Support Vector Machine

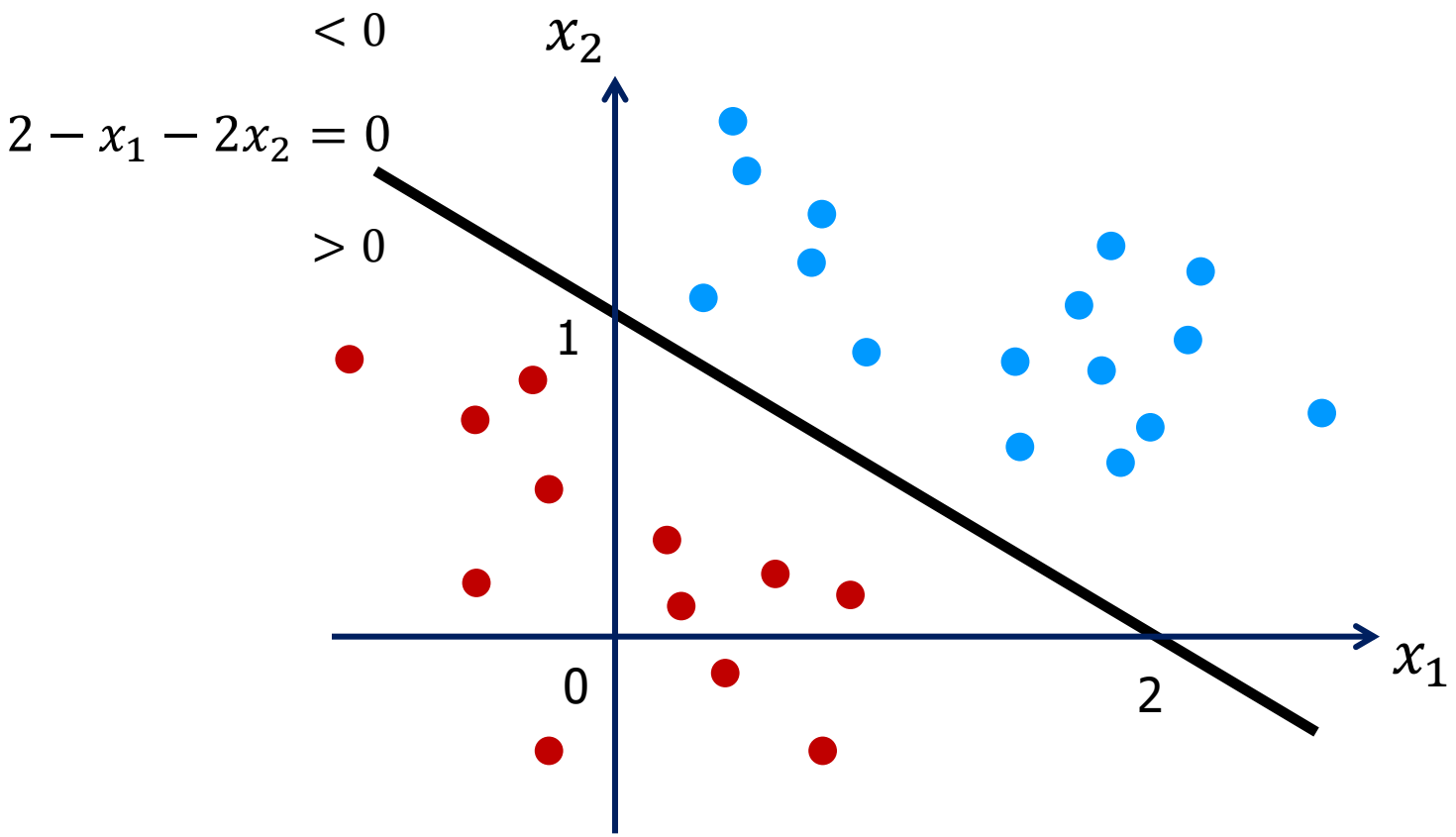
Zhiyao Duan
Professor of ECE and CS
University of Rochester

Some figures are copied from the following book

- **Géron -** Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition,* O'Reilly, 2022.

