Machine Learning

Lecture 4: Support Vector Machine

Asst. Prof. Dr. Santitham Prom-on

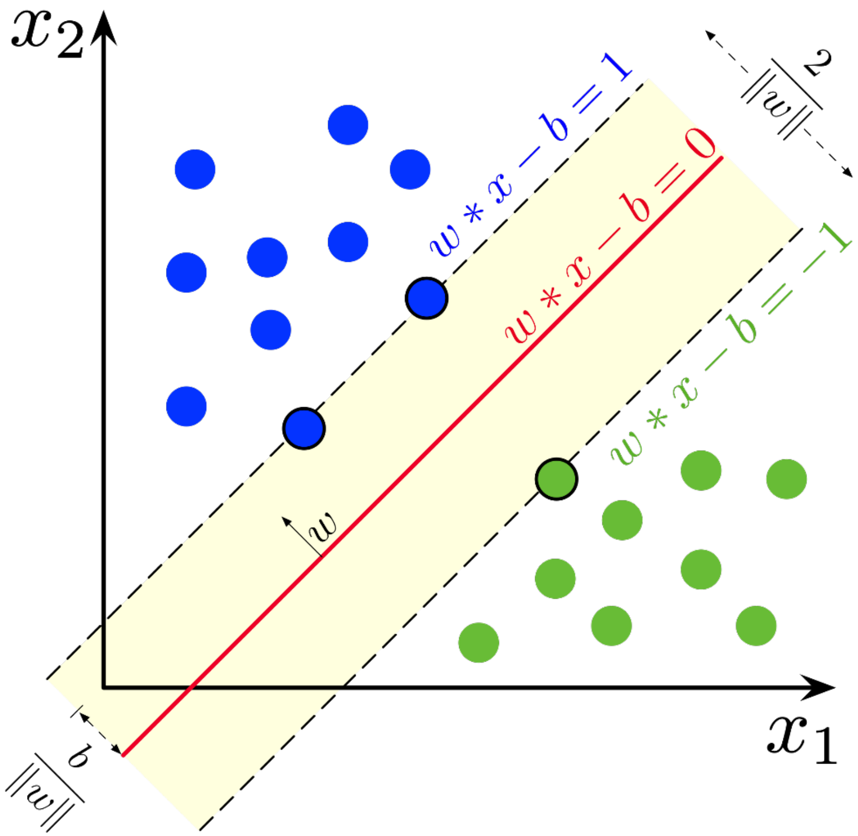
Department of Computer Engineering, Faculty of Engineering
King Mongkut's University of Technology Thonburi

Topics

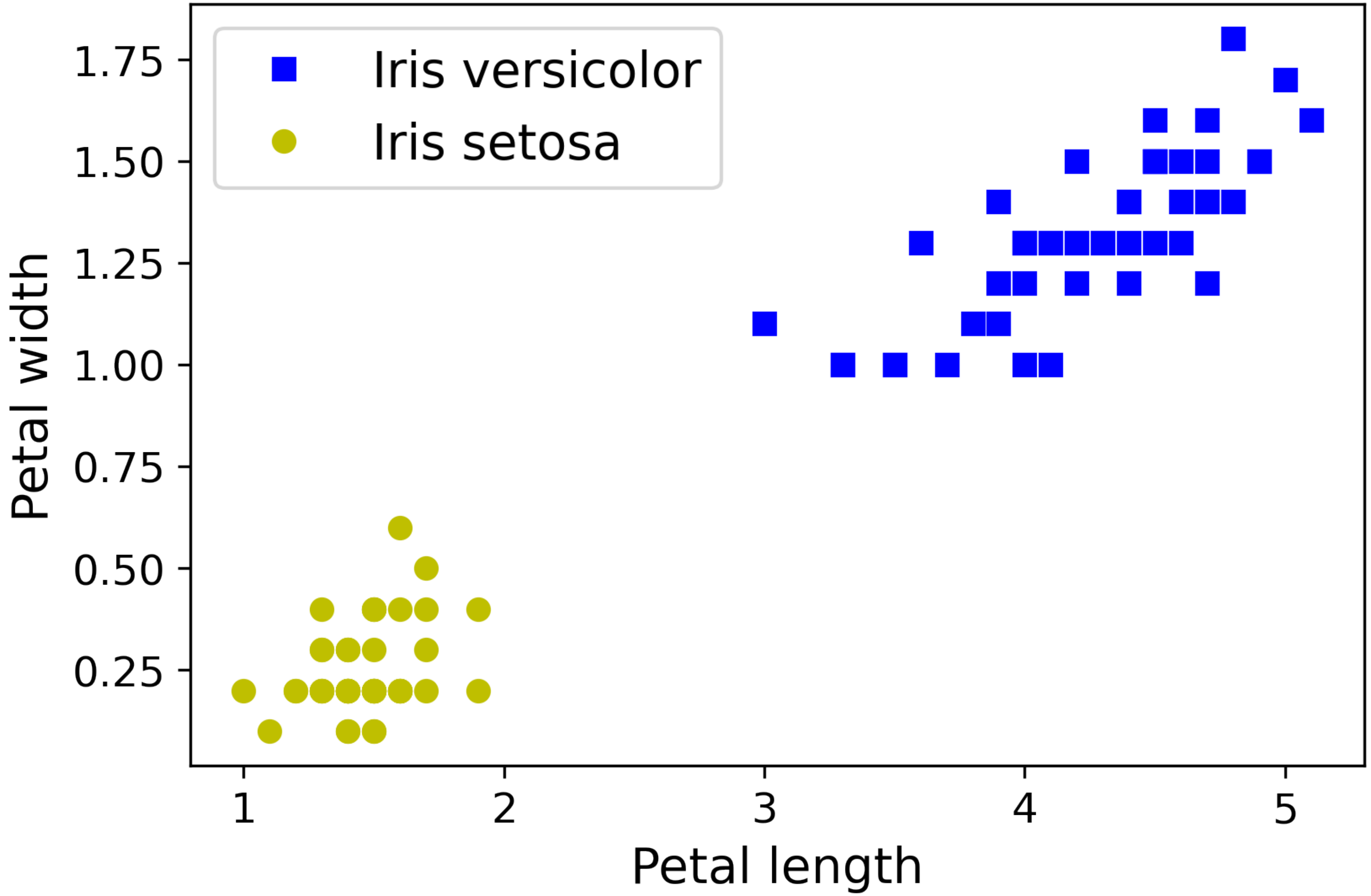
- Linear SVM classification
- Soft margin
- Training SVM
- Nonlinear SVM classification
- Kernel trick
- Hinge Loss

Support Vector Machine (SVM)

