Lecture 01: Convex Optimization and SVM

[AIX7026] Advanced Machine Learning

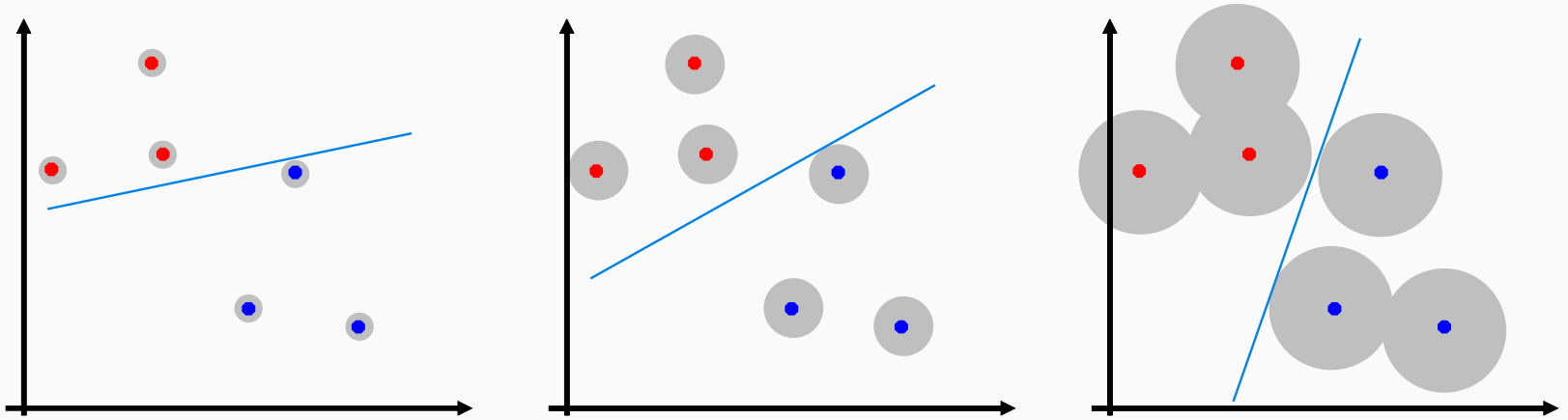
Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

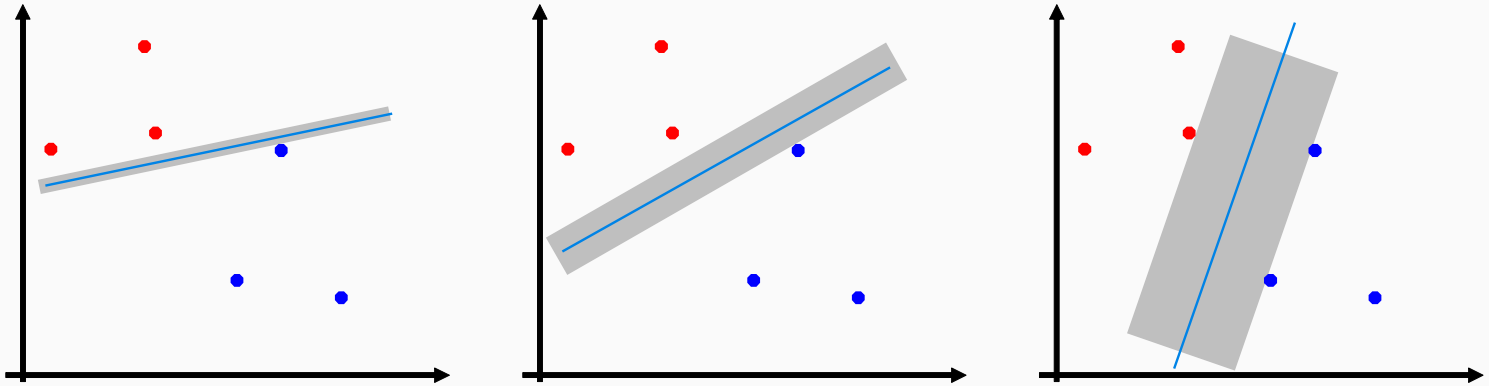
SVM: Maximum Margin Classifier

What is a good decision boundary?

- 데이터 노이즈에 대한 강건성 (Robustness)
  - 노이즈(측정 오차)에 대해서 강건한 것이 좋은 모델이다.



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



What is a good decision boundary?

의사 결정은 경계의 데이터(support vectors)에 의해서 결정됨

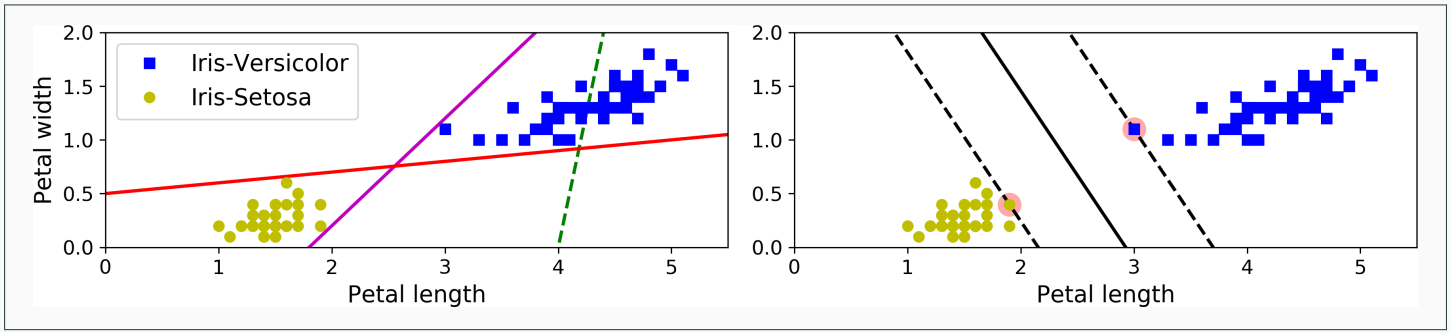


Figure 5-1. Large margin classification

Input feature의 scale에 민감한 support vector machine

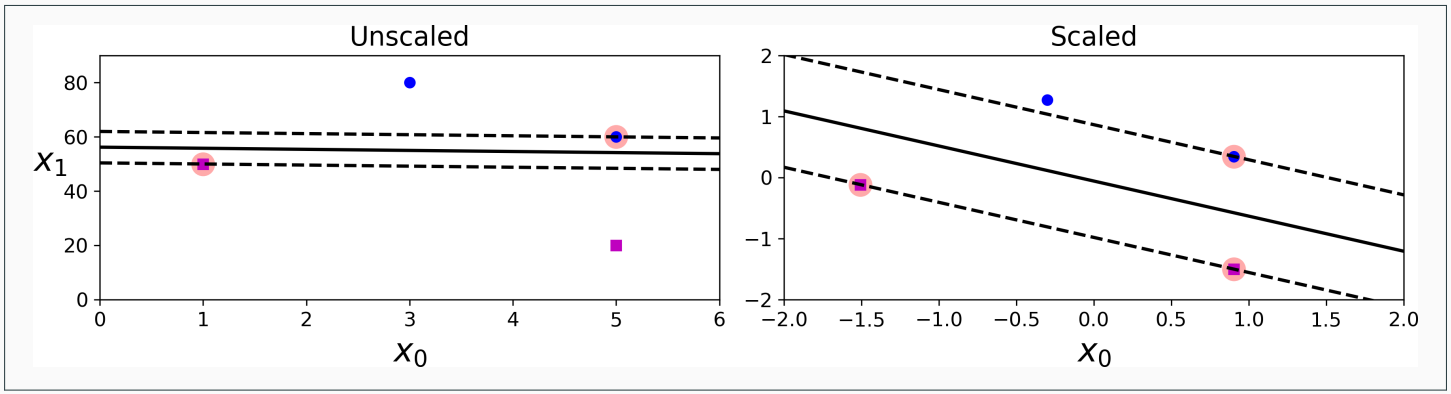


Figure 5-2. Sensitivity to feature scales

Hard margin vs. soft margin

Hard margin classification (hard-SVM)