Some slides are adapted from Neil Zhang's guest lecture slides

Linear Classification

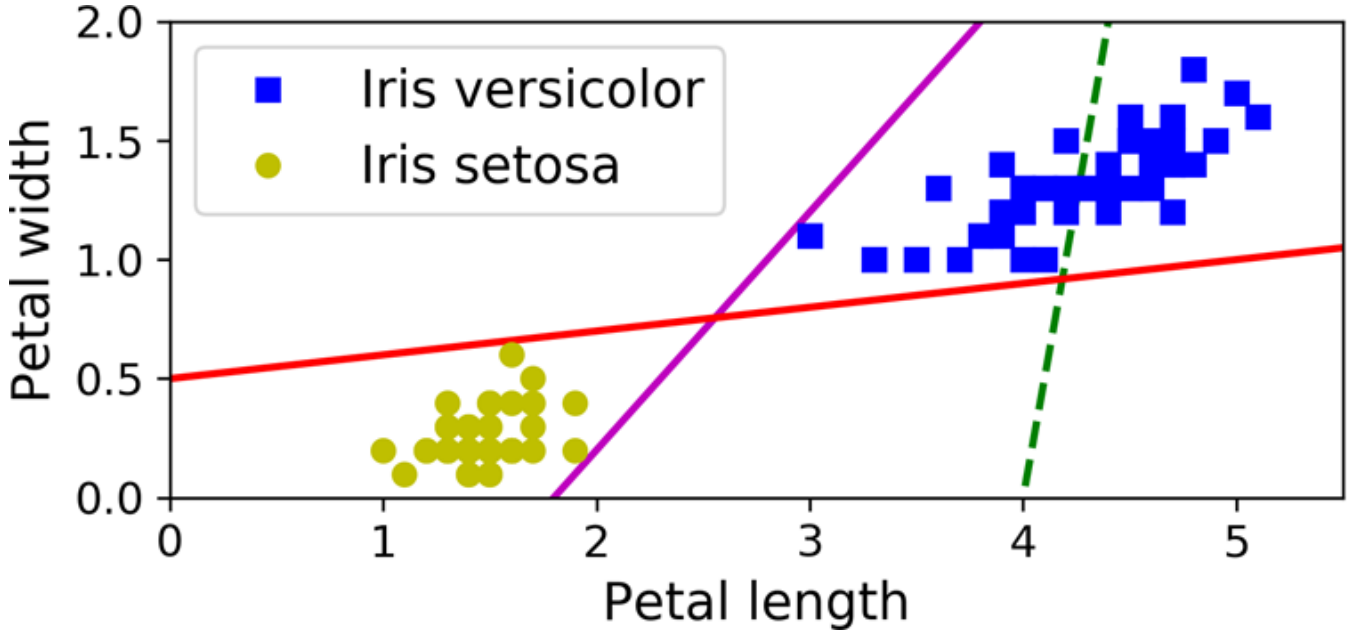


- A linear boundary is a hyperplane, represented by a linear equation

$$\boldsymbol{w}^T\boldsymbol{x} + b = 0$$

where $\boldsymbol{w} = [w_1, \dots, w_d]^T$ and $\boldsymbol{x} = [x_1, \dots, x_d]^T$. (Note the slight notation difference from our previous lecture)

