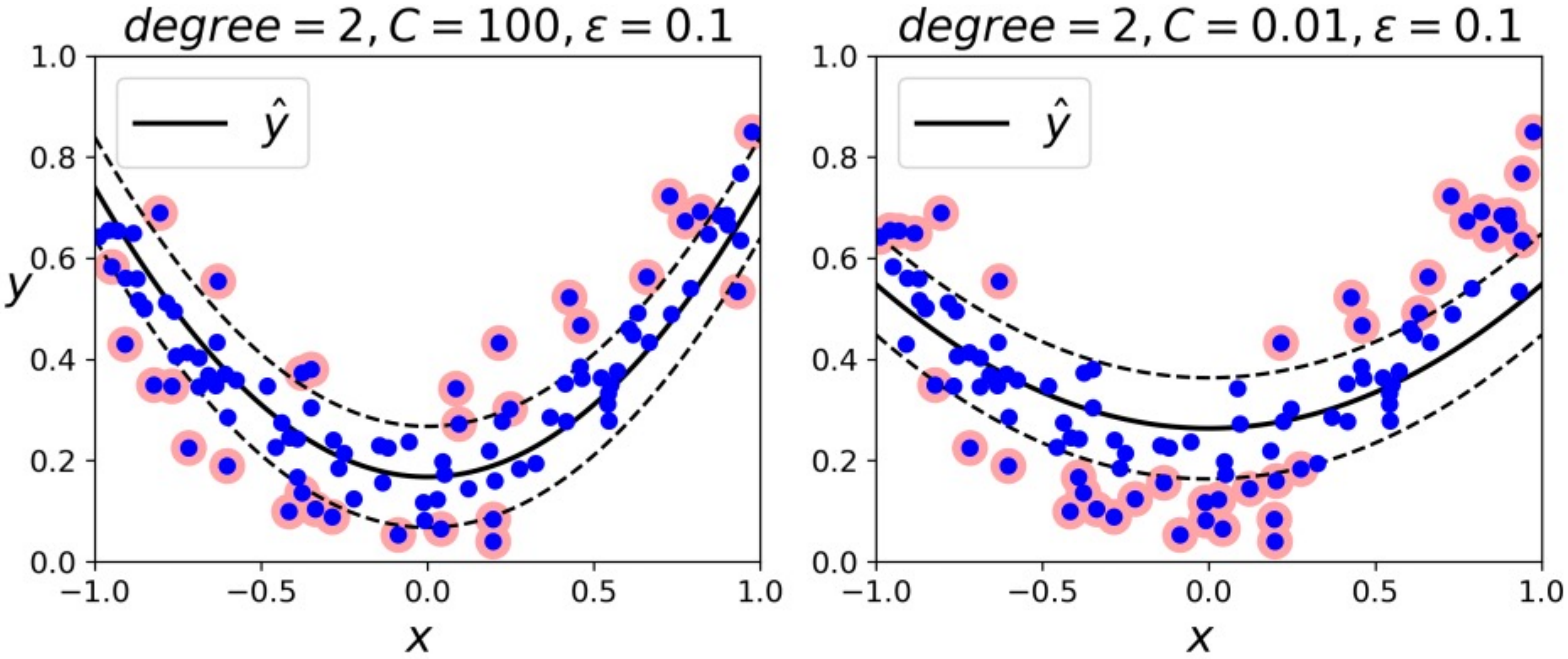
$$\min_{w,b,\zeta,\zeta^*} \boxed{\frac{1}{2} w^T w} + C \boxed{\sum_{i=1}^{n} (\zeta_i + \zeta_i^*)}$$

$$\begin{aligned} \text{subject to } & y_i - w^T \phi(x_i) - b \leq \varepsilon + \zeta_i, \\ & w^T \phi(x_i) + b - y_i \leq \varepsilon + \zeta_i^*, \\ & \zeta_i, \zeta_i^* \geq 0, i = 1, \ldots, n \end{aligned}$$

Regresi SVM Nonlinear

- Untuk data non-linear, kita dapat memakai SVM dengan kernel
- Ilustrasi: data kuadratik (polynomial derajat 2) dengan hyperparameter C yang berbeda
- Kasus mana lebih underfitting?

Underfitting



Tuhan Memberkati