Statistical Machine Learning

5주차

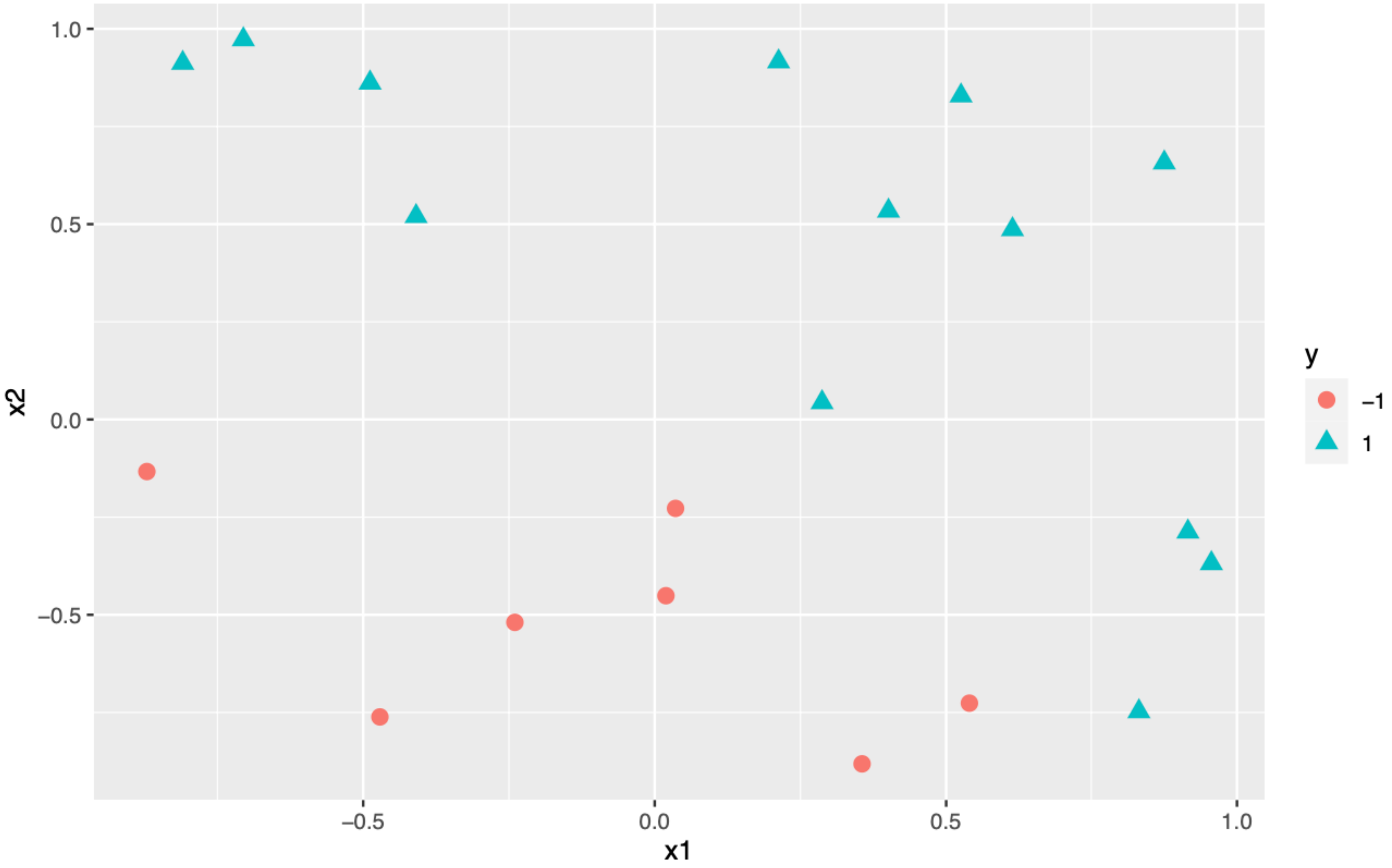
담당: 15기 김지호

1. Linear SVM
2. Kernel SVM
3. Decision Tree

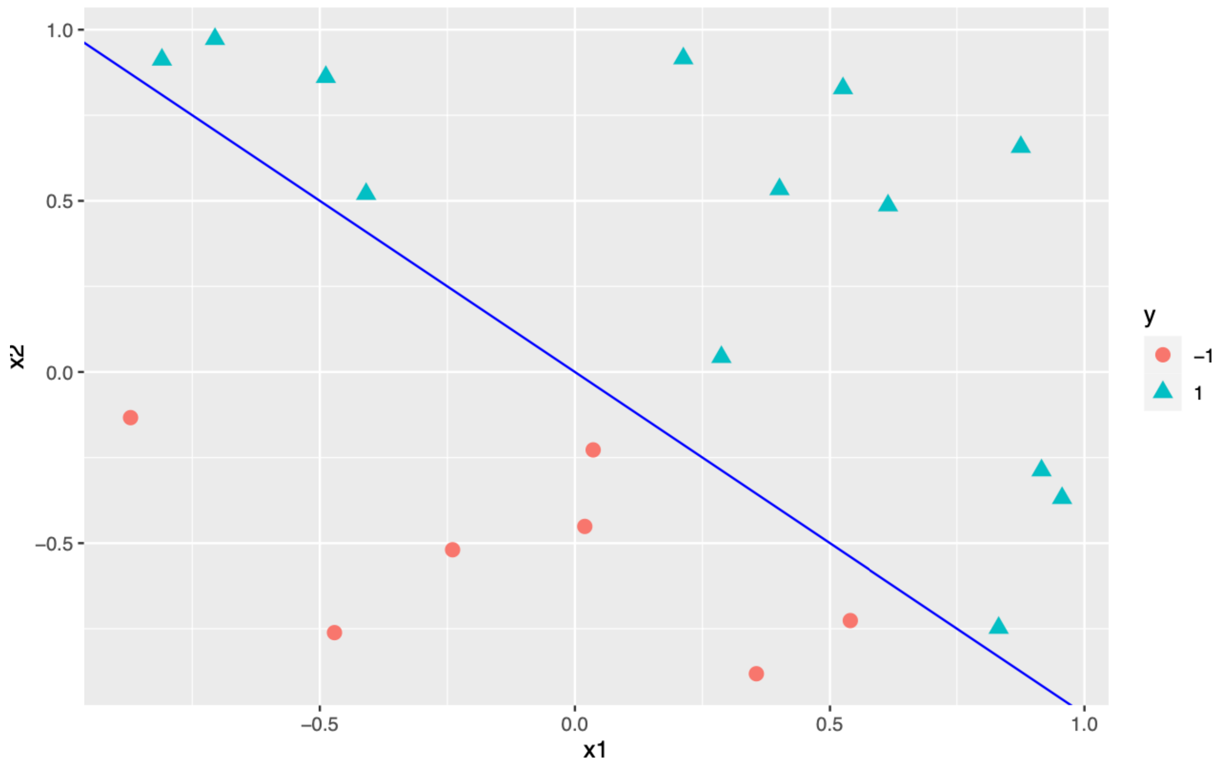
Support Vector Machine

- Classification

- Consider a simple (linearly separable) example.



- The line is called the classification/decision boundary or separating hyperplane.