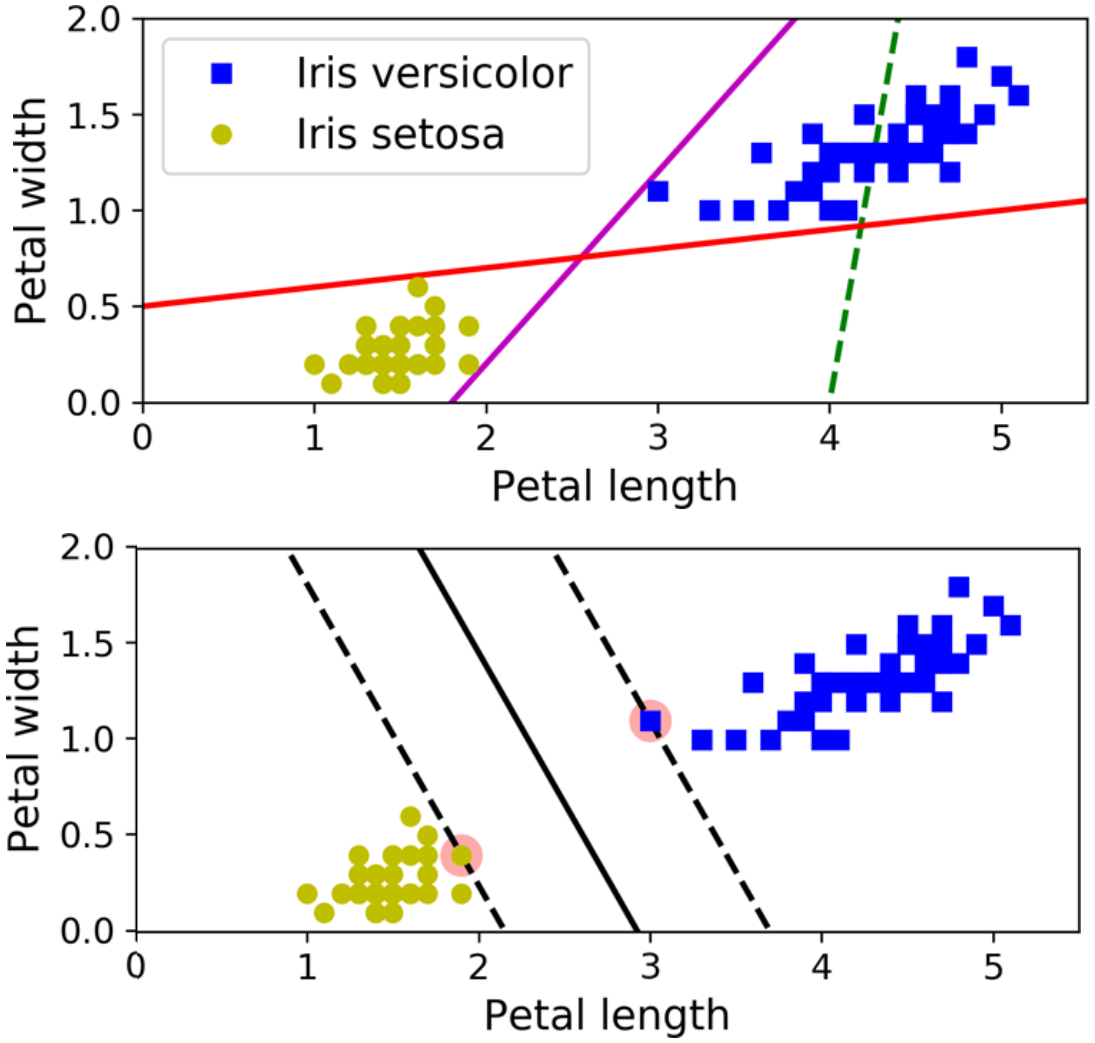
Linearly Separable Cases



- Which hyperplane is a better classifier?
- Are there better hyperplanes?