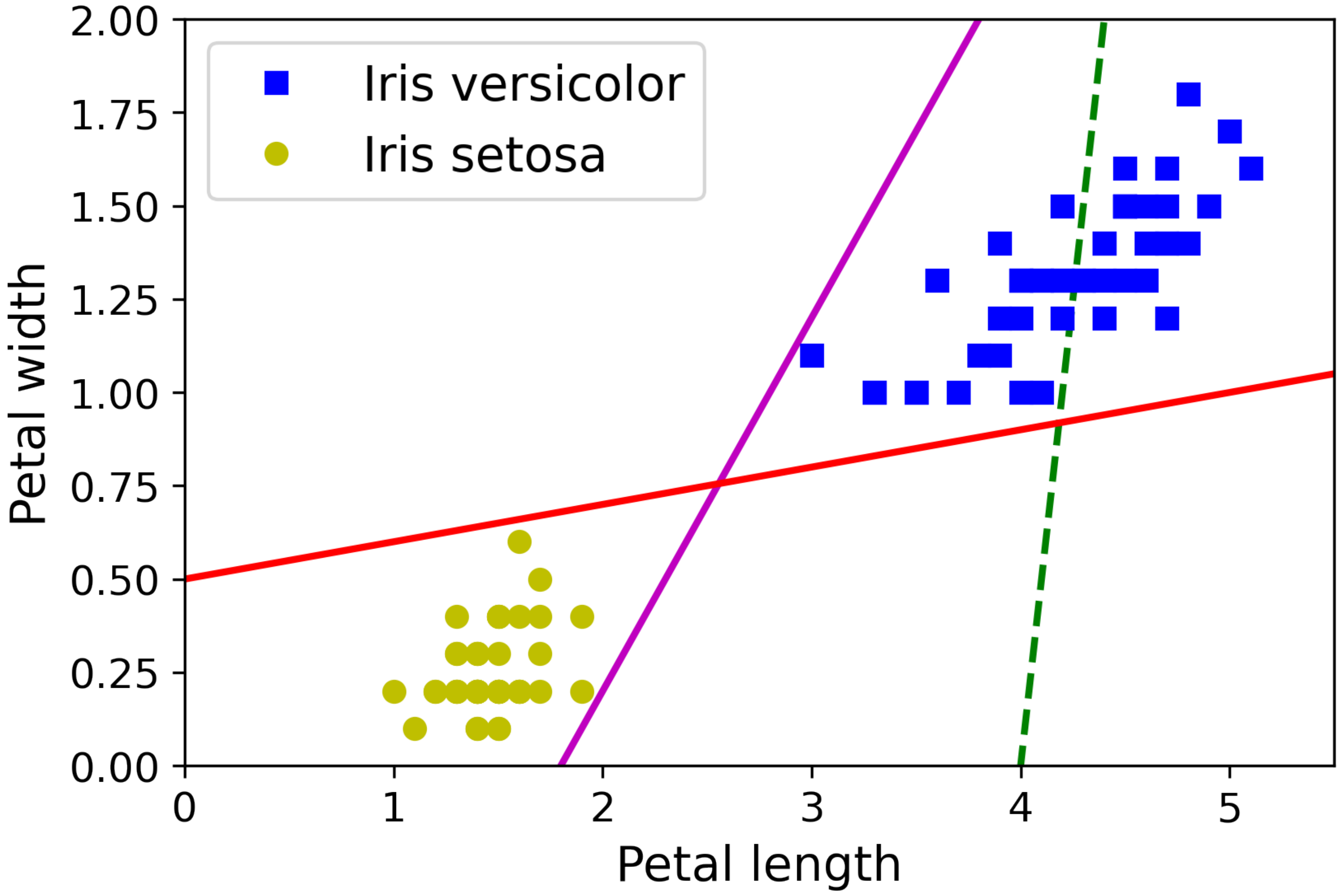
- A Support Vector Machine (SVM) is a very powerful and versatile Machine Learning model, capable of performing linear or nonlinear classification, regression, and even outlier detection.
- SVMs are particularly well suited for classification of complex but small-or medium-sized datasets.