- Decision boundary

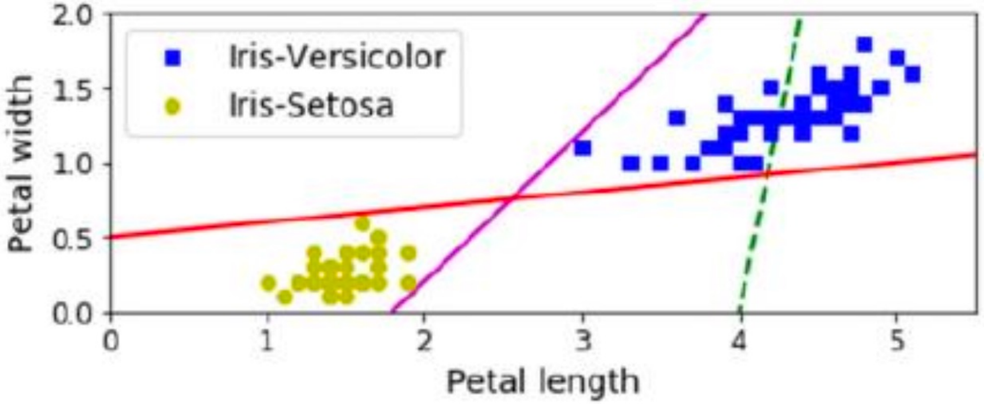
  $f(\mathbf{x}) = \beta_0 + \boldsymbol{\beta}^T \mathbf{x} = 0$

- $y \in \{-1, 1\}$

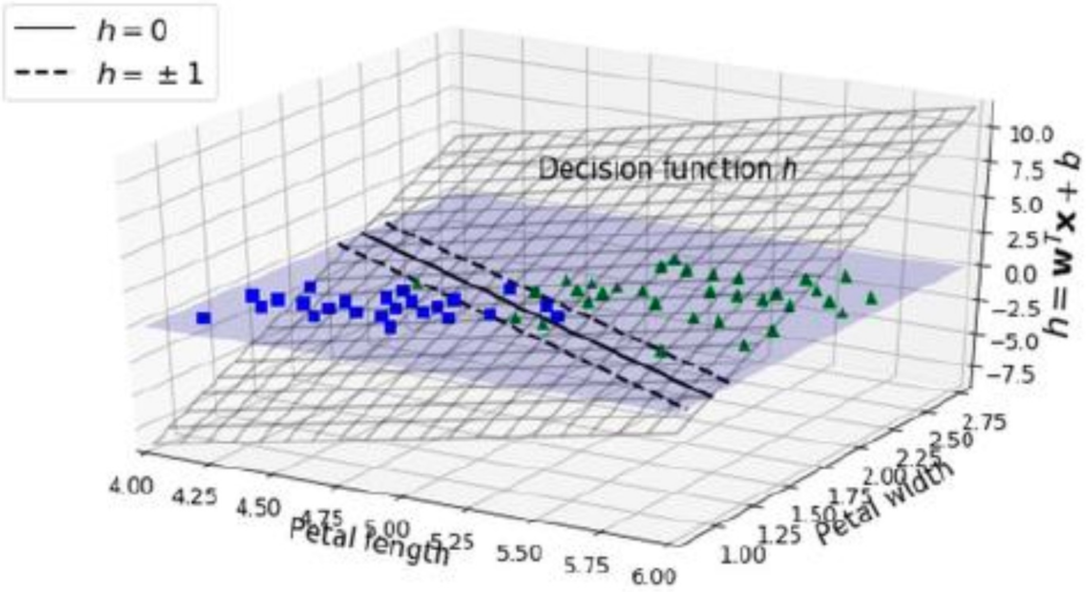
- Prediction of y given x

  $\hat{y} = \text{sign}\{f(\mathbf{x})\} = \text{sign}\{\beta_0 + \boldsymbol{\beta}^t \mathbf{x}\}$