Maximize Margin

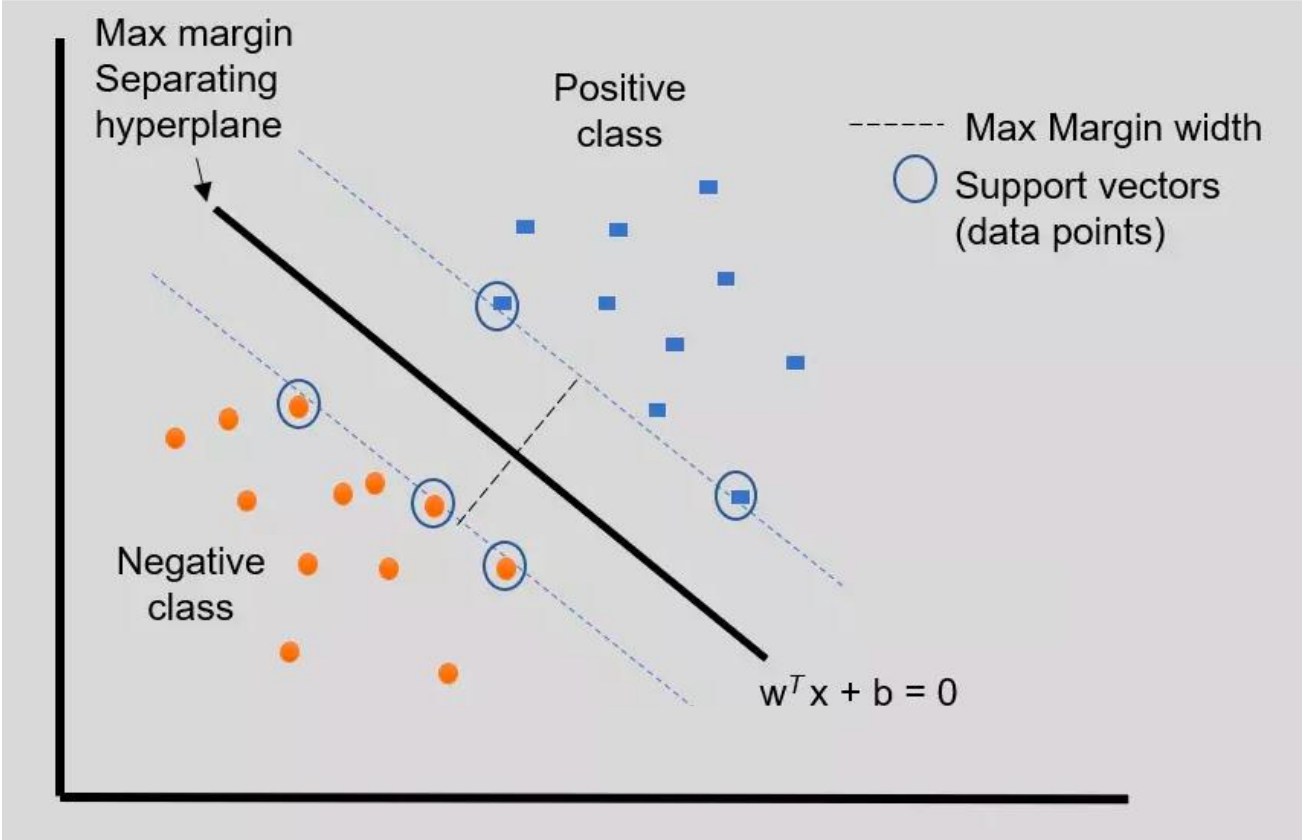
- Among all hyperplanes that separate the training data, find the one that is the furthest from the closest training example(s) for both classes
- The closes training examples are called support vectors.
- This makes a safe margin between the two classes, hence is likely to generalize to unseen test examples well



(Figure from Géron Figure 5-1)

Let's calculate the margin

- Positive and negative support vectors should be equally far from the hyperplane
  - $\boldsymbol{w}^T \boldsymbol{x}^{\text{pos_sv}} + b = \boldsymbol{w}^T \boldsymbol{x}^{\text{neg_sv}} + b$
- Scaling $\boldsymbol{w}$ and b together does not change the hyperplane!
  - So we can scale them such that $\boldsymbol{w}^T \boldsymbol{x} + b = \pm 1$ for support vectors
- Projection of $\boldsymbol{x}$ to $\boldsymbol{w}$ is $\frac{\boldsymbol{w}^T \boldsymbol{x}}{\|\boldsymbol{w}\|^2} \boldsymbol{w}$
  - For $\boldsymbol{x}$ on the hyperplane, the projection is $\frac{-b}{\|\boldsymbol{w}\|^2} \boldsymbol{w}$
- Distance from $\boldsymbol{x}$ to hyperplane: $\frac{|\boldsymbol{w}^T \boldsymbol{x}+b|}{\|\boldsymbol{w}\|^2} \|\boldsymbol{w}\| = \frac{|\boldsymbol{w}^T \boldsymbol{x}+b|}{\|\boldsymbol{w}\|}$
  - For support vectors, this distance is $\frac{1}{\|\boldsymbol{w}\|}$
  - So margin is $\frac{2}{\|\boldsymbol{w}\|}$