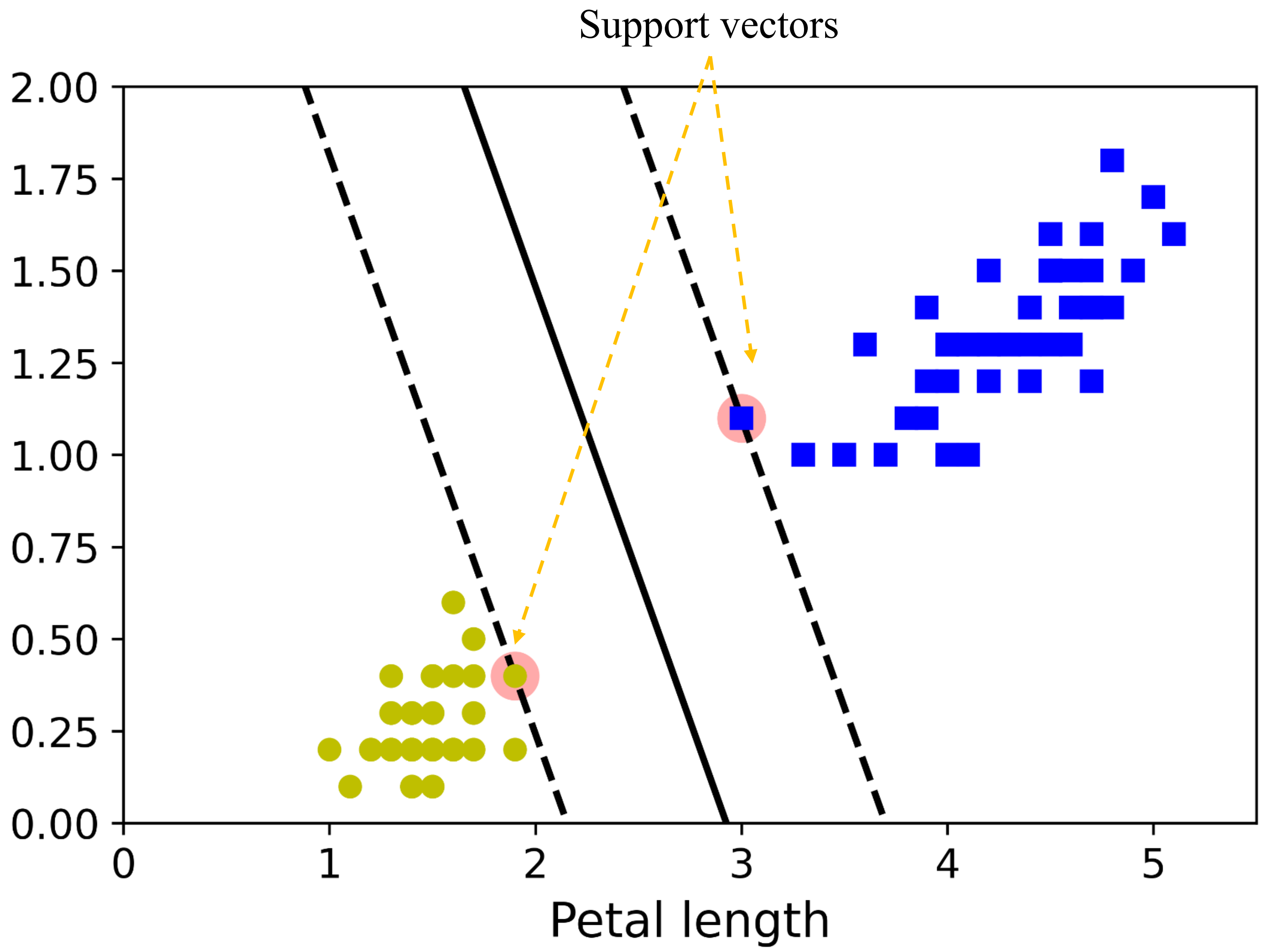
Linearly separable



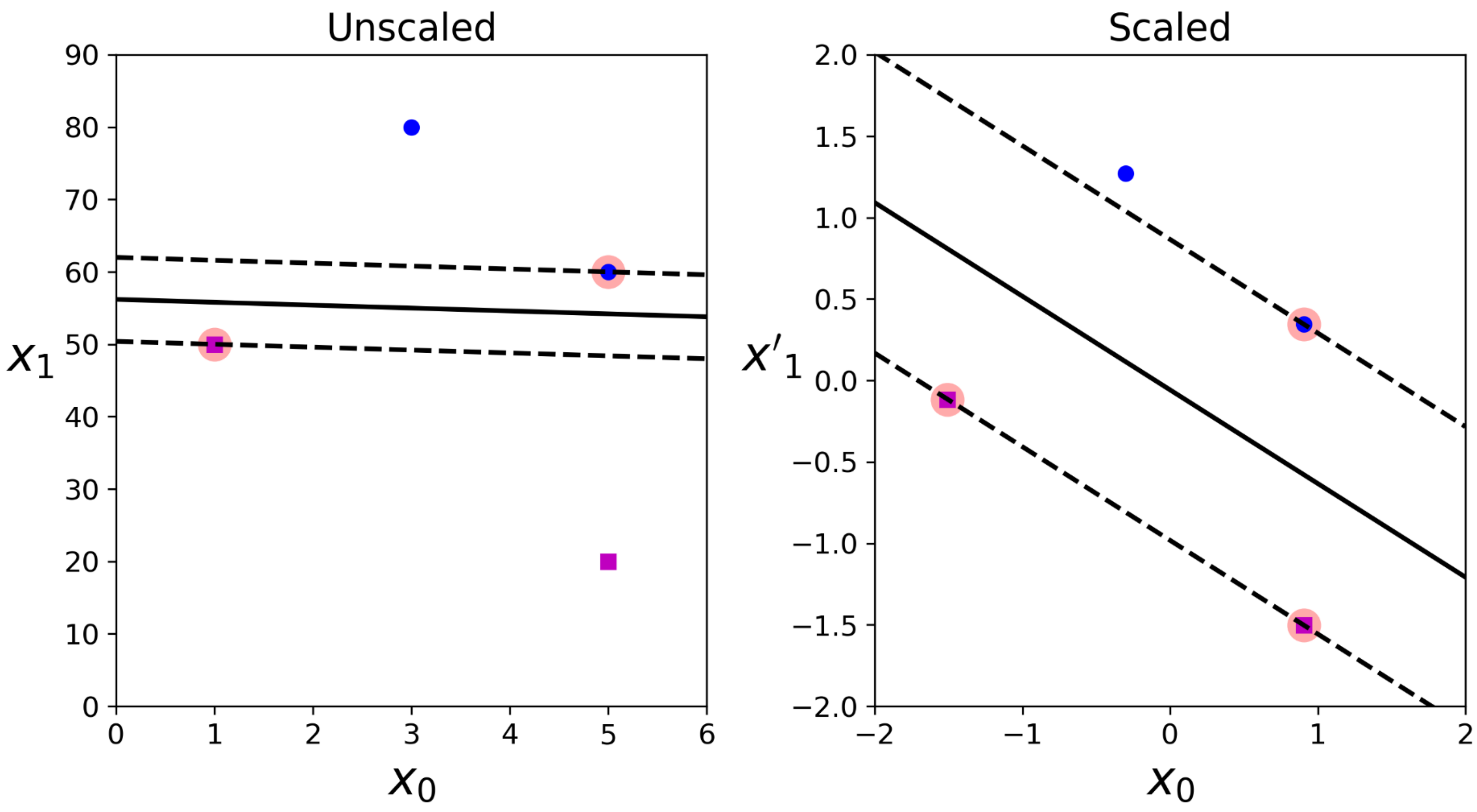
Which line?


Iris versicolor
Iris setosa
Petal width
Petal length
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
1
2
3
4
5

Support vectors



Feature scaling



Decision function: linear SVM

$$\hat{y} = \begin{cases} 0 & \text{if } \mathbf{w}^T\mathbf{x} + b < 0, \\ 1 & \text{if } \mathbf{w}^T\mathbf{x} + b \geq 0 \end{cases}$$

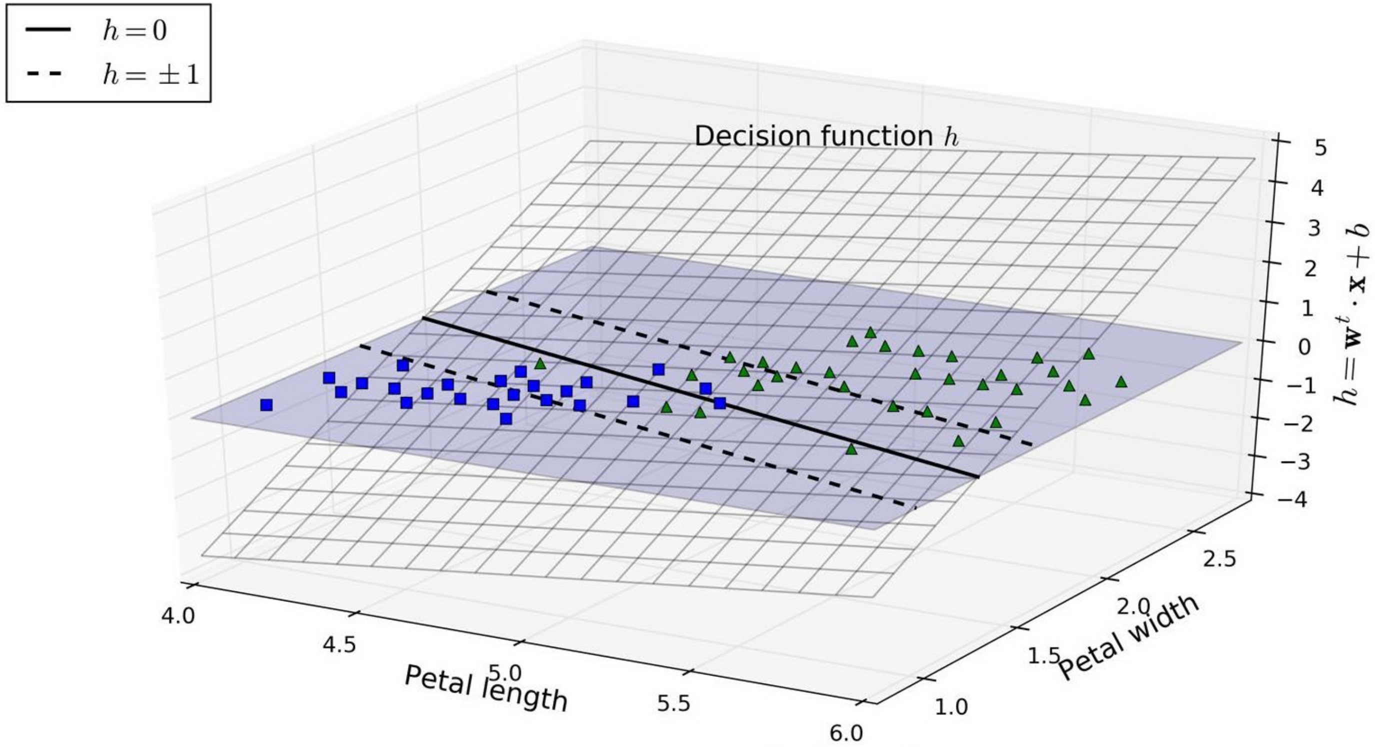


Figure 5-12. Decision function for the iris dataset

Training objective

- The smaller the weight vector, the larger the margin

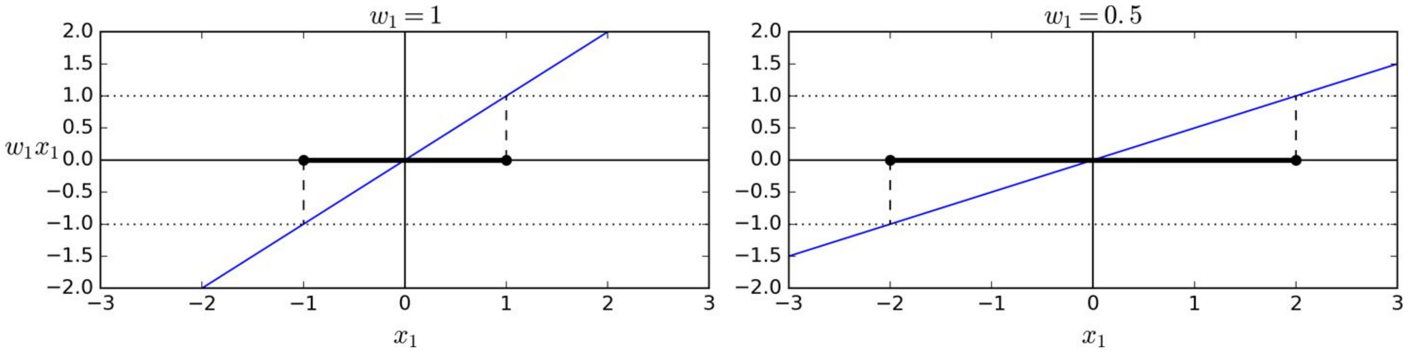


Figure 5-13. A smaller weight vector results in a larger margin

- Margin violation. Let $t^{(i)} = -1$ for negative instance (if $y^{(i)} = 0$) and $t^{(i)} = 1$ for positive instance (if $y^{(i)} = 1$)