Hyperplane

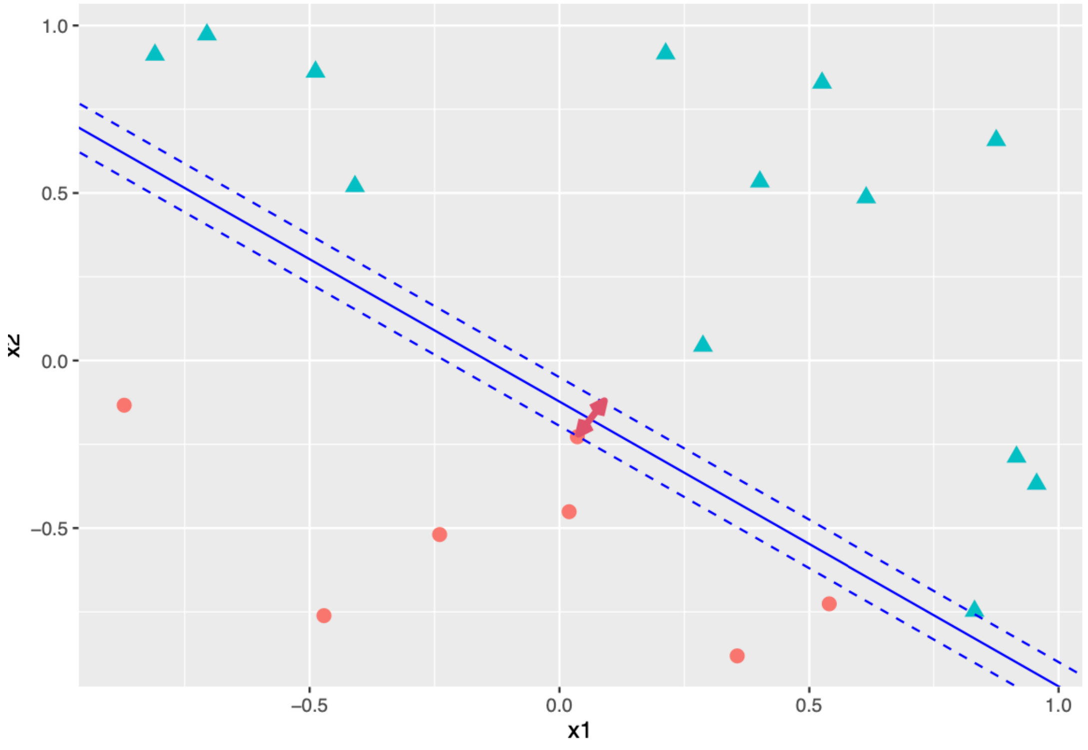


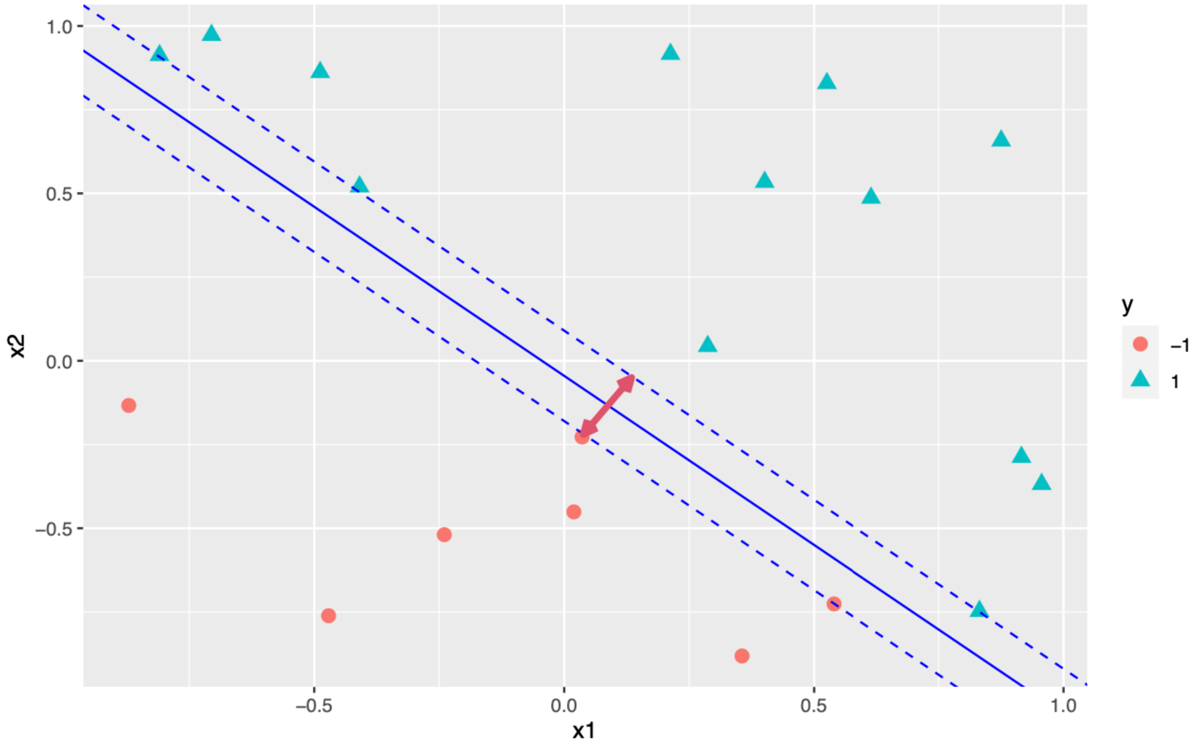
Iris-Versicolor
Iris-Setosa
Petal width
Petal length



$h = 0$
$h = \pm 1$
Decision function h
$h = \mathbf{w}^T\mathbf{x} + b$
Petal length
Petal width