- 모든 데이터들이 margin 밖에 위치하도록 boundary를 설정
- 데이터가 선형적으로 분리 가능(linearly separable)할 때만 적용 가능
- outlier에 매우 민감

Soft margin classification (soft-SVM)

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

Hard margin vs. soft margin

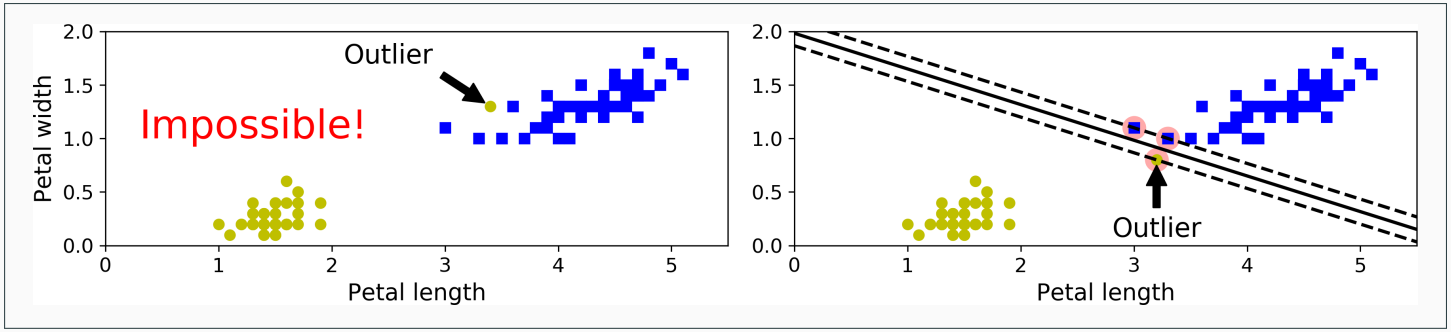


Figure 5-3. Hard margin sensitivity to outliers

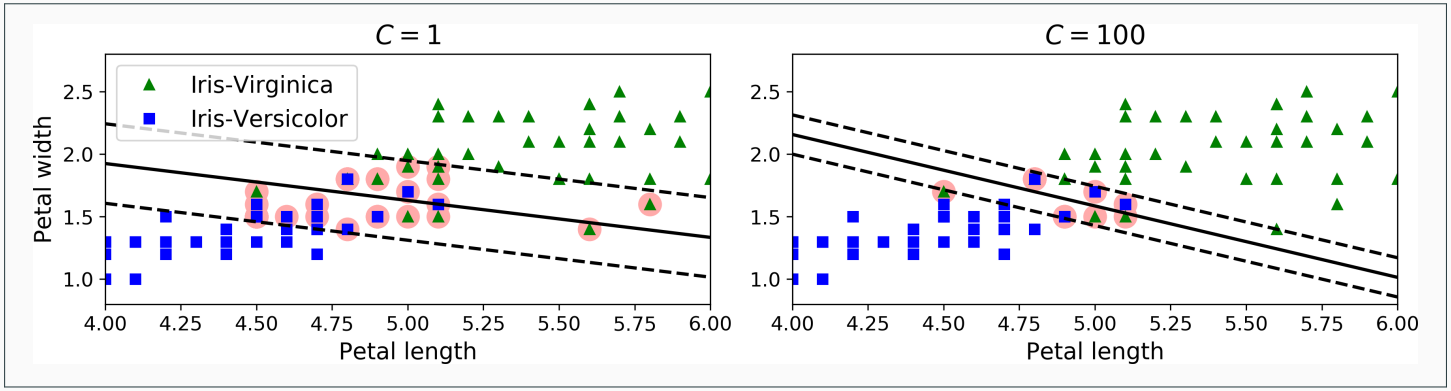


Figure 5-4. Large margin (left) versus fewer margin violations (right)

A brief history of SVM

- SVM은 1992년 Boser, Guyon and Vapnik에 의해서 소개됨
- Statistical Learning Theory에 이론적 바탕을 둔 알고리즘 (Vapnik & Chervonenkis)
- 손글씨 숫자 인식에서 뛰어난 성능을 보이면서 널리 쓰이게 됨
- SVM으로 1.1% Test error rate ≈ 신중히 설계된 신경망(e.g., LeNet 4)과 맞먹음
- 실용적으로 우수한 성능
- bioinformatics, text, image recognition 등을 포함한 많은 성공 사례
- 강력하고 다재 다능한 머신 러닝 모델
- 선형/비선형 분류 뿐 아니라 회귀, outlier detection 도 수행
- 복잡한 소규모/중규모 데이터셋의 분류에 특히 잘 맞음
- 머신 러닝에서 중요한 기법 중 하나인 Kernel 방법을 사용하는 대표적 알고리즘
- 머신 러닝에서 가장 널리 쓰이는 모델이며 머신 러닝을 하며 반드시 알아야 할 기법 중 하나
- Liblinear & libsvm: Scikit-Learn 에서 liblinear 및 libsvm 을 사용하여 구현

Convex Optimization and Duality

MoG: Lagrange multiplier

The method can be summarized as follows: in order to find the maximum or minimum of a function $f(x)$ subjected to the equality constraint $g(x) = 0$, form the Lagrangian function

$$\mathcal{L}(x, \lambda) = f(x) - \lambda g(x) \tag{1}$$

and find the stationary points of $\mathcal{L}$ considered as a function of x and the Lagrange multiplier λ. The solution corresponding to the original constrained optimization is always a saddle point of the Lagrangian function, which can be identified among the stationary points from the definiteness of the bordered Hessian matrix.

MoG: Lagrange multiplier

Minimize $f(x, y) = x + y$ subject to the constraint $x^2 + y^2 = 1$, i.e.,

$$g(x, y) = x^2 + y^2 - 1 = 0 \tag{2}$$

Hence,

$$\mathcal{L}(x, y, \lambda) = f(x, y) + \lambda g(x, y) = x + y + \lambda(x^2 + y^2 - 1) \tag{3}$$

Gradient

$$\nabla_{x,y,\lambda}\mathcal{L}(x, y, \lambda) = (1 + 2\lambda x, 1 + 2\lambda y, x^2 + y^2 - 1) \tag{4}$$

and therefore,

$$\nabla_{x,y,\lambda}\mathcal{L}(x, y, \lambda) \quad \Longleftrightarrow \quad \begin{cases} 1 + 2\lambda x = 0 \\ 1 + 2\lambda y = 0 \\ x^2 + y^2 - 1 = 0 \end{cases} \tag{5}$$

MoG: Lagrange multiplier

$$\nabla_{x,y,\lambda}\mathcal{L}(x,y,\lambda) \iff \begin{cases} 1+2\lambda x = 0 \\ 1+2\lambda y = 0 \\ x^2+y^2-1=0 \end{cases} \tag{6}$$

This yields

$$x = y = -\frac{1}{2\lambda}, \quad \lambda \neq 0 \tag{7}$$

$$\frac{1}{4\lambda^2} + \frac{1}{4\lambda^2} - 1 = 0 \tag{8}$$

So,

$$\lambda = \pm\frac{1}{\sqrt{2}} \tag{9}$$

which implies that the stationary points of $\mathcal{L}$ are

$$\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right), \quad \left(-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right) \tag{10}$$

Optimization problem in standard form

$$\begin{aligned} \text{minimize} \quad & f_0(x) && && (11)\\ \text{subject to} \quad & f_i(x) \leq 0, && i = 1, 2, ..., m && (12)\\ & h_i(x) = 0, && i = 1, 2, ..., p && (13) \end{aligned}$$