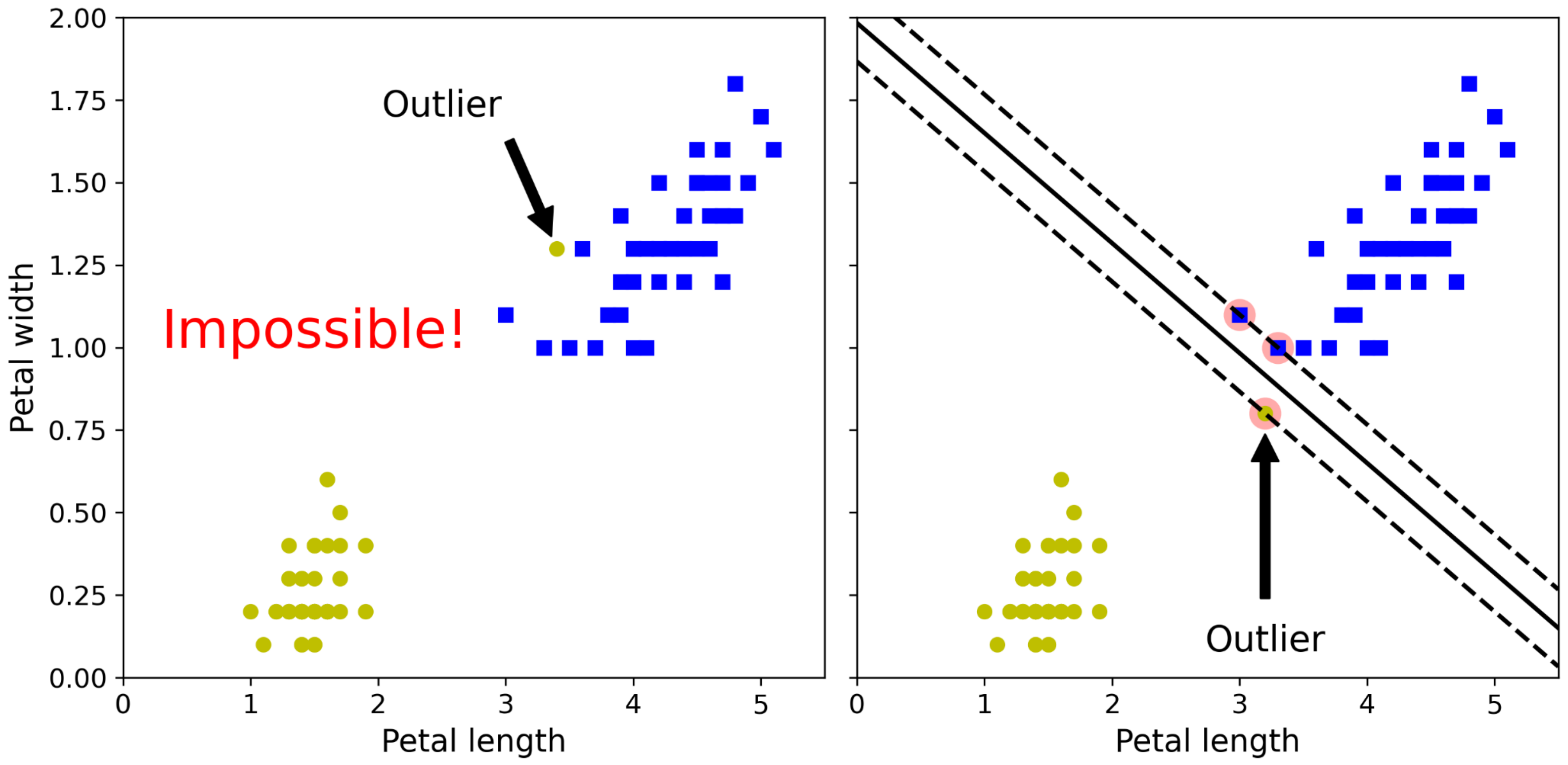
$$t^{(i)}\left(\mathbf{w}^T \mathbf{x}^{(i)} + b\right) \geq 1$$

Hard margin linear classifier
Objective function

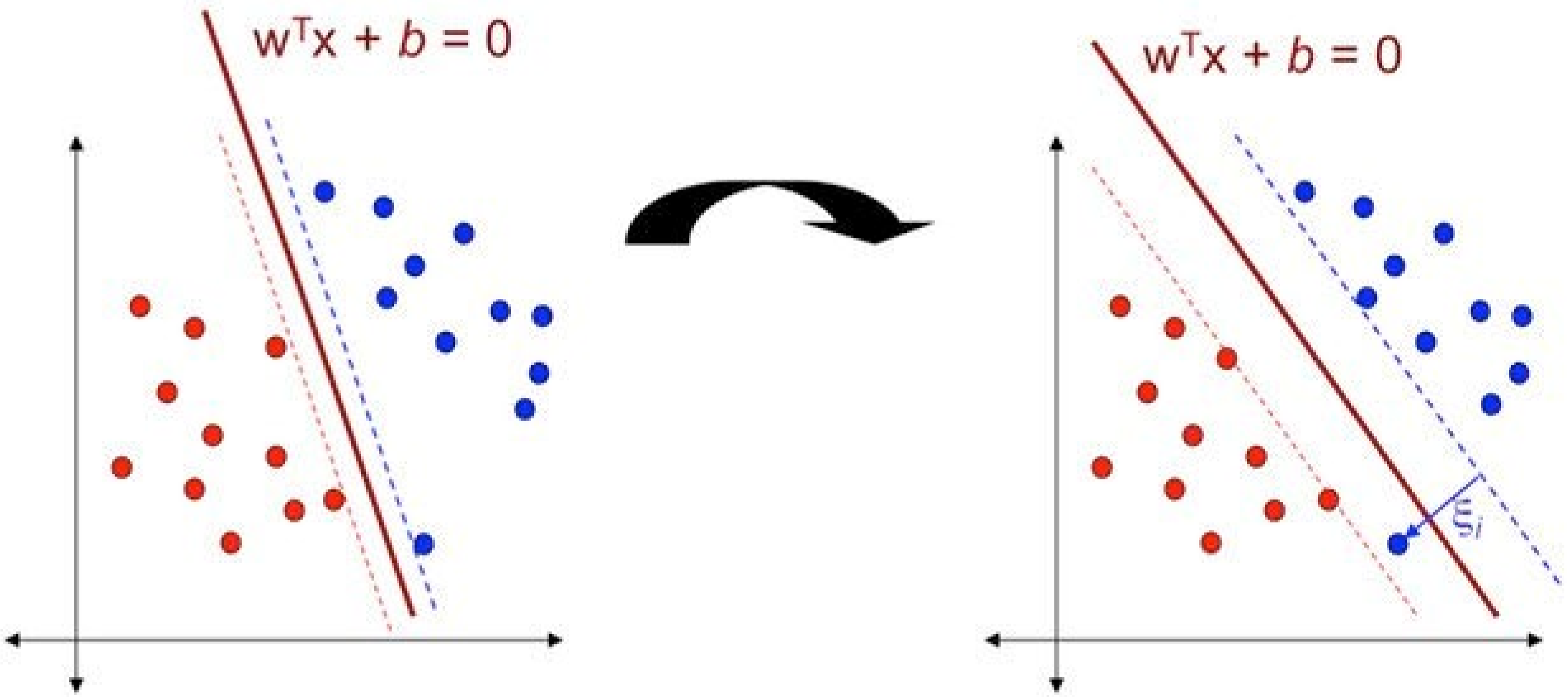
$$\underset{\mathbf{w},b}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

$$\text{subject to} \quad t^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 \quad \text{for } i = 1,\ 2,\ \cdots,\ m$$