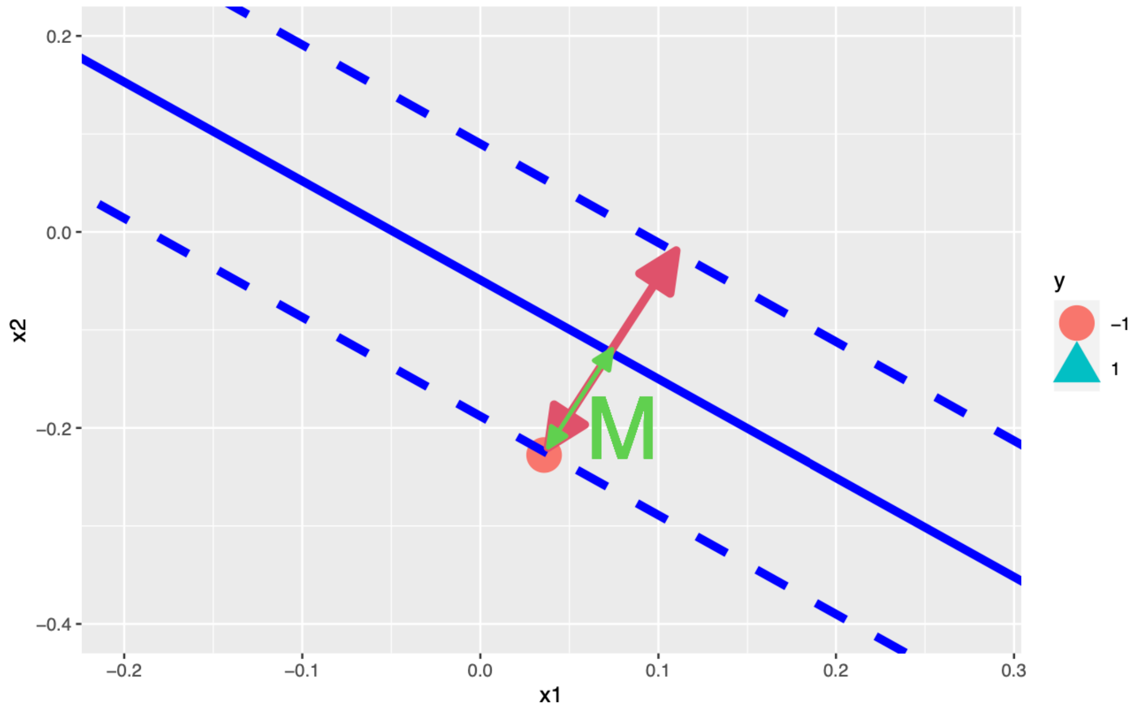
Optimal Separating Hyperplane



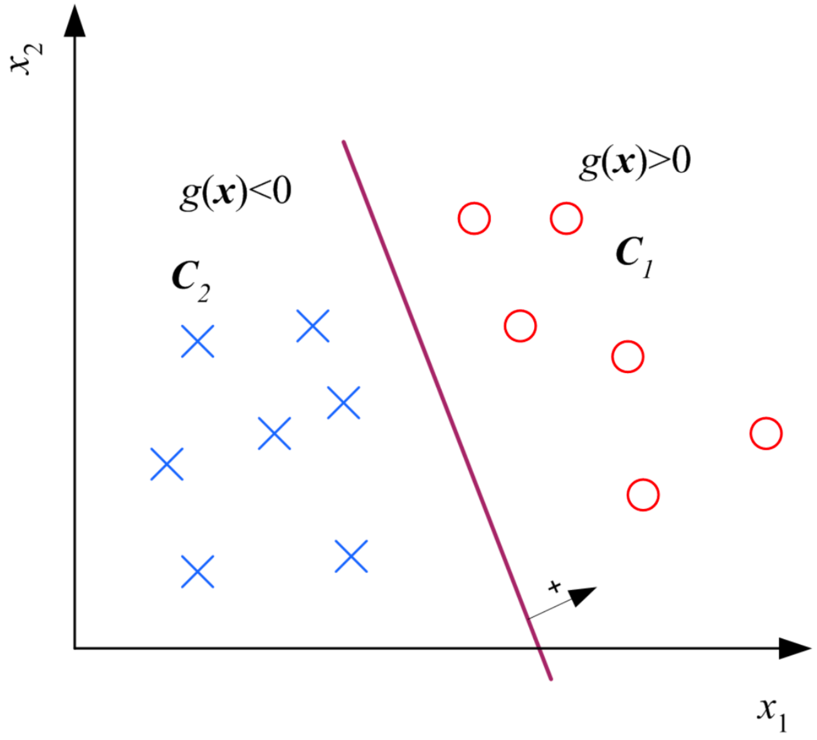


Optimal Separating Hyperplane

- Optimal Separating Hyperplane maximizes Geometric Margin, M.



Two Classes

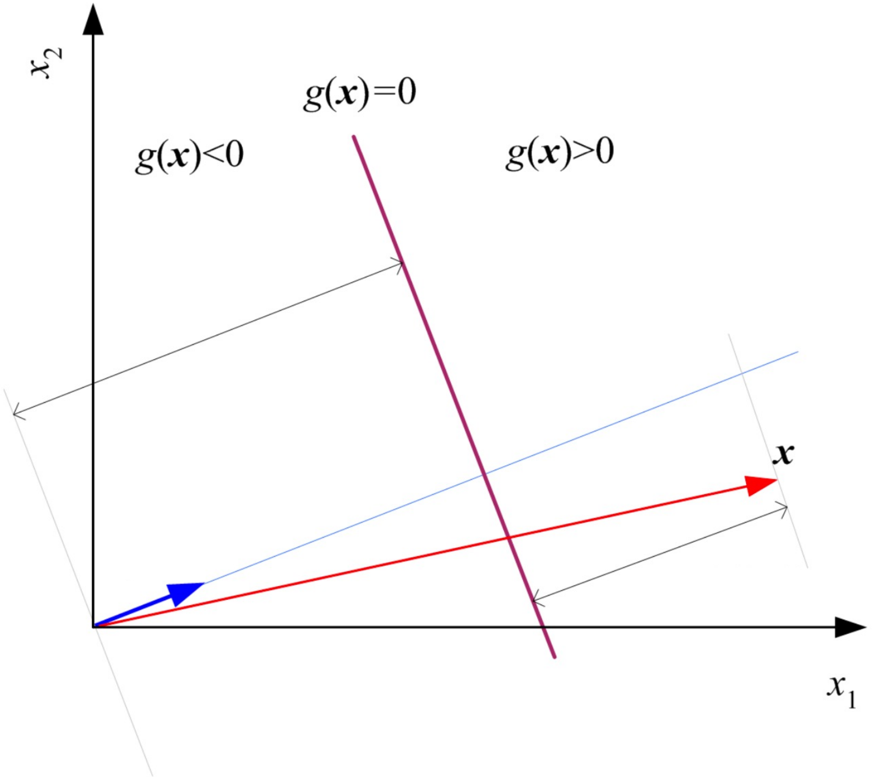


$$g(\mathbf{x}) = g_1(\mathbf{x}) - g_2(\mathbf{x})$$

$$\text{choose } \begin{cases} C_1 & \text{if } g(\mathbf{x}) > 0 \\ C_2 & \text{otherwise} \end{cases}$$