- $x \in \mathcal{R}^n$ is the optimization variable
- $f_0 : \mathcal{R}^n \to \mathcal{R}$ is the objective or cost function
- $f_i : \mathcal{R}^n \to \mathcal{R}, i = 1, 2, ..., m$ are the inequality constraint functions
- $h_i : \mathcal{R}^n \to \mathcal{R}$ are the equality constraint functions

Convex optimization problem

Standard form convex optimization problem

$$\begin{aligned} \text{minimize} \quad & f_0(x) && && (14)\\ \text{subject to} \quad & f_i(x) \leq 0, && i = 1, 2, ..., m && (15)\\ & a_i^T x = b_i, && i = 1, 2, ..., p && (16) \end{aligned}$$

- $f_0, f_1, ..., f_m$ are convex
- equality constraints are affine

Often written as

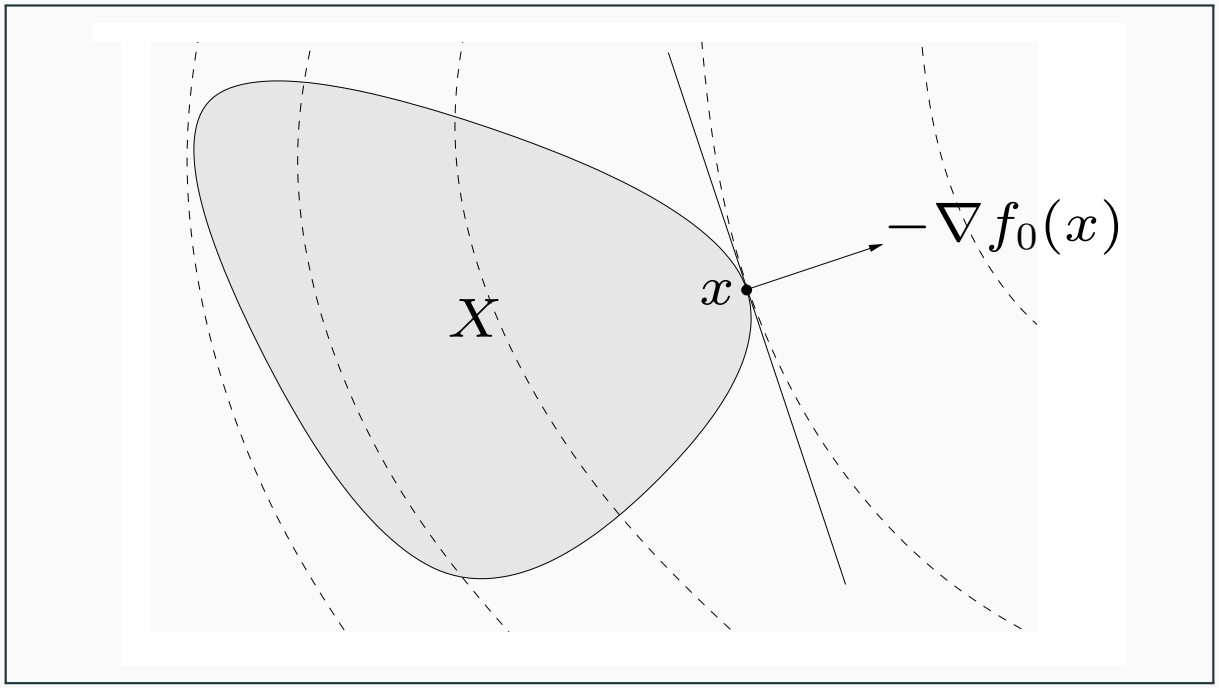
$$\begin{aligned} \text{minimize} \quad & f_0(x) && && (17)\\ \text{subject to} \quad & f_i(x) \leq 0, && i = 1, 2, ..., m && (18)\\ & Ax = b && && (19) \end{aligned}$$

Important property: feasible set of a convex optimization problem is convex

Optimality criterion for differentiable f_0

x is optimal if and only if it is feasible and

$$\nabla f_0(x)^T(y-x) \geq 0 \quad \text{for all feasible } y \tag{20}$$



if nonzero, $\nabla f_0(x)$ defines a supporting hyperplane to feasible set X at x

Lagrangian

standard form problem

$$\begin{aligned} \text{minimize} \quad & f_0(x) && (21)\\ \text{subject to} \quad & f_i(x) \leq 0, \qquad i = 1, 2, ..., m && (22)\\ & h_i(x) = 0, \qquad i = 1, 2, ..., p && (23) \end{aligned}$$

variable $x \in \mathcal{R}^n$, domain $\mathcal{D}$, optimal value p^*

Lagrangian: $L : \mathcal{R}^n \times \mathcal{R}^m \times \mathcal{R}^p \to \mathcal{R}$ with $\mathbf{dom}\, L = \mathcal{D} \times \mathcal{R}^m \times \mathcal{R}^p$

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \qquad (24)$$

- weighted sum of objective and constraint functions
- λ_i is Lagrange multiplier associated with $f_i(x) \leq 0$
- ν_i is Lagrange multiplier associated with $h_i(x) = 0$

Lagrange dual function

Lagrange dual function: $g : \mathcal{R}^m \times \mathcal{R}^p \to \mathcal{R}$

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \tag{25}$$

$$= \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right) \tag{26}$$

g is concave, can be $-\infty$ for some λ, ν

lower bound property: if $\lambda \geq 0$, then $g(\lambda, \nu) \leq p^*$

proof: if $\tilde{x}$ is feasible and $\lambda \geq 0$, then

$$f_0(\tilde{x}) \geq L(\tilde{x}, \lambda, \nu) \geq \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = g(\lambda, \nu) \tag{27}$$

minimizing over all feasible $\tilde{x}$ gives $p^* \geq g(\lambda, \nu)$

The dual problem

Lagrange dual problem

$$\begin{aligned} \text{maximize} \quad & g(\lambda, \nu) && (28) \\ \text{subject to} \quad & \lambda \geq 0 && (29) \end{aligned}$$

- finds best lower bound on p^*, obtained from Lagrange dual function
- a convex optimization problem; optimal value denoted d^*
- λ, ν are dual feasible if $\lambda \geq 0$, $(\lambda, \nu) \in \operatorname{dom} g$
- often simplified by making implicit constraint $(\lambda, \nu) \in \operatorname{dom} g$ explicit

Weak and strong duality

weak duality: $d^* \leq p^*$

- always holds (for convex and nonconvex problems)
- can be used to find nontrivial lower bounds for difficult problems

strong duality: $d^* = p^*$

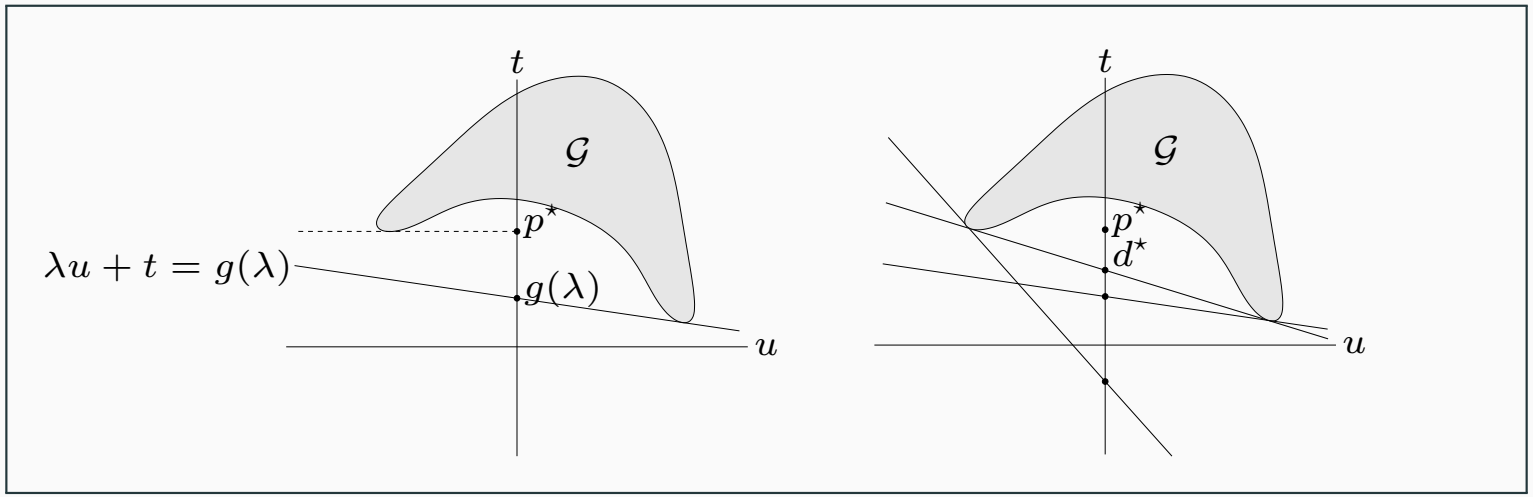
- does not hold in general
- holds for convex problems
- conditions that guarantee strong duality in convex problems are called constraint qualifications

