

- margin, max. margin classifier  
= SVM
- primal problem.  $\rightarrow$  dual  $\rightarrow$  KKT condition
- Support vector.

## Inclass 08: Support Vector Machine

[SCS4049] Machine Learning and Data Science

---

Seongsik Park (s.park@dgu.ac.kr)

School of AI Convergence & Department of Artificial Intelligence, Dongguk University

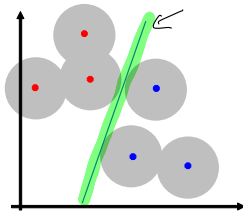
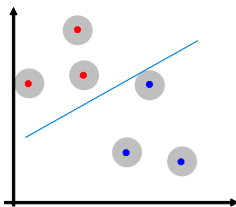
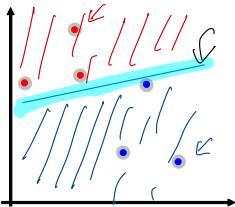
# Maximum Margin Classifier

# What is a good decision boundary?

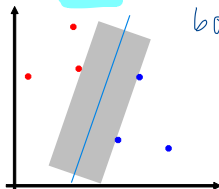
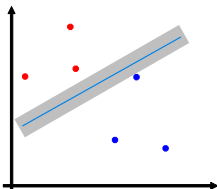
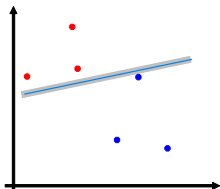
## ■ 데이터 노이즈에 대한 강건성 (Robustness)

노이즈(측정 오차)에 대해서 강건한 것이 좋은 모델이다.

샘플  $\leadsto$  boundary



여유로운 것이 더 강건하다  $\Rightarrow$  넓은 통로가 좋다  $\Rightarrow$  Large Margin Classification



boundary  $\rightarrow$  sample

margin = boundary에서 가장 가까운 sample까지 거리.

# What is a good decision boundary?

강제 training sample 의 일부분.

의사 결정은 경계의 데이터(support vectors)에 의해서 결정됨

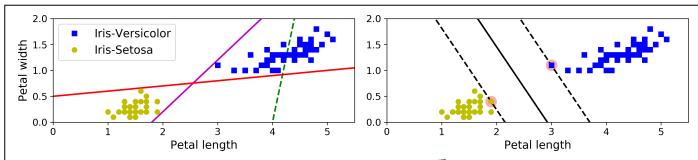


Figure 5-1. Large margin classification

이름이 활용되는 sample.

boundary를 만들어내는 vector (sample).

○ Input feature의 scale에 민감한 support vector machine

거리를 기반 (margin)

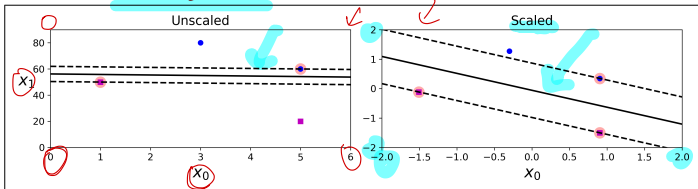


Figure 5-2. Sensitivity to feature scales

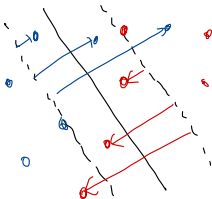
# Hard margin vs. soft margin

## Hard margin classification (hard-SVM)

- 모든 데이터들이 margin constraint. 밖에 위치하도록 boundary를 설정
- 데이터가 linearly separable할 때만 적용 가능
- ✗ outlier에 매우 민감

## Soft margin classification (soft-SVM)

- margin을 가능한 넓게 하면서도 margin 안에 들어오는 것도 허용
- hyperparameter C: 클수록 좁아짐 (엄격), 작을 수록 넓어짐 (허용)



# Hard margin vs. soft margin

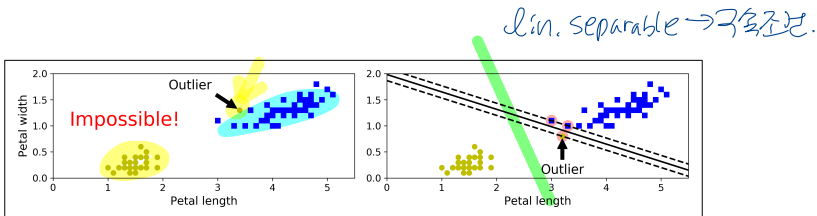


Figure 5-3. Hard margin sensitivity to outliers

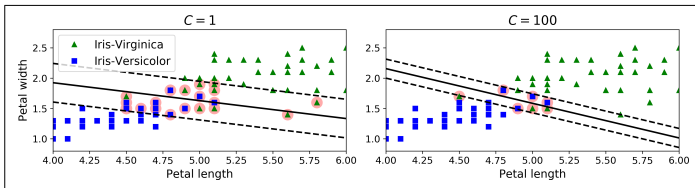


Figure 5-4. Large margin (left) versus fewer margin violations (right)