(Figure from Support Vector Machine (SVM) basics and implementation in Python)

Objective Function

- Maximizing margin $\frac{2}{\|\boldsymbol{w}\|}$ is equivalent to minimizing $\frac{1}{2}\|\boldsymbol{w}\|^2 = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}$
- Constraints: all training examples are outside the safe margin, i.e.,

$$y^{(n)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(n)} + b\right) \geq 1,\ \forall n \in \{1, \dots, N\}$$

$$\min_{\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \qquad \text{(Primal Problem)}$$

$$\text{subject to:} \quad y^{(n)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(n)} + b\right) \geq 1,\ \forall n \in \{1, \dots, N\}$$

- Minimizing Lagrangian function (primal function):

$$L_p(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \sum_{n=1}^{N} \alpha_n \left(1 - y^{(n)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(n)} + b\right)\right)$$

where Lagrangian multipliers $\alpha_n \geq 0$, and when $y^{(n)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(n)} + b\right) = 1$, then $\alpha_n > 0$.

Dual Problem

- Dual function: $L_d(\boldsymbol{\alpha}) = \min_{\boldsymbol{w},b} L_p(\boldsymbol{w}, b, \boldsymbol{\alpha})$ (Skipping derivation)

$$= \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y^{(n)} y^{(m)} \left(\boldsymbol{x}^{(n)^T} \boldsymbol{x}^{(m)} \right) = \mathbf{1}^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha}$$

where $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_N]^T$, $\boldsymbol{Q} = \begin{bmatrix} y^{(1)} y^{(1)} \boldsymbol{x}^{(1)^T} \boldsymbol{x}^{(1)} & \dots & y^{(1)} y^{(N)} \boldsymbol{x}^{(1)^T} \boldsymbol{x}^{(N)} \\ \vdots & \ddots & \vdots \\ y^{(N)} y^{(1)} \boldsymbol{x}^{(N)^T} \boldsymbol{x}^{(1)} & \dots & y^{(N)} y^{(N)} \boldsymbol{x}^{(N)^T} \boldsymbol{x}^{(N)} \end{bmatrix}$ is an $N \times N$ symmetric matrix of inner products of training data!

$$\max_{\boldsymbol{\alpha}} \mathbf{1}^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{Q} \boldsymbol{\alpha} \quad \text{(Dual Problem)}$$

$$\text{Subject to: } \boldsymbol{y}^T \boldsymbol{\alpha} = 0$$

SVM Solution

- Dual problem is quadratic programming, and it solves α_n for all n
  - The n-th training example is a support vector if $\alpha_n > 0$
  - The n-th training example is not a support vector if $\alpha_n = 0$
- Then solve $\boldsymbol{w}$ and b

$$\boldsymbol{w} = \sum_{n=1}^{N} \alpha_n y^{(n)} \boldsymbol{x}^{(n)} = \sum_{\boldsymbol{x}^{(n)} \in SV} \alpha_n y^{(n)} \boldsymbol{x}^{(n)}$$

$$b = y^{(m)} - \boldsymbol{w}^T \boldsymbol{x}^{(m)}, \forall \boldsymbol{x}^{(m)} \in SV \text{ (the set of support vectors)}$$

Take the average:

$$b = \frac{1}{|SV|} \sum_{\boldsymbol{x}^{(m)} \in SV} \left(y^{(m)} - \sum_{\boldsymbol{x}^{(n)} \in SV} \alpha_n y^{(n)} \boldsymbol{x}^{(n)^T} \boldsymbol{x}^{(m)} \right)$$

- The learned classifier only depends on support vectors!