Sensitivity to outliers



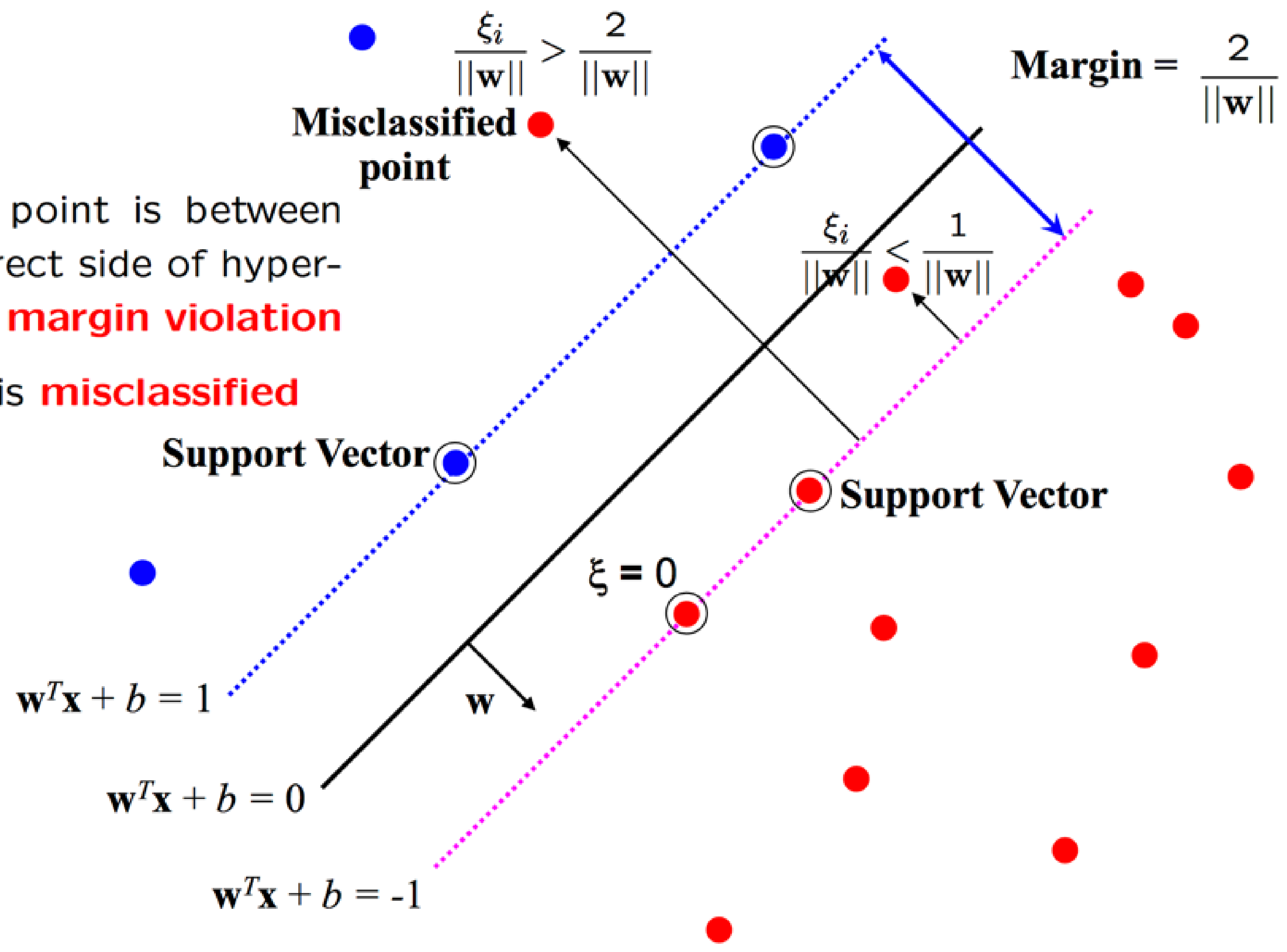
Soft margin classification



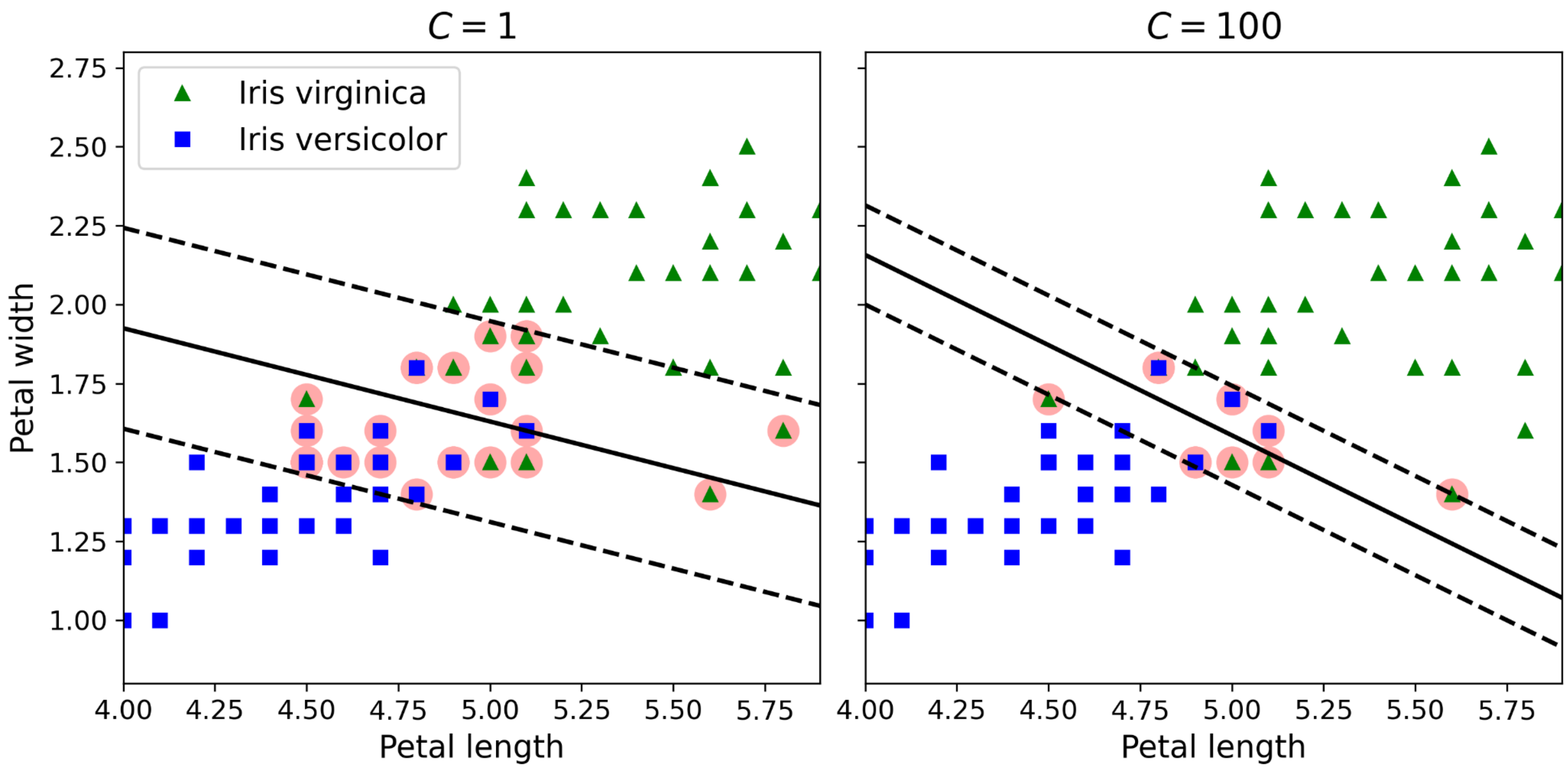
Error variables

$$\xi_i \geq 0$$

- for $0 < \xi \leq 1$ point is between margin and correct side of hyper-plane. This is a **margin violation**
- for $\xi > 1$ point is **misclassified**



Regularization via margin



Soft margin linear classifier Objective function

$$\underset{\mathbf{w},b,\zeta}{\text{minimize}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{m}\zeta^{(i)}$$

$$\text{subject to} \quad t^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1 - \zeta^{(i)} \quad \text{and} \quad \zeta^{(i)} \geq 0 \quad \text{for } i = 1,\, 2,\, \cdots,\, m$$

- C is a regularization parameter:
  - small C allows constraints to be easily ignored $\rightarrow$ large margin
  - large C makes constraints hard to ignore $\rightarrow$ narrow margin
  - $C = \infty$ enforces all constraints: hard margin

Solving hard and soft linear SVM Quadratic programming problem

Equation 5-5. Quadratic Programming problem

$$
\begin{aligned}
&\underset{\mathbf{p}}{\text{Minimize}} && \frac{1}{2}\mathbf{p}^T\mathbf{H}\mathbf{p} \quad + \quad \mathbf{f}^T\mathbf{p} \\
&\text{subject to} && \mathbf{A}\mathbf{p} \le \mathbf{b} \\
&\text{where} && \begin{cases}
\mathbf{p} & \text{is an } n_p\text{-dimensional vector } (n_p = \text{number of parameters}), \\
\mathbf{H} & \text{is an } n_p \times n_p \text{ matrix}, \\
\mathbf{f} & \text{is an } n_p\text{-dimensional vector}, \\
\mathbf{A} & \text{is an } n_c \times n_p \text{ matrix } (n_c = \text{number of constraints}), \\
\mathbf{b} & \text{is an } n_c\text{-dimensional vector.}
\end{cases}
\end{aligned}
$$