## A brief history of SVM

- SVM은 1992년 Boser, Guyon and Vapnik에 의해서 소개됨 *convex.*
- Statistical Learning Theory에 이론적 바탕을 둔 알고리즘 
- 손글씨 숫자 인식에서 뛰어난 성능을 보이면서 널리 쓰이게 됨
- SVM으로 1.1% Test error rate  $\approx$  신중히 설계된 신경망과 맞먹음
- 실용적으로 우수한 성능
- Bioinformatics, text, image recognition 등 많은 성공 사례
- 선형/비선형 분류 뿐 아니라 회귀, outlier detection 도 수행
- 복잡한 소규모/중규모 데이터셋의 분류에 특히 잘 맞음
- Kernel 방법을 사용하는 대표적 알고리즘
- Liblinear & libsvm: Scikit-Learn 에서 liblinear 및 libsvm 을 사용

## Hard SVM

---



# Maximum margin classifier

We begin our discussion of support vector machines to the two-class classification problem using linear models of the form

$$\text{linear model} \quad y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b \quad \text{with } \mathbf{w}, b \quad (1)$$

~~where  $\mathbf{x}$  denotes a fixed feature-space transformation, and we have made the bias parameter  $b$  explicit.~~

The training data set comprises  $N$  input vectors  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ , with corresponding target values  $t_1, t_2, \dots, t_N$  where  $t_n \in \{-1, 1\}$ , and new data points  $\mathbf{x}$  are classified according to the sign of  $y(\mathbf{x})$

$$t_n = +1$$

$$t_n = -1$$

$$y(\mathbf{x}) > 0$$

$$y(\mathbf{x}) < 0$$

# Maximum margin classifier

We shall assume that the training data set is linearly separable in feature space, so that by definition there exists at least one choice of the parameters  $\mathbf{w}$  and  $b$  such that a function satisfies  $y(\mathbf{x}_n) > 0$  for points having  $t_n = +1$  and  $y(\mathbf{x}_n) < 0$  for points having  $t_n = -1$ , so that  $t_n y(\mathbf{x}_n) > 0$  for all training data points.

⊗

$$t_n = +1, \quad y(\mathbf{x}_n) > 0,$$

$$t_n = -1, \quad y(\mathbf{x}_n) < 0,$$

$$\underline{t_n y(\mathbf{x}_n) > 0}$$

$$\underline{-t_n y(\mathbf{x}_n) > 0}$$

## Maximum margin classifier: optimality criterion

Thus the distance of a point  $\mathbf{x}_n$  to the decision surface is given by

$$\frac{y(\mathbf{x}_n) = \mathbf{w}^T \mathbf{x}_n + b}{\|\mathbf{w}\|_2} = \frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n (\mathbf{w}^T \mathbf{x}_n + b)}{\|\mathbf{w}\|}. \quad (2)$$

The margin is given by the perpendicular distance to the closest point  $\mathbf{x}_n$  from the data set, and we wish to optimize the parameters  $\mathbf{w}$  and  $b$  in order to maximize this distance. Thus the maximum margin solution is found by solving

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min [t_n (\mathbf{w}^T \mathbf{x}_n + b)] \right\} \quad (3)$$

where we have taken the factor  $1/\|\mathbf{w}\|$  outside the optimization over  $n$  because  $\mathbf{w}$  does not depend on  $n$ .

$$\begin{array}{c} t_1 \quad \underline{x_1} \\ t_2 \quad \underline{x_2} \\ \vdots \end{array}$$

$$t_{100} \quad \underline{x_{100}}$$

objective.