Inference Time

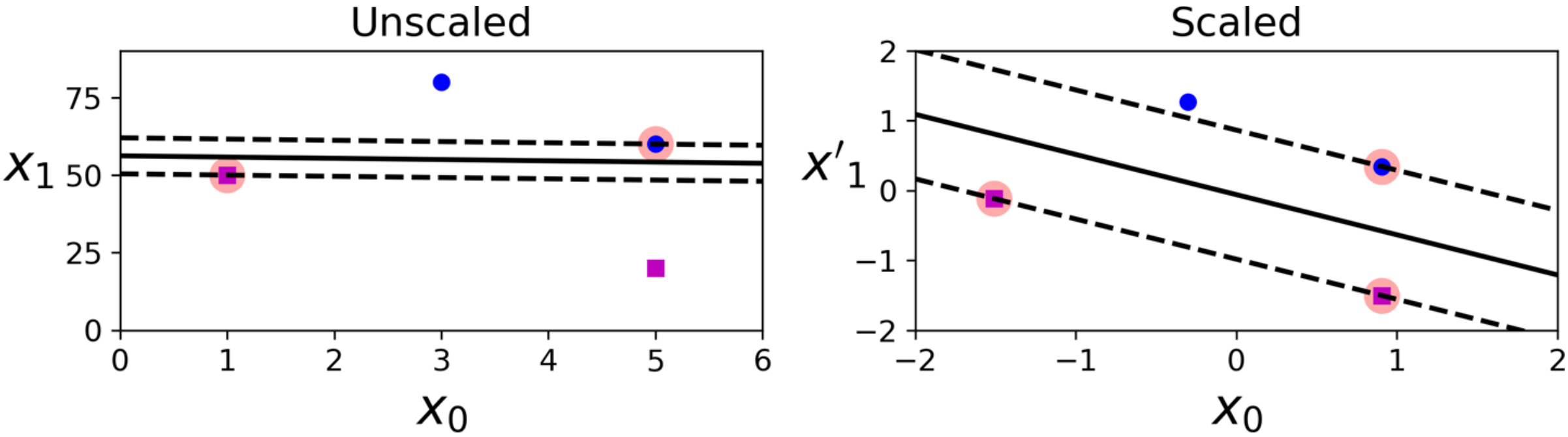
- For test data $\boldsymbol{x}$
- Compute

$$\boldsymbol{w}^T\boldsymbol{x} + b$$

$$= \sum_{\boldsymbol{x}^{(n)} \in SV} \alpha_n y^{(n)} \boldsymbol{x}^{(n)^T} \boldsymbol{x} + \frac{1}{|SV|} \sum_{\boldsymbol{x}^{(m)} \in SV} \left(y^{(m)} - \sum_{\boldsymbol{x}^{(n)} \in SV} \alpha_n y^{(n)} \boldsymbol{x}^{(n)^T} \boldsymbol{x}^{(m)} \right)$$