Dual problem
Lagrange multiplier

- Transform constrained optimization objective into an unconstrained one
- For example,

  minimize $f(x,y) = x^2+2y$

  subjective $3x+2y+1 = 0$ (equality constrain)
- Lagrangian

  minimize $g(x,y,\alpha) = f(x,y) - \alpha(3x+2y+1)$

Dual problem
Generalized Lagrangian

$\alpha^{(i)}$ as Karush-Kuhn-Tucker (KKT) multiplier

Equation C-1. Generalized Lagrangian for the hard margin problem

$$\mathscr{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{m}\alpha^{(i)}\left(t^{(i)}\left(\mathbf{w}^T\mathbf{x}^{(i)} + b\right) - 1\right)$$

$$\text{with} \quad \alpha^{(i)} \geq 0 \quad \text{for } i = 1, 2, \cdots, m$$

Partial derivative of generalized Lagrangian

Equation C-2. Partial derivatives of the generalized Lagrangian

$$\nabla_{\mathbf{w}} \mathscr{L}(\mathbf{w}, b, \alpha) = \mathbf{w} - \sum_{i=1}^{m} \alpha^{(i)} t^{(i)} \mathbf{x}^{(i)}$$

$$\frac{\partial}{\partial b} \mathscr{L}(\mathbf{w}, b, \alpha) = -\sum_{i=1}^{m} \alpha^{(i)} t^{(i)}$$

Stationary points, set to extrema

Equation C-3. Properties of the stationary points

$$\hat{\mathbf{w}} = \sum_{i=1}^{m} \hat{\alpha}^{(i)} t^{(i)} \mathbf{x}^{(i)}$$

$$\sum_{i=1}^{m} \hat{\alpha}^{(i)} t^{(i)} = 0$$

Substitute to generalized Lagrangian Loss function

Equation C-4. Dual form of the SVM problem

$$\mathcal{L}(\hat{\mathbf{w}}, \hat{b}, \alpha) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha^{(i)}\alpha^{(j)}t^{(i)}t^{(j)}\mathbf{x}^{(i)T}\mathbf{x}^{(j)} - \sum_{i=1}^{m}\alpha^{(i)}$$

$$\text{with} \quad \alpha^{(i)} \geq 0 \quad \text{for } i = 1, 2, \cdots, m$$

Dual problem
Objective function

Equation 5-6. Dual form of the linear SVM objective

$$\underset{\alpha}{\text{minimize}}\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha^{(i)}\alpha^{(j)}t^{(i)}t^{(j)}\mathbf{x}^{(i)T}\mathbf{x}^{(j)} \quad - \quad \sum_{i=1}^{m}\alpha^{(i)}$$

$$\text{subject to} \quad \alpha^{(i)} \geq 0 \quad \text{for } i = 1, 2, \cdots, m$$

Primal solution

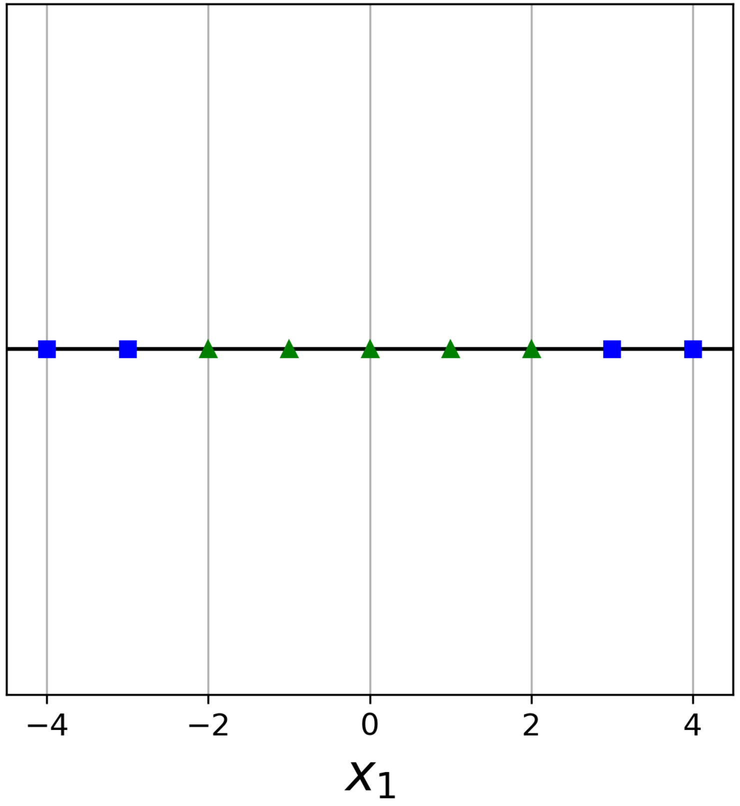
- After obtaining the solution of dual problem, $\alpha^{(i)}$, we can use them to compute the solution of primal problem as follows

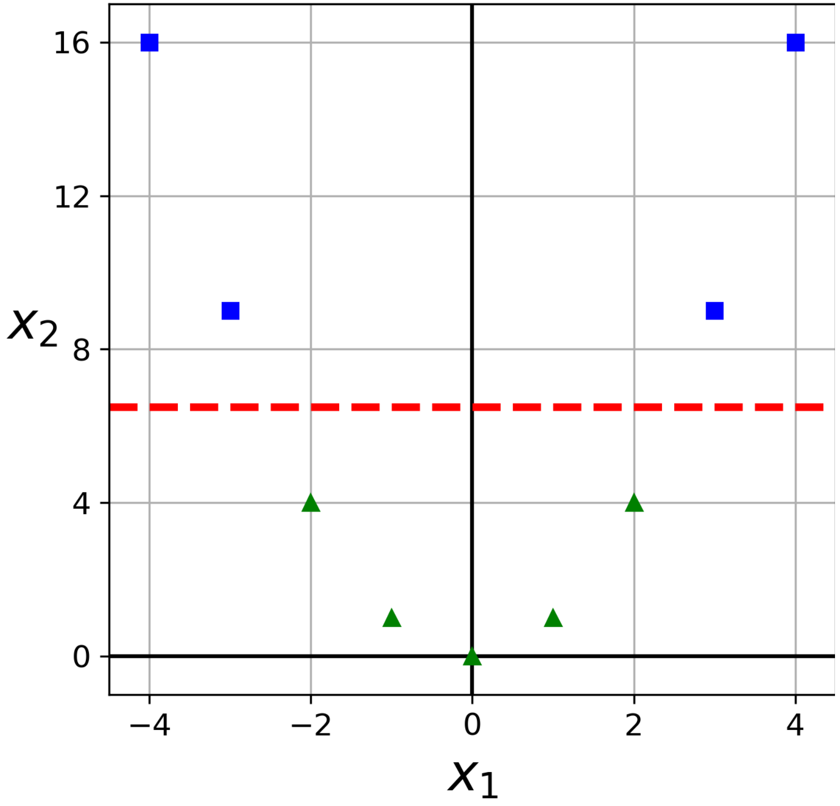
$$\hat{\mathbf{w}} = \sum_{i=1}^{m} \hat{\alpha}^{(i)} t^{(i)} \mathbf{x}^{(i)}$$

$$\hat{b} = \frac{1}{n_s} \sum_{\substack{i=1 \\ \hat{\alpha}^{(i)}>0}}^{m} \left(t^{(i)} - \hat{\mathbf{w}}^T \mathbf{x}^{(i)}\right)$$