$$\max \min_n \frac{t_n (w^T x_n + b)}{\|w\|}$$

$$= \max \underline{\text{margin}}$$

$$\max_{w, b} \min_n \frac{t_n(w^T x_n + b)}{\|w\|}$$

$$= \max_{w, b} \frac{1}{\|w\|} \min_n \underline{t_n(w^T x_n + b)} \quad \checkmark$$

$$\text{s.t.} \quad \underline{t_n(w^T x_n + b)} > 0$$

$$n = 1, 2, \dots, N$$

$$\max \frac{1}{\|w\|} \min t_n(w^T x_n + b) = 1$$

$$\text{s.t.} \quad \underbrace{t_n(w^T x_n + b)}_{\sim} \underbrace{\min t_n(w^T x_n + b)}_{=1} \sim > 0$$

$$\Rightarrow \max \frac{1}{\|w\|} \Rightarrow \min \frac{1}{2} \|w\|^2$$

$$\text{s.t.} \quad \underline{t_n(w^T x_n + b) \geq 1}$$

$$\textcircled{1} \quad y(x) = x_1 + x_2 + 1 \quad \Leftarrow$$


---


$$w = [1 \ 1]^T \quad \underline{b=1}$$

$$\frac{|x_1 + x_2 + 1|}{\sqrt{2}} \quad \Leftarrow$$

||

$$\textcircled{2} \quad y(x) = 2x_1 + 2x_2 + 2 \quad \Leftarrow$$


---


$$w = [2 \ 2]^T \quad \underline{b=2}$$

$$\frac{|2x_1 + 2x_2 + 2|}{2\sqrt{2}} \quad \Leftarrow$$

$$\frac{|w^T x_n + b|}{\|w\|} = \frac{t_n(w^T x_n + b)}{\|w\|} \quad \Leftarrow$$

# Dual problem for convex optimization

Primal optimization problem for hard SVM

$$\text{maximize } \frac{1}{\|\mathbf{w}\|} \min[t_n(\mathbf{w}^T \mathbf{x}_n + b)] \quad (4)$$

$$\text{subject to } t_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 0 \quad (5)$$

(6)

Equivalently,

$$\text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 \quad (7)$$

$$\text{subject to } t_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1 \quad (8)$$

Direct solution of this optimization problem would be very complex, so we shall convert it into an equivalent problem that is much easier to solve.



# Lagrangian function with constraint

Setting the derivatives of  $L(\mathbf{w}, b, \mathbf{a})$  with respect to  $\mathbf{w}$  and  $b$  equal to zero, we obtain the following two conditions

$$\mathbf{w} = \sum_{n=1}^N a_n t_n \mathbf{x}_n$$

(9)

$$0 = \sum_{n=1}^N a_n t_n$$

(10)

$$a_1 - t_1 (w^T x_1 + b) \geq 1$$

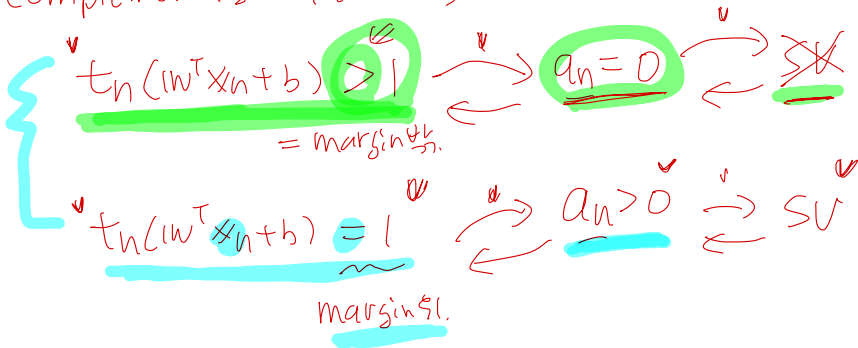
$$a_2 - t_2 (w^T x_2 + b) \geq 1$$

$$\vdots$$

$$W = \sum a_n t_n x_n$$

$$y(x) = \sum a_n t_n x_n^T x + b$$

Complementary slackness



# Lagrangian function with constraint

Eliminating  $\mathbf{w}$  and  $b$  from  $L(\mathbf{w}, b, \mathbf{a})$  using these conditions then gives the *dual representation* of the maximum margin problem in which we maximize

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \quad (11)$$

with respect to  $\mathbf{a}$  subject to the constraints

$$a_n \geq 0, \quad n = 1, \dots, N \quad (12)$$

$$\sum_{n=1}^N a_n t_n = 0. \quad (13)$$

Here the kernel function is defined by  $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$ .