Geometric interpretation

for simplicity, consider problem with one constraint $f_1(x) \leq 0$

interpretation of dual function

$$g(\lambda) = \inf_{(u,t)\in\mathcal{G}} (t + \lambda u) \qquad \text{where } \mathcal{G} = \{(f_1(x), f_0(x)) \mid x \in \mathcal{D}\} \qquad (30)$$



- $\lambda u + t = g(\lambda)$ is supporting hyperplane to $\mathcal{G}$
- hyperplane intersects t-axis at $t = g(\lambda)$

Geometric interpretation

- Primal problem:

$$\begin{aligned} &\text{minimize} \quad f(\mathbf{x}) \\ &\text{subject to} \quad g(\mathbf{x}) \leq 0, \\ &\mathbf{x} \in \mathcal{X} \end{aligned}$$

where $f: \mathbb{R}^n \to \mathbb{R}, \quad g: \mathbb{R}^n \to \mathbb{R}^m$

  - $n = 2$ 에 대해서, 집합 $\mathcal{G}$ 를 다음과 같이 정의하자.

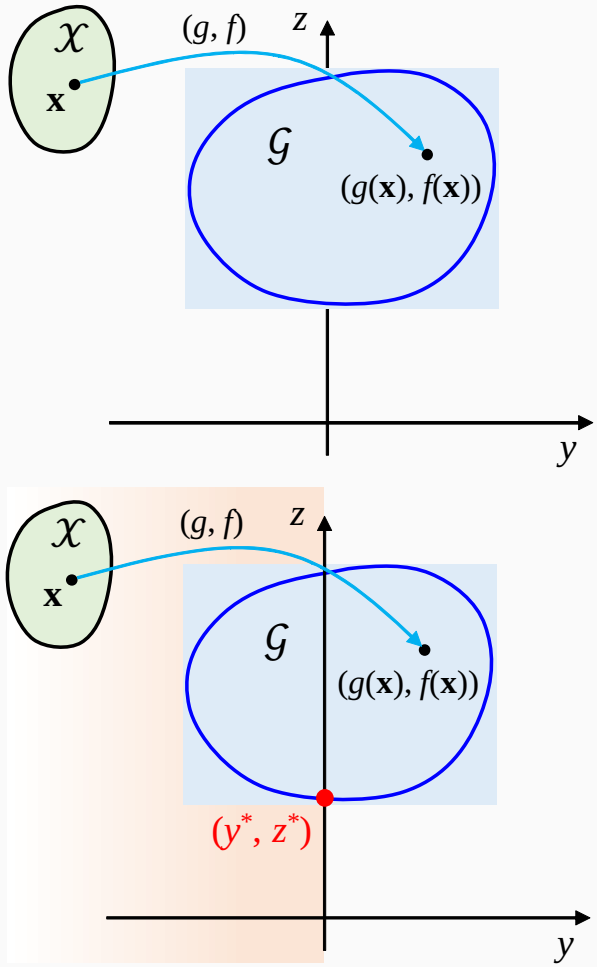
$$\mathcal{G} = \{(y, z) \mid y = g(\mathbf{x}), z = f(\mathbf{x}) \text{ for some } \mathbf{x} \in \mathcal{X}\}$$

$\Longrightarrow$ 그러면, $\mathcal{G}$ 는 사상 (g, f) 하에서 $\mathcal{X}$ 의 치역(image)이다.

$$(g, f): \mathcal{X} \to \mathcal{G}$$

  - 그러면 Primal solution은 최소 세로 좌표값 z를 갖는 $y \leq 0$ 인 $\mathcal{G}$ 내의 점이다.

$\Longrightarrow$ clearly (y^*, z^*)



Geometric interpretation

- Lagrange Dual Problem

$$\text{maximize}_u \ \ \theta(u)$$
$$\text{subject to} \ \ u \geq 0$$

where (Lagrangian subproblem):

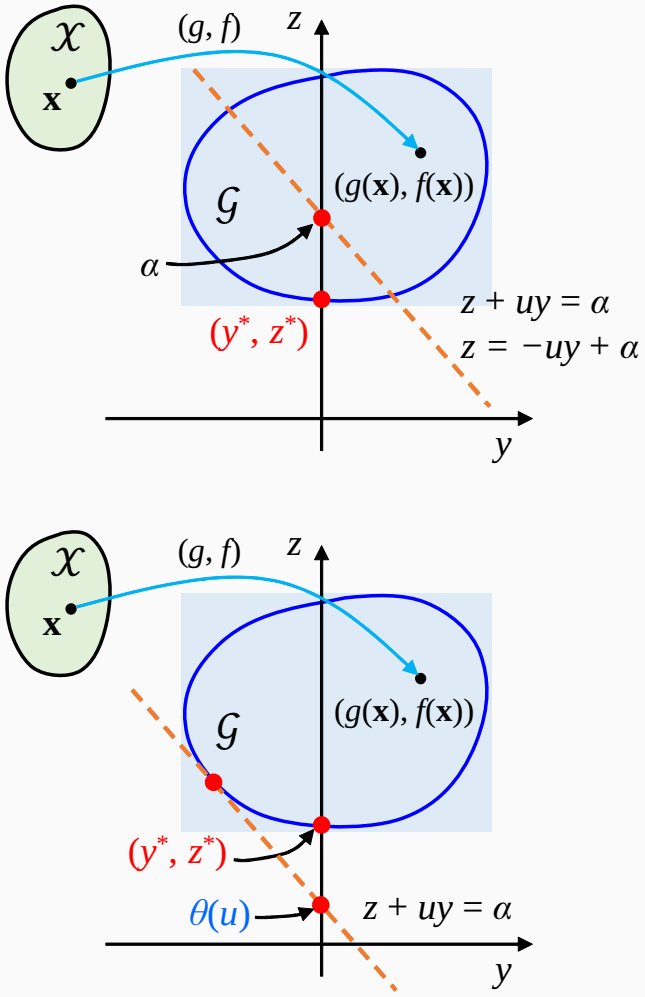
$$\theta(u) \ = \ \inf\{f(\mathbf{x}) + ug(\mathbf{x}) : \mathbf{x} \in \mathcal{X}\}.$$

1. $u \geq 0$ 일 때, Lagrangian dual subproblem 은 다음과 동등하다.

   minimize $z + uy$ over points (y, z) in $\mathcal{G}$,

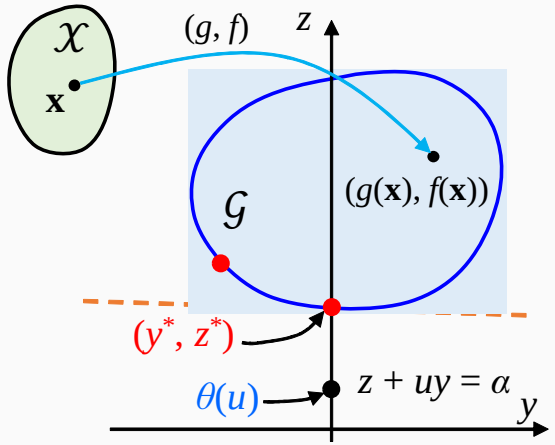
   여기서 $z + uy = \alpha$ 는 기울기가 $-u$ 이고 z 축과 만나는 점이 α 인 직선식이다.