Nonlinear SVM

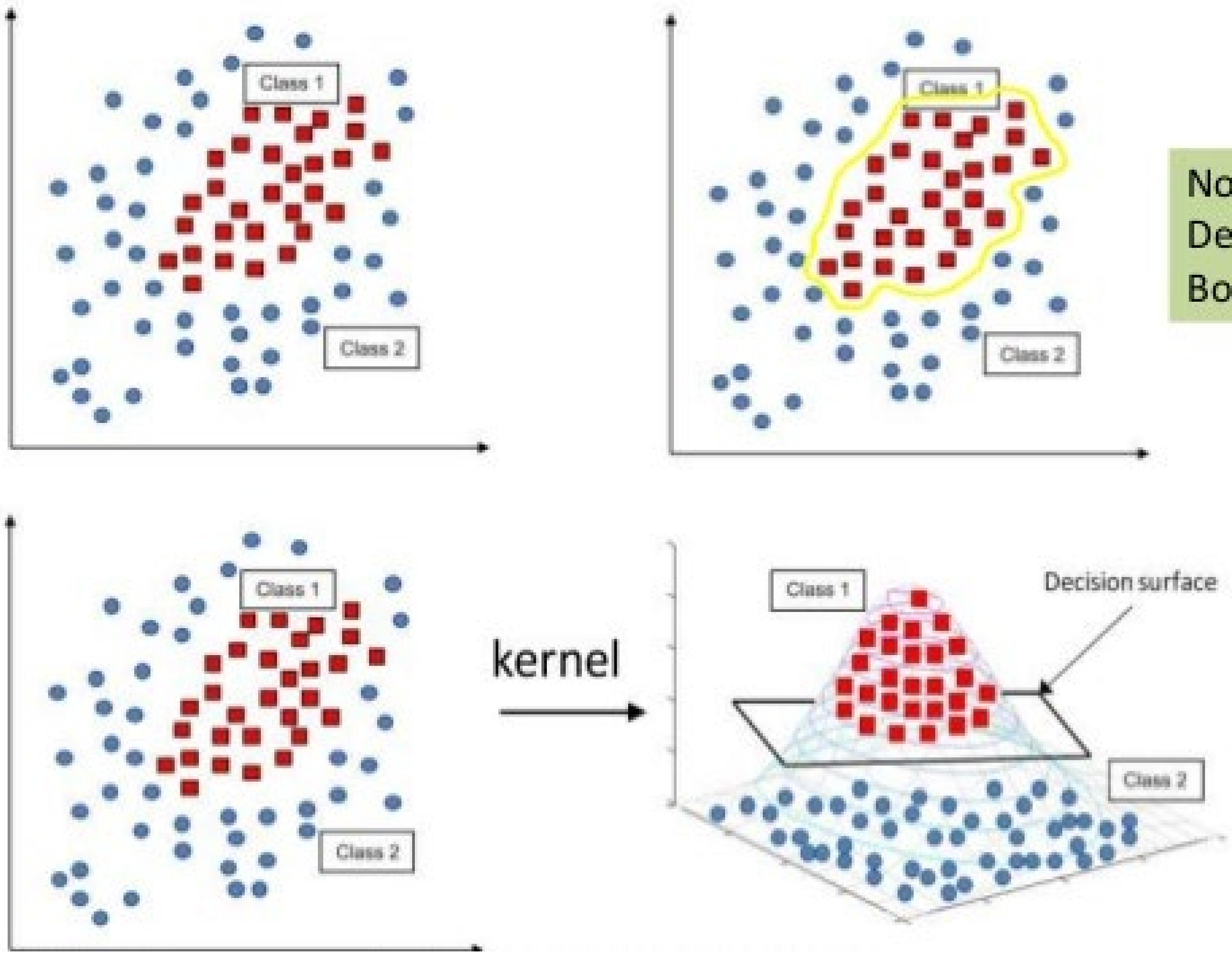
Adding features to make a dataset linearly separable





$$x_2 = (x_1)^2$$

SVM Kernel



Non Linear Decision Boundary

Kernel method

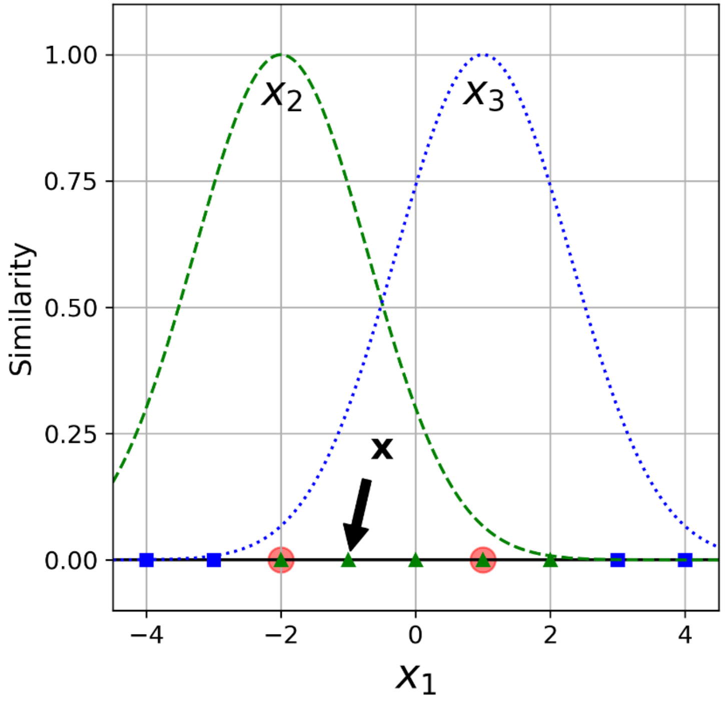
Adding similarity feature

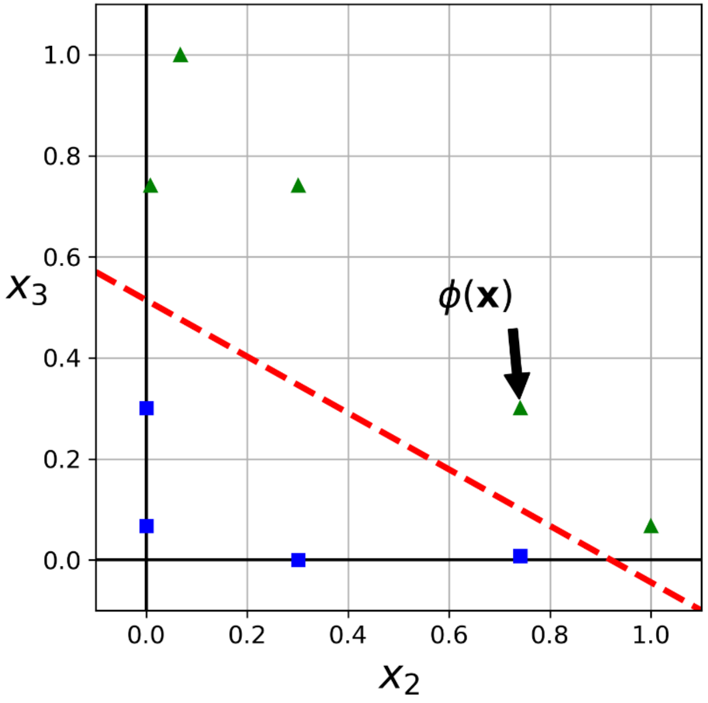
Spread

Mean

Equation 5-1. Gaussian RBF

$$\phi_\gamma(\mathbf{x}, \ell) = \exp\left(-\gamma \| \mathbf{x} - \ell \|^2\right)$$