## Prediction for a new sample: support vector machine

In order to classify new data points using the trained model, we evaluate the sign of  $y(\mathbf{x})$ . This can be expressed in terms of the parameter  $\{a_n\}$  and the kernel function by substituting for  $\mathbf{w}$  to give

$$y(\mathbf{x}) = \sum_{n=1}^N a_n t_n \mathbf{x}_n^T \mathbf{x} + b \quad (14)$$

$$= \sum_{n=1}^N a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \quad (15)$$

## KKT condition: complementary slackness

We show that a constrained optimization of this form satisfies the *Karush-Kuhn-Tucker* (KKT) conditions, which in this case require that the following three properties hold

dual constraint.  $a_n \geq 0$  (16) ↙

primal constraint.  $t_n y(\mathbf{x}_n) - 1 \geq 0$  ↘ (17)

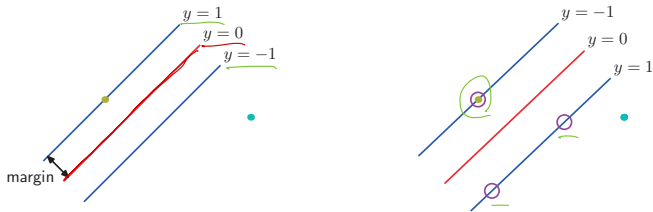
comple. slackness.  $a_n \{t_n y(\mathbf{x}_n) - 1\} = 0$  (18)

Thus for every data point, either  $a_n = 0$  or  $t_n y(\mathbf{x}_n) = 1$ . Any data point for which  $a_n = 0$  will not appear in the sum and hence plays no role in making predictions for new data points. The remaining data points are called support vectors, and because they satisfy  $t_n y(\mathbf{x}_n) = 1$ , they correspond to points that lie on the maximum margin hyperplanes in feature space.

$$w = \sum a_n t_n x_n$$

$$y(x) = \sum a_n t_n x_n^T x + b$$

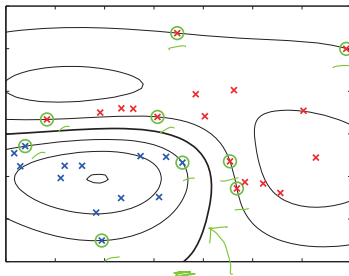
# KKT condition: complementary slackness



**Figure 7.1** The margin is defined as the perpendicular distance between the decision boundary and the closest of the data points, as shown on the left figure. Maximizing the margin leads to a particular choice of decision boundary, as shown on the right. The location of this boundary is determined by a subset of the data points, known as support vectors, which are indicated by the circles.

# KKT condition: complementary slackness

**Figure 7.2** Example of synthetic data from two classes in two dimensions showing contours of constant  $y(\mathbf{x})$  obtained from a support vector machine having a Gaussian kernel function. Also shown are the decision boundary, the margin boundaries, and the support vectors.





### 장점

- 강력하고 우수한 이론을 바탕으로 함 *global convex, opt.*
- 많은 블랙박스 알고리즘과는 대조적으로 비교적 직관적인 해석과 이해가 가능
- 학습이 상대적으로 쉬움 
- 신경망처럼 지역 최적값에 빠지는 일이 없음
- 학습 시간이 차원에 의존하지 않으며 kernel trick 덕분에 고정된 입력에만 의존
- 과적합이 잘 조절되는 경향
- 많은 분야에서 신경망 및 기타 알고리즘과 필적하는 성능
- 데이터가 작은 조건이나 고차원 공간에서도 잘 일반화

## 단점

- 노이즈에 민감
- 비교적 적은 수의 잘못된 label로 성능이 심각하게 악화
- 커널 함수를 선택하는 방법에 대한 정리된 원칙이 없음
- hyperparameter  $C$ 의 적정값을 정하기 위한 원칙이 없음
- 컴퓨터의 메모리와 계산 시간 측면에서 비용이 높은 편이며  
⇒ multiclass에서 더욱더 심화됨



# Appendix

---

## Reference and further reading

- “Chap 7 | Sparse Kernel Machines” of C. Bishop, Pattern Recognition and Machine Learning
- “Chap 5 | Support Vector Machines” of A. Geron, Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow
- “Chap 4 | Convex Optimization Problems”, “Chap 5 | Duality” of S. Boyd, Convex Optimization
- “Lecture 6 | Support Vector Machines” of Kwang Il Kim, Machine Learning (2019)