Geometric Margin

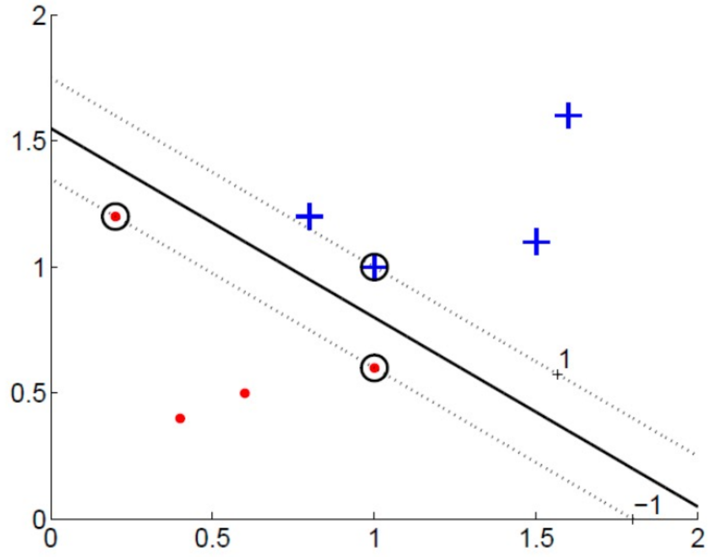
- Geometric margin of x^* $= \dfrac{|\beta^T x + \beta_0|}{\|\beta\|}$



Geometric Margin

Assume $\|\boldsymbol{\beta}\| = 1$, the geometric margin M of $\mathbf{x}_i$ to the hyperplane $\beta_0 + \beta^T \mathbf{x}$ is

$$M = y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i)$$



if $\mathbf{x}_i$ is on right (i.e. $\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i > 0$ and $y_i = 1$) $\quad M = \beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i$

if $\mathbf{x}_i$ is on left (i.e. $\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i < 0$ and $y_i = 1$) $\quad M = -(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i)$

Maximal Margin Classifier

$$\max_{\beta_0,\boldsymbol{\beta}} M \quad \text{subject to} \;\; y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) \geq M, i = 1, \cdots, n;$$

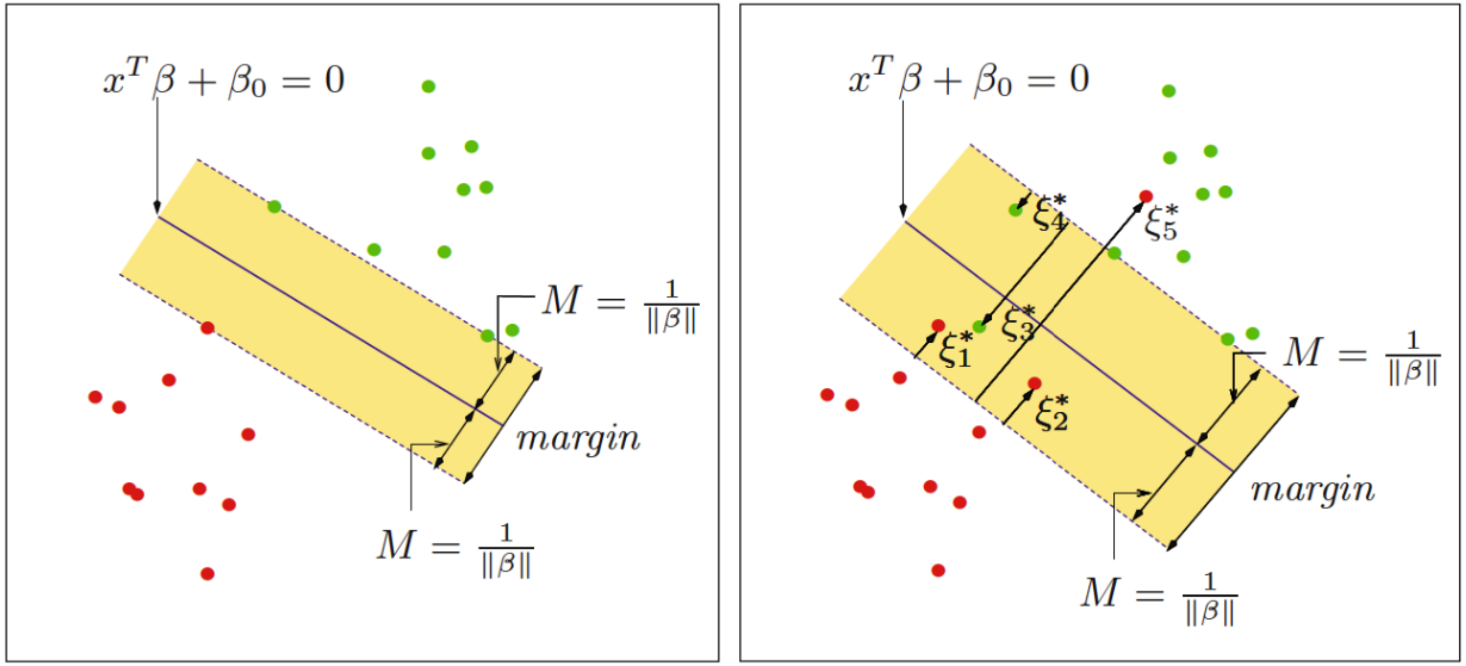
- For a unique solution , fix $M \parallel \beta \parallel = 1$.
- Maximize M $\Leftrightarrow$ minimize $\parallel \beta \parallel$

  $\Leftrightarrow$ minimize $\parallel \beta \parallel^2$

$$\min_{\beta_0,\boldsymbol{\beta}} \frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} \quad \text{subject to} \;\; y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) \geq M, i = 1, \cdots, n;$$

What if Nonseparable?

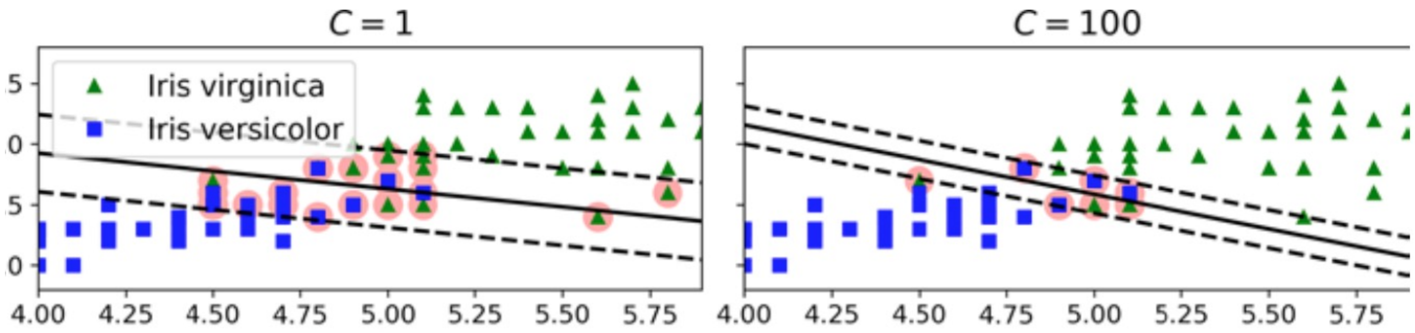
- Let's relax the constraints by introducing slack variables $\xi_i \geq 0$, and add penalty C for the violations.