2. $\mathcal{G}$ 에 대해서 $\theta(u) = z + uy$ 를 최소화하기 위해서 $\mathcal{G}$ 와의 접촉을 유지하면서 직선 $z + uy = \alpha$ 를 평행 이동해 내려가면, 최후로 얻어지는 z-축 절편값은 주어진 $u \geq 0$ 에 대응하는 $\theta(u)$ 값이다.



Geometric interpretation

3. 마지막으로 dual problem 을 풀기 위해서는, 최후의 z-축 절편 값 $\theta(u)$ 가 최대값이 되는 기울기 $-u$ $(u \geq 0)$ 를 찾아야 한다.
   이러한 직선은 기울기가 −u* 이고 점 (y*, z*) 에서 집합 $\mathcal{G}$ 를 지지(support)한다.
   그러므로, dual problem 의 해는 $-u^*$ 이며 최적 dual objective value는 z^* 이다.



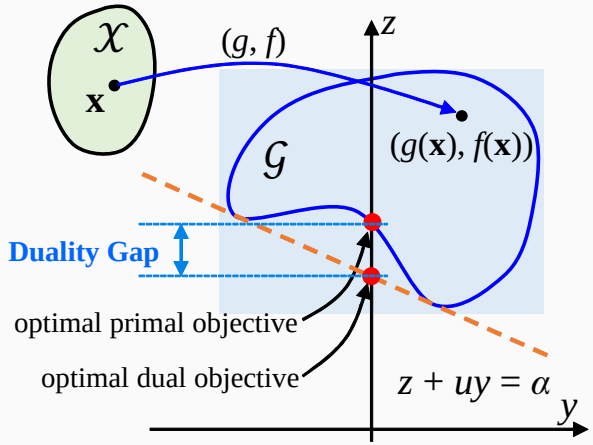
- Primal problem 과 dual problem 의 최적해가 동일한 경우, duality gap 이 없다고 말한다 (**strong duality**).

- 동일하지 않은 경우는 duality gap 이 존재한다고 말한다 (**weak duality**).

- 적절한 convexity condition 들이 충족되면 primal 과 dual 최적화 문제에 duality gap 이 없다.

- [우측 그림] 집합 $\mathcal{G}$ 의 nonconvexity 로 인한 Duality Gap

$$f(\mathbf{x}) \geq \theta(\mathbf{u})$$



Karush-Kuhn-Tucker (KKT) conditions

the following four conditions are called KKT conditions (for a problem with differentiable f_i, h_i)

1. primal constraints: $f_i(x) \leq 0$ i = 1, ..., m, $h_i(x) = 0$, i = 1, ..., p
2. dual constraints: $\lambda \geq 0$
3. complementary slackness: $\lambda_i f_i(x) = 0, i = 1, ..., m$
4. gradient of Lagrangian with respect to x vanishes:

$$\nabla f_0(x) + \sum_{i=1}^{m} \lambda_i \nabla f_i(x) + \sum_{i=1}^{p} \nu_i \nabla h_i(x) = 0 \tag{31}$$

if strong duality holds and x, λ, ν are optimal, then they must satisfy the KKT conditions

SVM: Theory

Maximum margin classifier

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

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \tag{32}$$

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

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

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.

Maximum margin classifier: optimality criterion

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

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)}{\|\mathbf{w}\|}. \tag{33}$$

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

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min[t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)] \right\} \tag{34}$$

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

Dual problem for convex optimiztion

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

In order to solve this constrained optimization problem, we introduce Lagrange multipliers $a_n \geq 0$, with one multiplier a_n for each of the constraints, giving the Lagrangian function

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n\{t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) - 1\} \tag{35}$$

where $\mathbf{a} = (a_1, ..., a_N)^T$. Note the minus sign in front of the Lagrange multiplier term, because we are minimizing with respect to $\mathbf{w}$ and b, and maximizing with respect to $\mathbf{a}$.

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

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \tag{36}$$

$$0 = \sum_{n=1}^{N} a_n t_n \tag{37}$$

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

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{38}$$

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

$$a_n \geq 0, \qquad n = 1, ..., N \tag{39}$$

$$\sum_{n=1}^{N} a_n t_n = 0. \tag{40}$$

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

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

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \tag{41}$$

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

$$a_n \geq 0 \tag{42}$$

$$t_n(\mathbf{x}_n) - 1 \geq 0 \tag{43}$$

$$a_n\{t_n y(\mathbf{x}_n) - 1\} = 0 \tag{44}$$

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.

KKT condition: complementary slackness

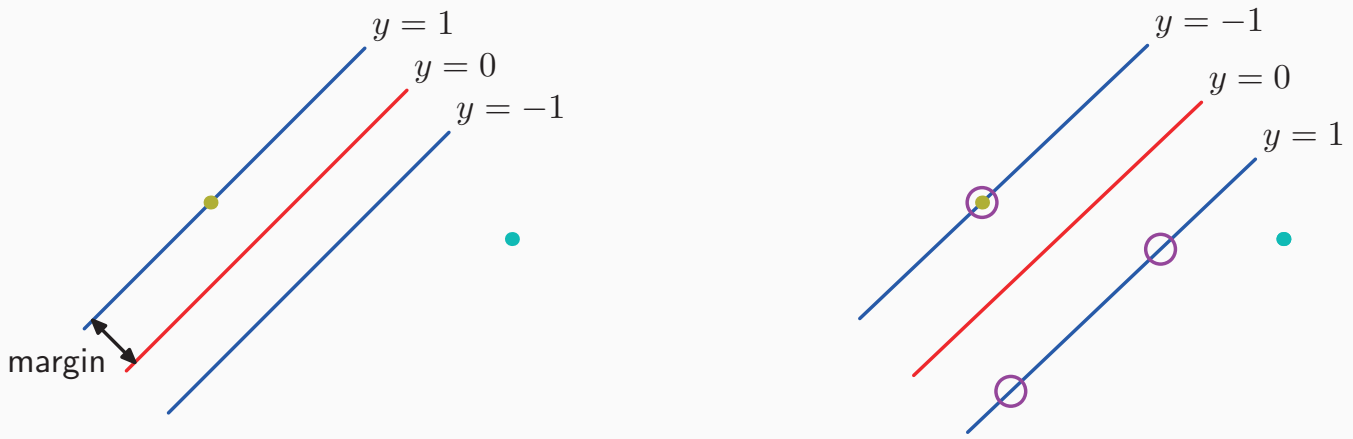
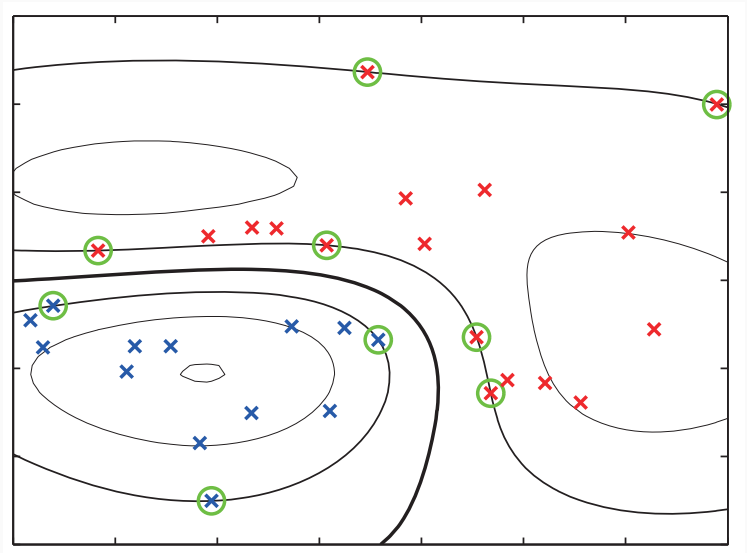


Figure 7.1 The margin is defined as the perpendicular distance between the decision boundary and the closest of the data points, as shown on the left figure. Maximizing the margin leads to a particular choice of decision boundary, as shown on the right. The location of this boundary is determined by 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.