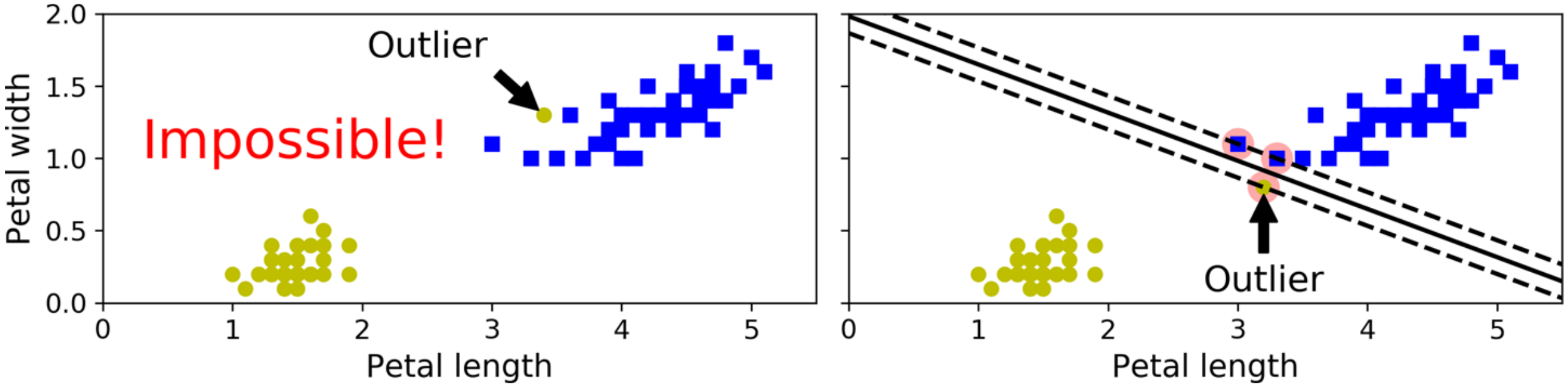
- It's all about inner products
  - Between support vectors $\boldsymbol{x}^{(n)}$ and test data $\boldsymbol{x}$
  - Between support vectors $\boldsymbol{x}^{(n)}$ and $\boldsymbol{x}^{(m)}$
- Feature vectors do not appear alone!

Sensitive to Feature Scales



(Figure from Géron Figure 5-2)

Sensitive to Outliers



(Figure from Géron Figure 5-3)

What if the problem is not linearly separable?

- Let's relax the margin requirement!
- Just implement the margin idea softly

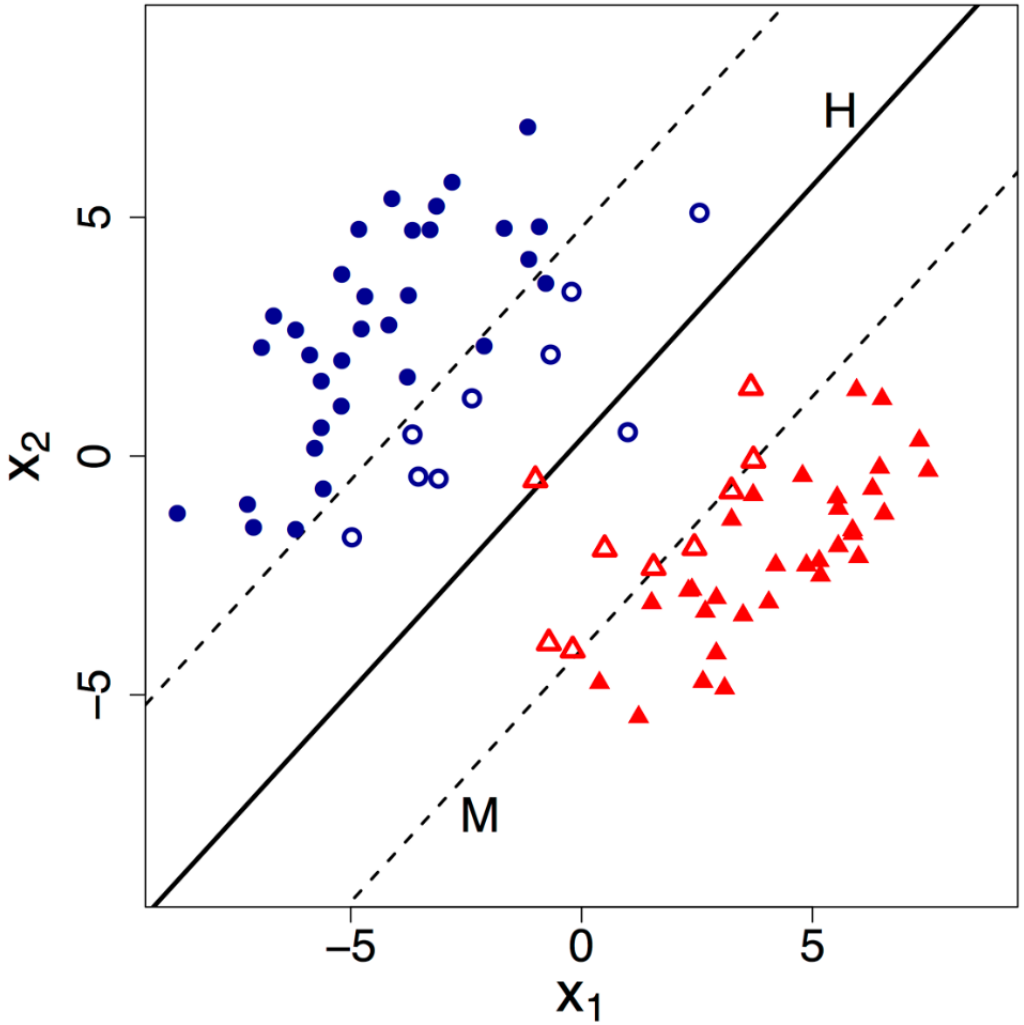
- Soft Margin Classification
  - Add slack variables ϵ_n