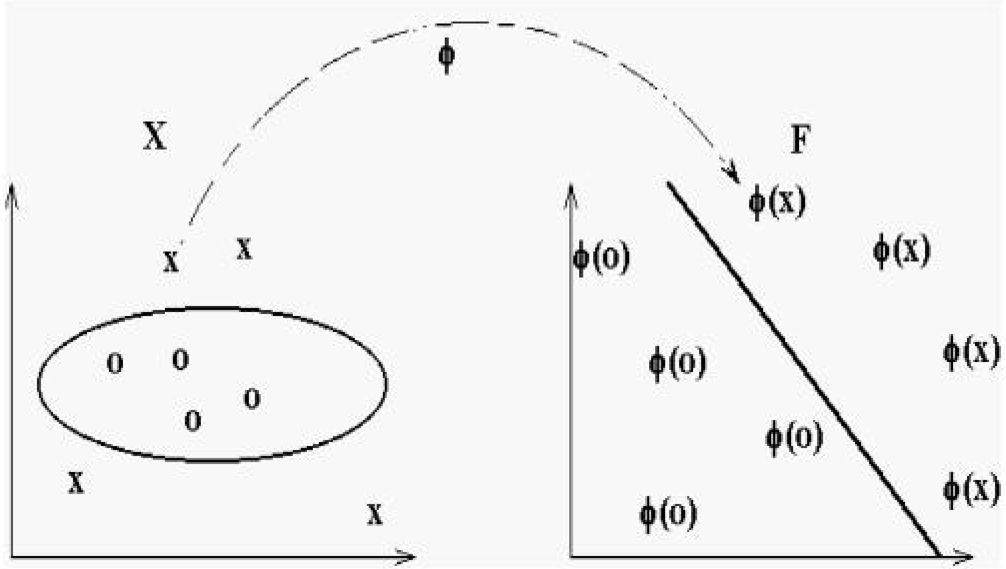




Kernelized SVM
Second-degree polynomial

Equation 5-8. Second-degree polynomial mapping

$$\phi(\mathbf{x}) = \phi\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = \begin{pmatrix} {x_1}^2 \\ \sqrt{2}\, x_1 x_2 \\ {x_2}^2 \end{pmatrix}$$



Kernel trick

Equation 5-9. Kernel trick for a 2nd-degree polynomial mapping

$$
\begin{aligned}
\phi(\mathbf{a})^T\phi(\mathbf{b}) &= \begin{pmatrix} {a_1}^2 \\ \sqrt{2}\, a_1 a_2 \\ {a_2}^2 \end{pmatrix}^T \begin{pmatrix} {b_1}^2 \\ \sqrt{2}\, b_1 b_2 \\ {b_2}^2 \end{pmatrix} = {a_1}^2 {b_1}^2 + 2 a_1 b_1 a_2 b_2 + {a_2}^2 {b_2}^2 \\
&= (a_1 b_1 + a_2 b_2)^2 = \left(\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}^T \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right)^2 = (\mathbf{a}^T\mathbf{b})^2
\end{aligned}
$$

Recall the objective function

Equation 5-6. Dual form of the linear SVM objective

$$\underset{\alpha}{\text{minimize}}\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha^{(i)}\alpha^{(j)}t^{(i)}t^{(j)}\underline{\mathbf{x}^{(i)^T}\mathbf{x}^{(j)}} \quad - \quad \sum_{i=1}^{m}\alpha^{(i)}$$

$$\text{subject to} \quad \alpha^{(i)} \geq 0 \quad \text{for } i = 1, 2, \cdots, m$$

Replacing with $\phi(\mathbf{x}^{(i)})^{T}\phi(\mathbf{x}^{(j)})$

For 2-nd degree polynomial kernel, $\phi(\mathbf{x}^{(i)})^{T}\phi(\mathbf{x}^{(j)}) = (\mathbf{x}^{(i)T}\mathbf{x}^{(j)})^{2}$

Common kernel

Equation 5-10. Common kernels

$$\text{Linear:} \quad K(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T\mathbf{b}$$

$$\text{Polynomial:} \quad K(\mathbf{a}, \mathbf{b}) = \left(\gamma \mathbf{a}^T\mathbf{b} + r\right)^d$$

$$\text{Gaussian RBF:} \quad K(\mathbf{a}, \mathbf{b}) = \exp\left(-\gamma \left\| \mathbf{a} - \mathbf{b} \right\|^2\right)$$

$$\text{Sigmoid:} \quad K(\mathbf{a}, \mathbf{b}) = \tanh\left(\gamma \mathbf{a}^T\mathbf{b} + r\right)$$

Prediction with kernelized SVM

Equation 5-11. Making predictions with a kernelized SVM

$$
\begin{aligned}
h_{\hat{\mathbf{w}},\hat{b}}\left(\phi\left(\mathbf{x}^{(n)}\right)\right) &= \hat{\mathbf{w}}^T \phi\left(\mathbf{x}^{(n)}\right) + \hat{b} = \left(\sum_{i=1}^{m} \hat{\alpha}^{(i)} t^{(i)} \phi\left(\mathbf{x}^{(i)}\right)\right)^T \phi\left(\mathbf{x}^{(n)}\right) + \hat{b} \\
&= \sum_{i=1}^{m} \hat{\alpha}^{(i)} t^{(i)} \left(\phi\left(\mathbf{x}^{(i)}\right)^T \phi\left(\mathbf{x}^{(n)}\right)\right) + \hat{b} \\
&= \sum_{\substack{i=1 \\ \hat{\alpha}^{(i)}>0}}^{m} \hat{\alpha}^{(i)} t^{(i)} K\left(\mathbf{x}^{(i)}, \mathbf{x}^{(n)}\right) + \hat{b}
\end{aligned}
$$

No need of **w**!!

Bias term for kernel trick

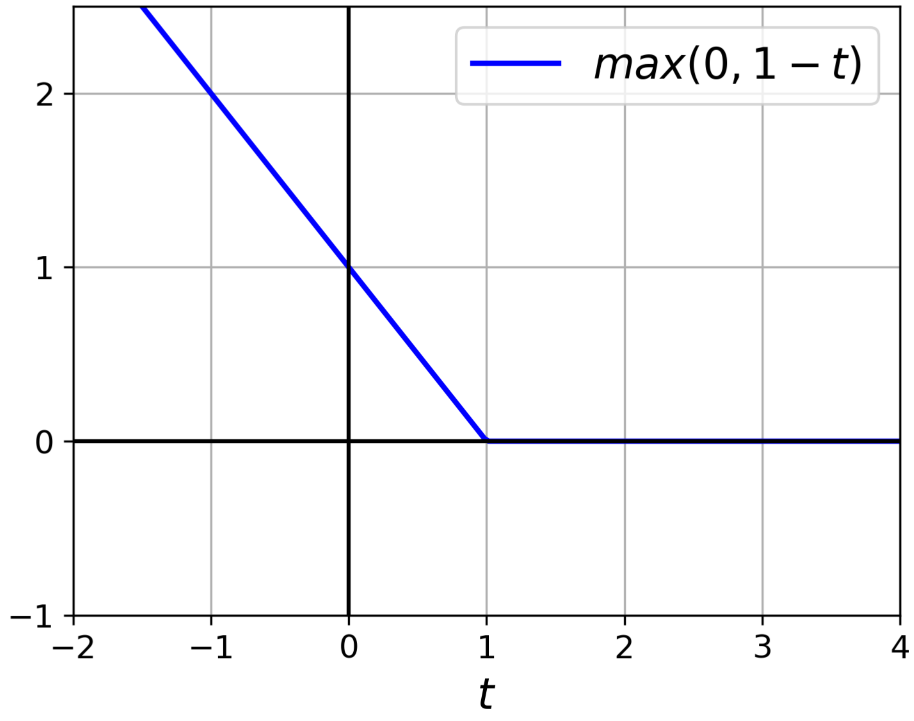
Equation 5-12. Computing the bias term using the kernel trick

$$\hat{b} = \frac{1}{n_s} \sum_{\substack{i=1 \\ \hat{\alpha}^{(i)}>0}}^{m} \left(t^{(i)} - \hat{\mathbf{w}}^T \phi(\mathbf{x}^{(i)})\right) = \frac{1}{n_s} \sum_{\substack{i=1 \\ \hat{\alpha}^{(i)}>0}}^{m} \left(t^{(i)} - \left(\sum_{j=1}^{m} \hat{\alpha}^{(j)} t^{(j)} \phi(\mathbf{x}^{(j)})\right)^T \phi(\mathbf{x}^{(i)})\right)$$

$$= \frac{1}{n_s} \sum_{\substack{i=1 \\ \hat{\alpha}^{(i)}>0}}^{m} \left(t^{(i)} - \sum_{\substack{j=1 \\ \hat{\alpha}^{(j)}>0}}^{m} \hat{\alpha}^{(j)} t^{(j)} K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})\right)$$

Hinge Loss

- Derivative equals to -1 when $t^{(i)} < 1$
- This makes it suitable for using with Gradient Descent in online learning



$$J(\mathbf{w}, b) = \frac{1}{2}\mathbf{w}^T\mathbf{w} \quad + \quad C\sum_{i=1}^{m} max(0,\ 1 - t^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b))$$

End of Lecture 4

Question?