Hard margin SVM: Theory

Maximum margin classifier

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

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \tag{45}$$

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

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

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.

Maximum margin classifier: optimality criterion

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

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)}{\|\mathbf{w}\|}. \tag{46}$$

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

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min[t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)] \right\} \tag{47}$$

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

Dual problem for convex optimization

Primal optimization problem for hard SVM

$$\text{maximize} \quad \frac{1}{\|\mathbf{w}\|} \min[t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)] \tag{48}$$

$$\text{subject to} \quad t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) \geq 0 \tag{49}$$

$$\tag{50}$$

Equivalently,

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{51}$$

$$\text{subject to} \quad t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) \geq 1 \tag{52}$$

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

In order to solve this constrained optimization problem, we introduce Lagrange multipliers $a_n \geq 0$, with one multiplier a_n for each of the constraints, giving the Lagrangian function

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n\{t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) - 1\} \tag{53}$$

where $\mathbf{a} = (a_1, ..., a_N)^T$. Note the minus sign in front of the Lagrange multiplier term, because we are minimizing with respect to $\mathbf{w}$ and b, and maximizing with respect to $\mathbf{a}$.

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

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \tag{54}$$

$$0 = \sum_{n=1}^{N} a_n t_n \tag{55}$$

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

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{56}$$

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

$$a_n \geq 0, \qquad n = 1, ..., N \tag{57}$$

$$\sum_{n=1}^{N} a_n t_n = 0. \tag{58}$$

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

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

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \tag{59}$$

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

$$a_n \geq 0 \tag{60}$$

$$t_n y(\mathbf{x}_n) - 1 \geq 0 \tag{61}$$

$$a_n \{t_n y(\mathbf{x}_n) - 1\} = 0 \tag{62}$$

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.

KKT condition: complementary slackness

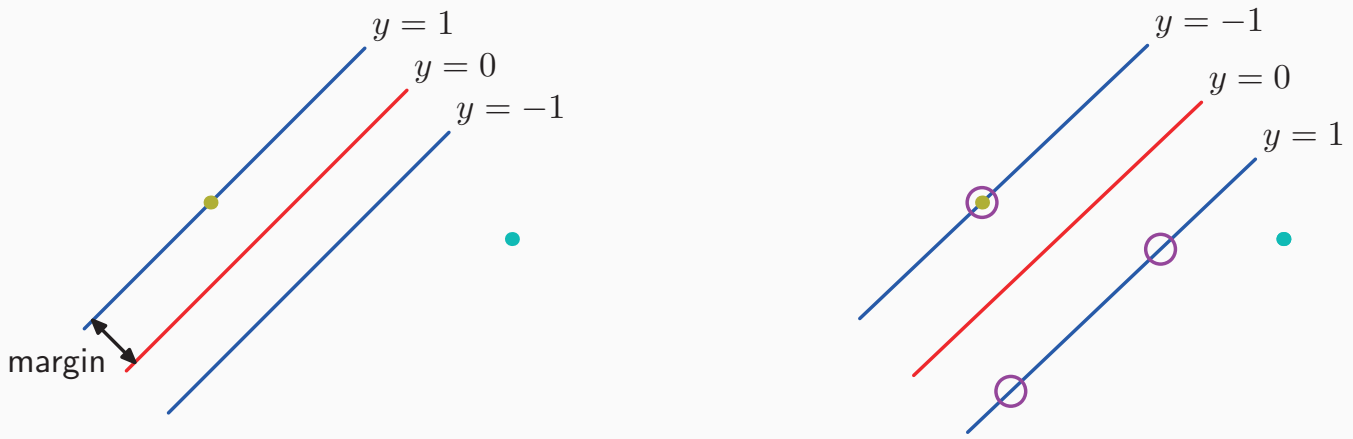
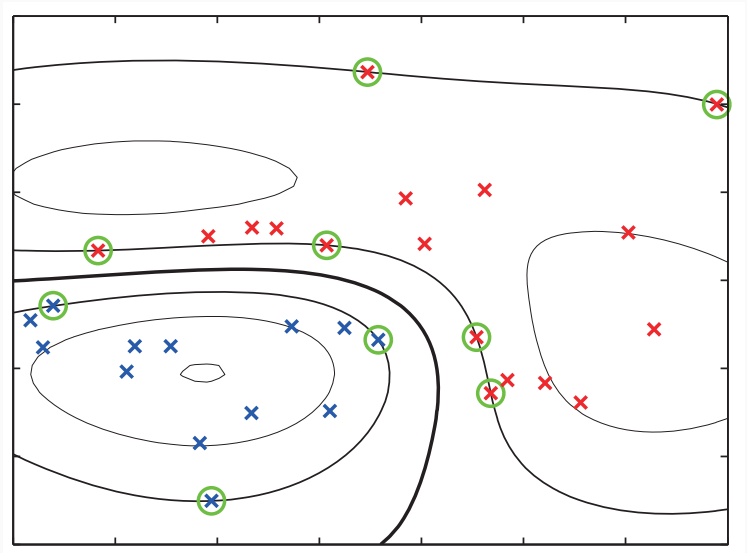


Figure 7.1 The margin is defined as the perpendicular distance between the decision boundary and the closest of the data points, as shown on the left figure. Maximizing the margin leads to a particular choice of decision boundary, as shown on the right. The location of this boundary is determined by 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.



Soft margin SVM: Theory

Slack variables

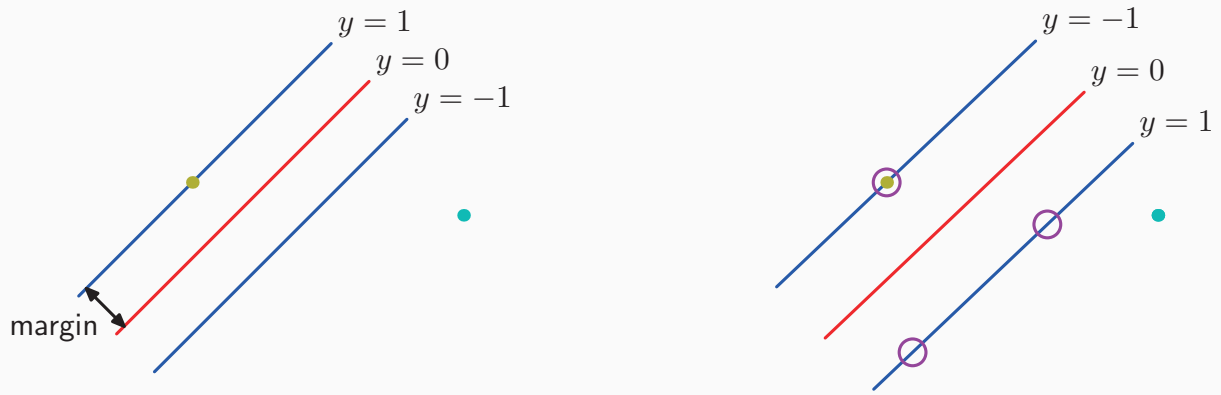
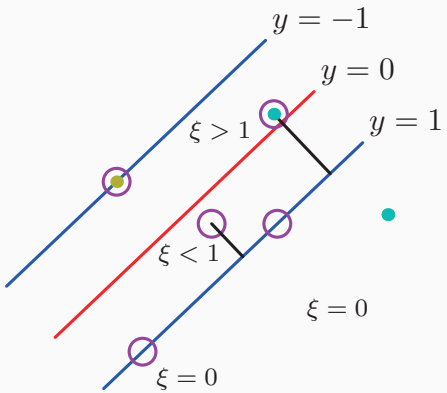


Figure 7.1 The margin is defined as the perpendicular distance between the decision boundary and the closest of the data points, as shown on the left figure. Maximizing the margin leads to a particular choice of decision boundary, as shown on the right. The location of this boundary is determined by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

Figure 7.3 Illustration of the slack variables $\xi_n \geqslant 0$. Data points with circles around them are support vectors.