Support Vector Machine – Soft Margin Classifier

$$\min_{\beta_0, \boldsymbol{\beta}, \xi_i} \boldsymbol{\beta}^T \boldsymbol{\beta} + C \sum_{i=1}^{n} \xi_i \qquad \text{subject to} \quad y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) \geq 1 - \xi_i, \quad i = 1, \cdots, n$$

$$\xi_i \geq 0, \qquad i = 1, \cdots, n.$$



Computation of SVM

Lagrangian Method

- Constraint Optimization for **x**:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{subject to} \quad & h_i(\mathbf{x}) \leq 0, i = 1, \cdots, n \end{aligned} \tag{1}$$

- The Lagrangian associated the problem (1) is $f(\mathbf{x}) + \sum_{i=1}^{n} \alpha_i h_i(\mathbf{x})$

- where $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_m)$ are called the dual variables or Lagrange multipliers associated with the problem (1).

Computation of SVM

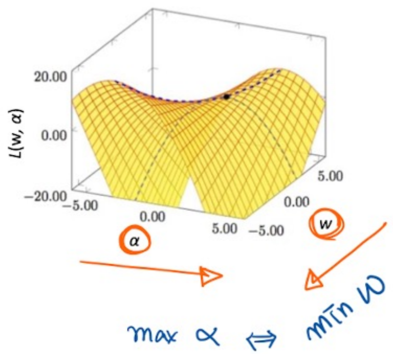
- Lagrangian function of the linear SVM is

$$L_p : \quad \frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} + C\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i\{1 - y_i(\beta_0 - \boldsymbol{\beta}^T\mathbf{x}_i) - \xi_i\} - \gamma_i\sum_{i=1}^{n}\xi_i \qquad (2)$$

where α_i and γ_i are (non-negative) Lagrangian multiplier.

- KKT Stationary conditions

Karush-Kuhn-Tucker theorem:
If (w^*,α^*) is a saddle point of L(w, α) in $\alpha \geq 0$, then x^* is an optimal vector.



$$\frac{\partial}{\partial \boldsymbol{\beta}} L_p : \quad \boldsymbol{\beta} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial}{\partial \beta_0} L_p : \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\frac{\partial}{\partial \xi_i} L_p : \quad \alpha_i = C - \gamma_i$$

Dual Problem of SVM

- Dual function for the linear SVM : $$g(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

- Dual problem of the linear SVM :

$$\max_{\boldsymbol{\alpha}} \; g(\boldsymbol{\alpha})$$

$$\text{subject to } 0 \leq \alpha_i \leq C, \quad i = 1, \cdots, n;$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0.$$

- In Matrix notation :

$$\max_{\boldsymbol{\alpha}} \; \mathbf{1}^T \boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{K}^* \boldsymbol{\alpha}$$

$$\text{subject to } \mathbf{1} \leq \boldsymbol{\alpha} \leq C\mathbf{1},$$

$$\mathbf{y}^T \boldsymbol{\alpha} = 0$$

Where $\{\mathbf{K}^*\}_{ij} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j = y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle$

Computation of SVM

- Primal problem of the linear SVM : $\min_{\beta_0, \boldsymbol{\beta}} \frac{1}{2} \boldsymbol{\beta}^T \boldsymbol{\beta}$ subject to $y_i(\beta_0 + \boldsymbol{\beta}^T \mathbf{x}_i) \geq M, i = 1, \cdots, n;$

- Dual problem of the linear SVM : $\max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha})$ subject to $0 \leq \alpha_i \leq C, \quad i = 1, \cdots, n;$

$$\sum_{i=1}^{n} \alpha_i y_i = 0.$$

- SVM Solution : $f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} = \beta_0 + \sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle$

Computation of SVM

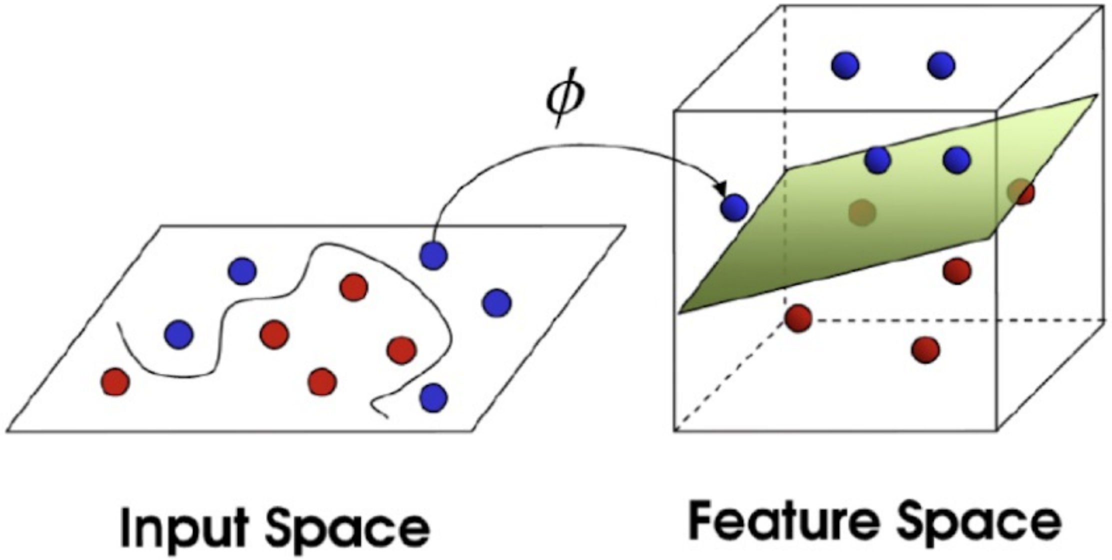
- Decision boundary : $f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} = \beta_0 + \sum_{i=1}^{n} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle$

- SVM solution depends on x_is only through their inner products.