$$\min_{\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{n=1}^{N}\epsilon_n$$

$$\text{subject to:} \quad y^{(n)}\left(\boldsymbol{w}^T\boldsymbol{x}^{(n)} + b\right) + \epsilon_n \geq 1$$

$$\epsilon_n \geq 0, \ \forall n \in \{1, \dots, N\},$$

  - C controls how much relaxation we want

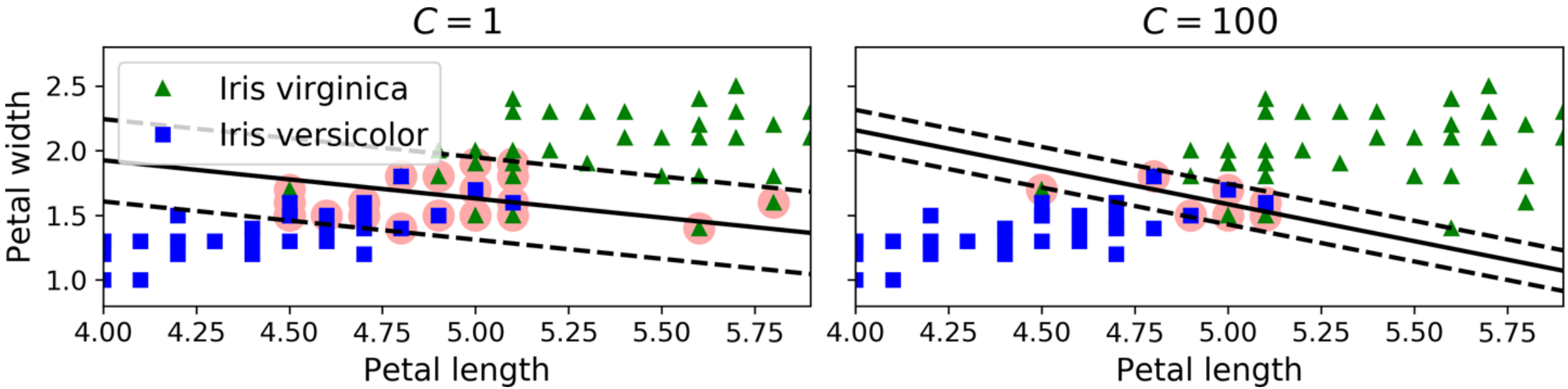


(Figure from Using Support Vector Machines for Survey Research | Published in Survey Practice)

Soft Margin Classification

more margin violations

less margin violations



(Figure from Géron figure 5-4)

Summary

- Linear SVM finds the hyperplane that leads to the maximal margin between the two classes

- Soft margin classification allows margin violations, and this is controlled by hyperparameter C

- Scikit implementation:
  - https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html