Slack variables

So far, we have assumed that the training data points are linearly separable in the feature space $\phi(x)$. The resulting support vector machine will give exact separation of the training data in the original input space x, although the corresponding decision boundary will be nonlinear. In practice, however, the class-conditional distributions may overlap, in which case exact separation of the training data can lead to poor generalization.

We therefore need a way to modify the support vector machine so as to allow some of the training points to be misclassified.

Slack variables

We introduce *slack variables*, $\xi_n \geq 0$ where $n = 1, 2, ..., N$, with one slack variable for each training data point. These are defined by $\xi_n = 0$ for data points that are on or inside the correct margin boundary and $\xi_n = |t_n - y(\mathbf{x}_n)|$ for other points.

Thus a data point that is on the decision boundary $y(\mathbf{x}_n) = 0$ will have $\xi_n = 1$, and points with $\xi_n > 1$ will be misclassified.

The exact classification constraints are then replaced with

$$t_n y(\mathbf{x}_n) \geq 1 - \xi_n, \qquad n = 1, ..., N \tag{63}$$

in which the slack variables are constrained to satisfy $\xi_n \geq 0$.

Slack variables

Points for which $0 < \xi_n \leq 1$ lie inside the margin, but on the correct side of the decision boundary, and those data points for which $\xi_n > 1$ lie on the wrong side of the decision boundary and are misclassified. This is sometimes described as relaxing the hard margin constraint to give a *soft margin* and allows some of the training set data points to be misclassified.

Soft margin SVM

Our goal is to maximize the margin while softly penalizing points that lie on the wrong side of the margin boundary. We therefore

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n \tag{64}$$

$$\text{subject to} \quad t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) \geq 1 - \xi_n \tag{65}$$

$$\xi_n \geq 0. \tag{66}$$

The parameter C is analogous to a regularization coefficient because it controls the trade-off between minimizing training errors and controlling model complexity.

FYI, hard margin SVM

$$\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 \tag{67}$$

$$\text{subject to} \quad t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) \geq 1 \tag{68}$$

Soft margin SVM: Lagrangian dual problem

The corresponding Lagrangian is given by

$$L(\mathbf{w}, b, \mathbf{a}, \mu_n) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N} a_n\{t_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N}\mu_n\xi_n \tag{69}$$

where $\{a_n \geq 0\}$ and $\{\mu_n \geq 0\}$ are Lagrangian multipliers. The corresponding set of KKT conditions are given by

$$a_n \geq 0 \tag{70}$$

$$t_n y(\mathbf{x}_n) - 1 + \xi_n \geq 0 \tag{71}$$

$$a_n(t_n y(\mathbf{x}_n) - 1 + \xi_n) = 0 \tag{72}$$

$$\mu_n \geq 0 \tag{73}$$

$$\xi_n \geq 0 \tag{74}$$

$$\mu_n\xi_n = 0 \tag{75}$$

where $n = 1, 2, ..., N$.

Soft margin SVM: Lagrangian dual problem

We now optimize out $\mathbf{w}$, b, and $\{\xi_n\}$ making use of the definition of $y(x)$ to give

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \Longrightarrow \quad \mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \tag{76}$$

$$\frac{\partial L}{\partial b} = 0 \quad \Longrightarrow \quad \sum_{n=1}^{N} a_n t_n = 0 \tag{77}$$

$$\frac{\partial L}{\partial \xi_n} = 0 \quad \Longrightarrow \quad a_n = C - \mu_n \tag{78}$$

We obtain the dual Lagrangian in the form

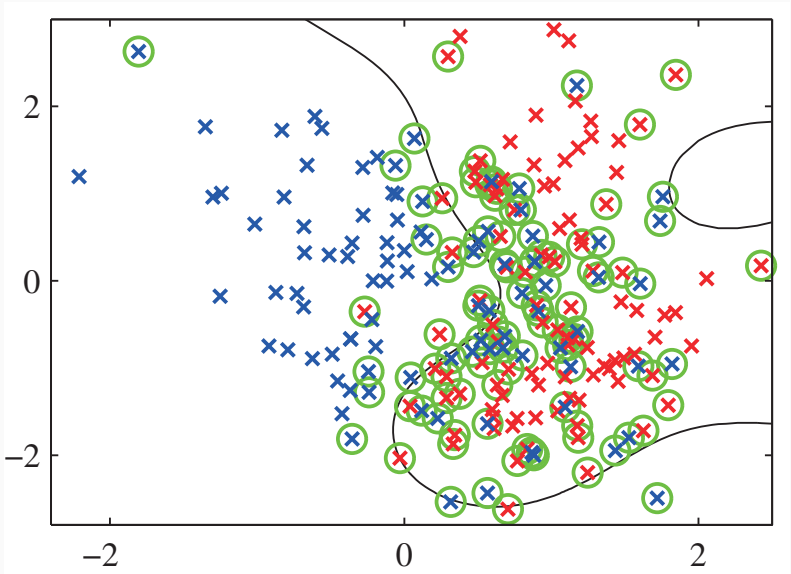
$$\text{maximize} \quad \tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \tag{79}$$

$$\text{subject to} \quad 0 \geq a_n \geq C \tag{80}$$

$$\sum_{n=1}^{N} a_n t_n = 0 \tag{81}$$

Soft margin SVM

Figure 7.4 Illustration of the ν-SVM applied to a nonseparable data set in two dimensions. The support vectors are indicated by circles.