Extension to Nonlinear Classification

- Using Kernel Trick

Kernel Trick



$$\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \cdots, \phi_n(\mathbf{x}))$$

Kernel Trick

- decision function on space of x is

$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} y_i \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle$$

- decision function on feature space is

$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} y_i \alpha_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle$$

- Thus what we need is not the feature ϕ, but its inner product : Kernel Function

$$K(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$$

Kernel Trick

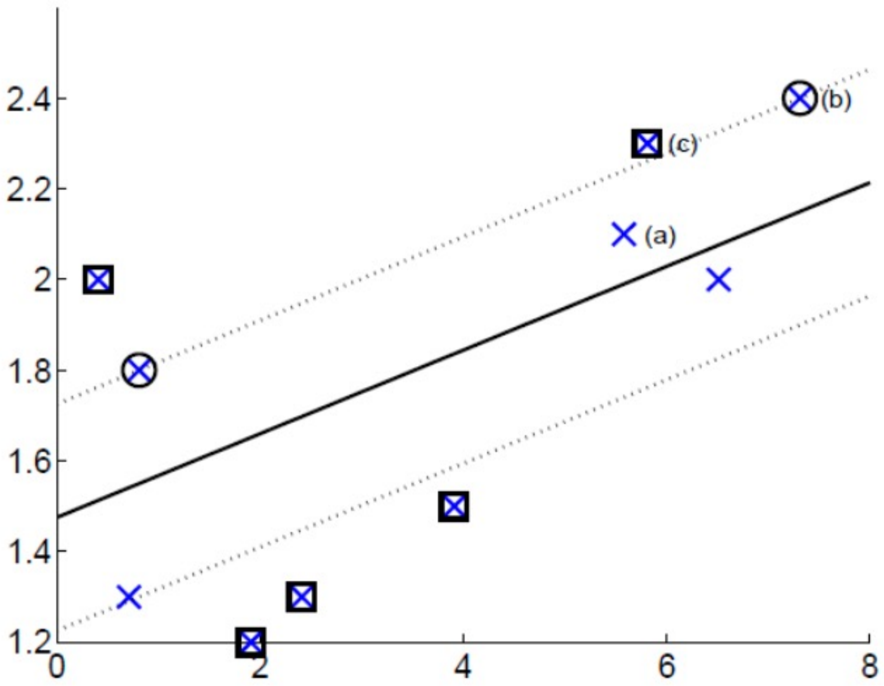
$$f(\mathbf{x}) = \beta_0 + \sum_{i=1}^{n} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x})$$

$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$	*Linear Kernel*
$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j))$	*Gaussian Kernel (Radial Basis function)*
$K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma + \gamma\, \mathbf{x}_i^T \mathbf{x}_j)^p$	*polynomial Kernel*
$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(k_1 \mathbf{x}_i^T \mathbf{x}_j + k_2)$	*Sigmoid Kernel*