Kernel Method

Kernel method

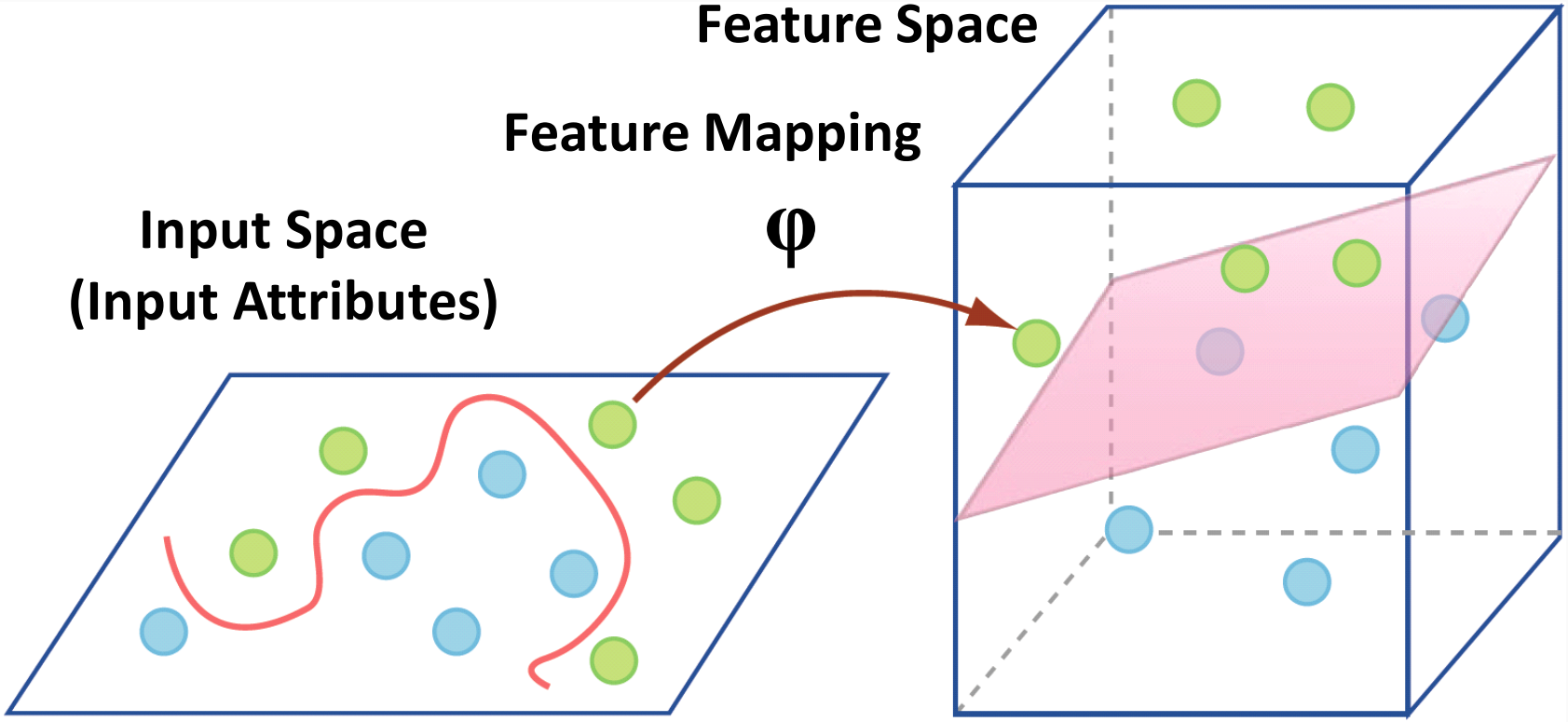


Image by MIT OpenCourseWare

Kernel method

Kernel function: Kernel function이란 어떤 feature space에서 feature의 inner product와 동등한 함수를 말함. Feature mapping $\boldsymbol{\phi}(\mathbf{x})$에 대한 kernel function은 다음과 같이 정의할 수 있음.

$$k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}') \tag{82}$$

Similarity: Kernel function $k(\mathbf{x}, \mathbf{x}')$는 $\mathbf{x}$와 $\mathbf{x}'$ 간의 similarity로 볼 수 있음.

또한 일반적으로 feature mapping을 통한 $\boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}')$에 비해 $k(\mathbf{x}, \mathbf{x}')$의 계산이 훨씬 간단함. 예를 들어

$$k(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T \mathbf{x}')^2 \tag{83}$$

$$\boldsymbol{\phi}(\mathbf{x}) = \begin{bmatrix} 1 & x_1^2 & \sqrt{2}x_1 x_2 & x_2^2 & \sqrt{2}x_1 & \sqrt{2}x_2 \end{bmatrix} \tag{84}$$

Construction of kernel

임의의 kernel function 대해서 $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{x}')$가 성립하는 feature mapping이 존재하는지 검사하는 것은 매우 까다로움.

Mercer's theorem

Every positive semi-definite symmetric function is a kernel.

다음과 같이 정의되는 Gram matrix K가 positive semi-definite symmetric matrix면 function k는 kernel function임.

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \cdots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \tag{85}$$

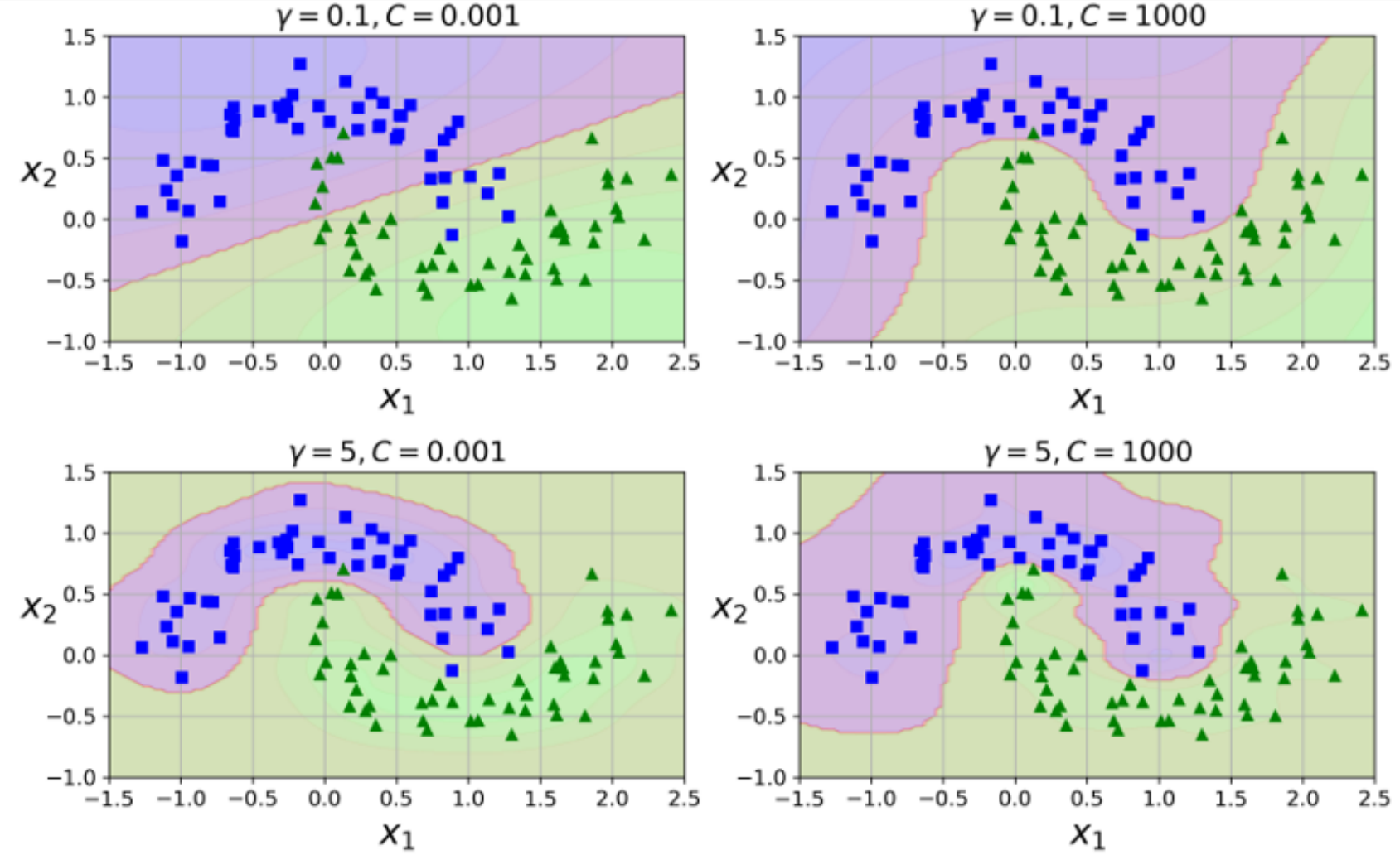
Gaussian radial basis function (RBF) kernel

Gaussian RBF kernel

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2) \tag{86}$$

- kernel value는 $\mathbf{x}'$가 landmark $\mathbf{x}$와 가까울수록 큰 값
- γ 값이 커지는 경우 landmark에서 멀어질수록 kernel 값이 급격하게 작아짐 = 영향력의 범위가 좁아짐

Hyperparameter of soft SVM-RBF



SVM: strength and limitations

장점

- 강력하고 우수한 이론을 바탕으로 함
- 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

SVM: strength and limitations

단점

- 노이즈에 민감: 비교적 적은 수의 잘못된 label로 성능이 심각하게 악화
- 커널 함수의 선택/구축하는 방법에 대한 정리된 원칙이 없음
- hyperparameter C의 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)