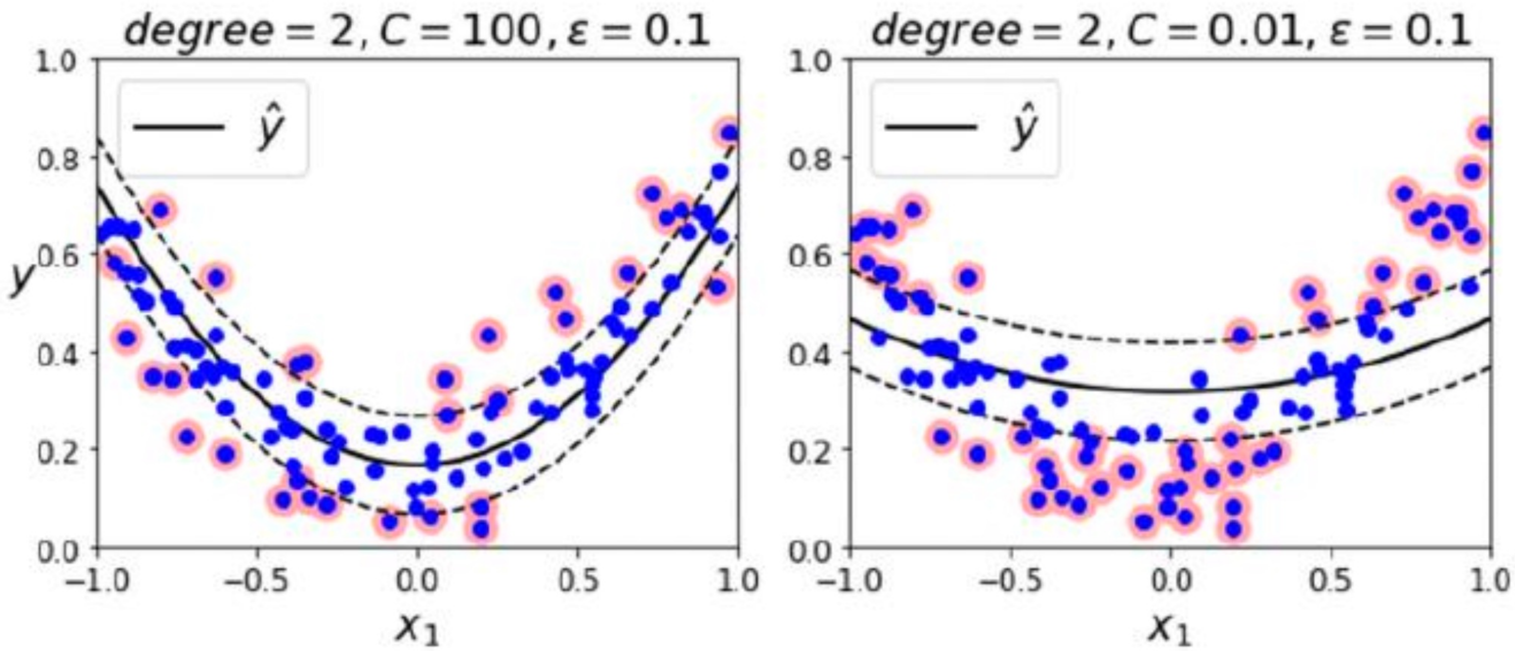
Support Vector Machine

- Regression

SVM – Regression



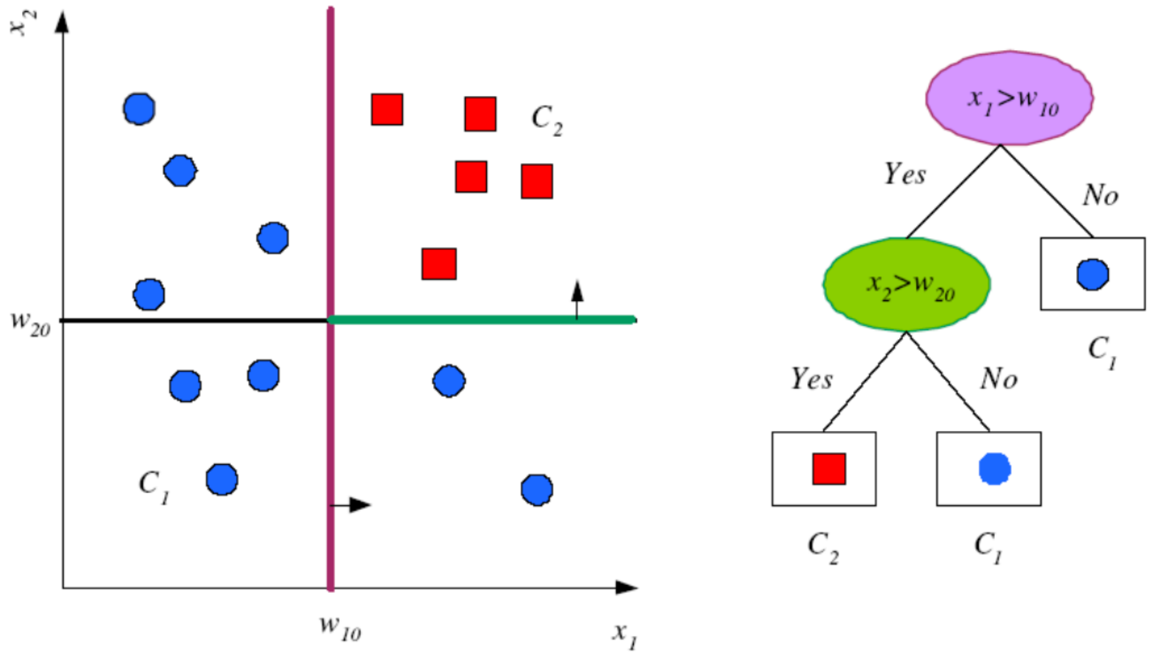
SVM – Regression



Decision Tree

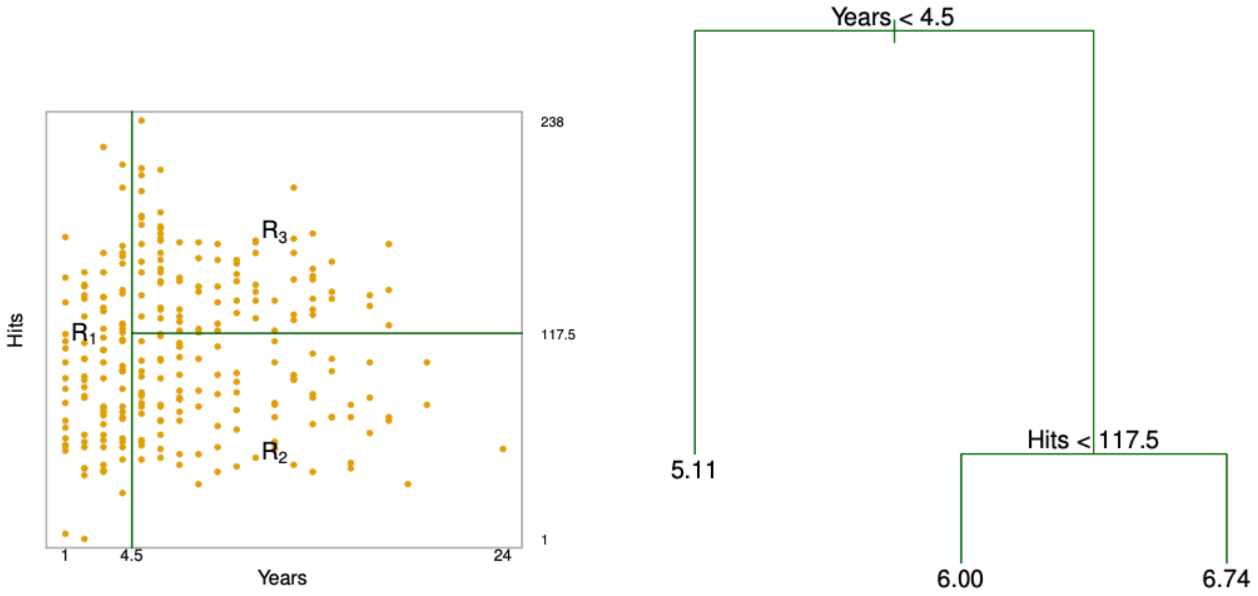
Decision Tree

- Classification



Decision Tree

- Regression



(a) Partitioned Predictor Space (b) Fitted Model

Impurity

- Minimize the impurity of leaf node
- "Best Split" = Minimize the total impurity
- Measure of Impurity → Classification : Entropy $\sum_k p_k^\ell (1 - p_k^\ell)$ or $\sum_k p_k^\ell \log p_k^\ell$

  → Regression : MSE $\sum_i (y_i^\ell - \bar{y}^\ell)^2$

Pruning Trees

Size of tree is a tuning parameter.

- Too large Tree: Overfitting (High Variance/Low Bias)
- Too small Tree: Underfitting (Low Variance/High Bias)

Pruning methods

- Prepruning : Early stopping
- Postpruning : Grow the whole tree then prune subtrees