

# Lecture 12: Support Vector Machine II

[SCS4049-02] Machine Learning and Data Science

---

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

# Tentative schedule

week	topic	date (수 / 월)
1	Machine Learning Introduction & Basic Mathematics	09.02 / 09.07
2	Python Practice I & Regression	09.09 / 09.14
3	AI Department Seminar I & Clustering I	09.16 / 09.21
4	Clustering II & Classification I	09.23 / 09.28
5	Classification II	(추석) / 10.05
6	Python Practice II & Support Vector Machine I	10.07 / 10.12
7	Support Vector Machine II & Ensemble Learning and Random Forest	10.14 / 10.19
8	(휴강) & <b>Mid-term exam</b>	10.21 / <b>10.26</b>
9	Neural networks	10.28 / 11.02
10	Backpropagation	11.04 / 11.09
11	Convolutional Neural Network	11.11 / 11.16
12	Model Optimization	11.18 / 11.23
13	Recurrent Neural network	11.25 / 11.30
14	Autoencoders	12.02 / 12.07
15	<b>Final exam</b>	(휴강) / <b>12.14</b>

## Mid-term exam: 비대면

- 일시: 10월 26일 (월), 10:00 - 12:00 (2시간, 연장 없음)
- 장소: 비대면 webex (수업 미팅룸)
- 범위
  - 시험 전까지의 모든 강의
  - 강의자료, 수업필기, 강의내용, 숙제 등
- 방식: 서술형 위주, closed book
  - ※ 프로그래밍은 범위에 포함되지 않음
- 준비물
  - webex용 카메라 (웹캠, 노트북, 휴대폰 등)
  - 답안용 A4 용지
  - 스마트폰 스캔앱 (Adobe Scan 등)
  - 카메라, 스마트폰, A4 용지가 필요한 경우 교수에게 문의 (s.park@dgu.edu)

# Mid-term exam: 비대면

- 응시 방법
  - 책상, 손, 얼굴이 나오도록 카메라를 설치
  - 모든 수강생이 확인된 후, eclass로 시험지 배포
  - 답안지 첫 페이지: 학번, 학과, 이름, 총 페이지 수를 반드시 기입
  - 답안지 각 페이지: 해당 페이지 수를 반드시 기입
- 제출 방법
  - 시험 종료까지 미팅룸을 먼저 나갈 수 없음
  - 시험시간 종료 후 본인이 작성한 모든 답지를 카메라에 차례로 보여줌
  - 스캔앱으로 답지를 스캔하여 pdf 파일을 메일로 제출 (s.park@dgu.edu)
  - 10월 30일 (금)까지 만해관 210호로 답안지 실물 제출
- 문의 사항: 담당 교수 박성식
  - 이메일: s.park@dgu.edu
  - 사무실: 만해관 210호
  - 전화: 02) 2260-3784

# 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

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \quad (1)$$

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  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ , 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  $y(\mathbf{x})$

# 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

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

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

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

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

# Dual problem for convex optimization

Primal optimization problem for hard SVM

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

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

$$(6)$$

Equivalently,

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

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

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

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

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N a_n \{t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - 1\} \quad (9)$$

where  $\mathbf{a} = (a_1, \dots, a_N)^T$ . 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

$$\mathbf{w} = \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n) \quad (10)$$

$$0 = \sum_{n=1}^N a_n t_n \quad (11)$$

# 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

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \quad (12)$$

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

$$a_n \geq 0, \quad n = 1, \dots, N \quad (13)$$

$$\sum_{n=1}^N a_n t_n = 0. \quad (14)$$

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

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

## 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

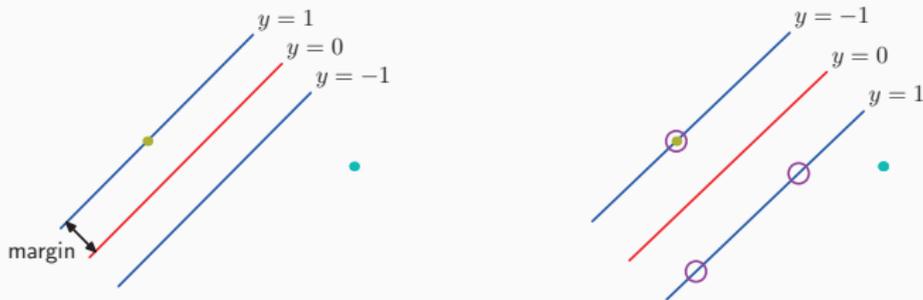
$$a_n \geq 0 \tag{16}$$

$$t_n y(\mathbf{x}_n) - 1 \geq 0 \tag{17}$$

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

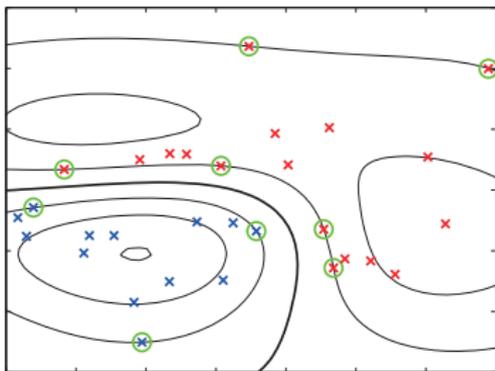
Thus for every data point, either  $a_n = 0$  or  $t_n y(\mathbf{x}_n) = 1$ . Any data point for which  $a_n = 0$  will not appear in the sum and hence plays no role in making predictions for new data points. The remaining data points are called *support vectors*, and because they satisfy  $t_n y(\mathbf{x}_n)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

# 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.

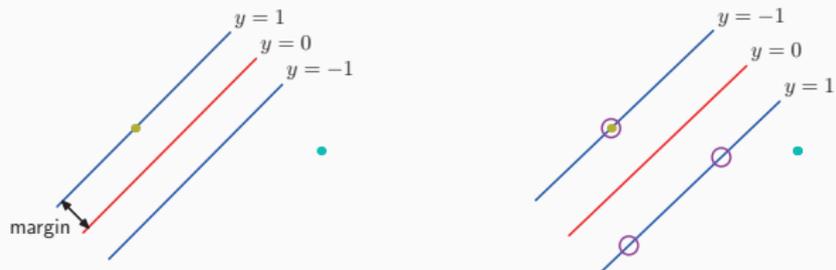
**Figure 7.2** Example of synthetic data from two classes in two dimensions showing contours of constant  $y(\mathbf{x})$  obtained from a support vector machine having a Gaussian kernel function. Also shown are the decision boundary, the margin boundaries, and the support vectors.



# Soft margin SVM: Theory

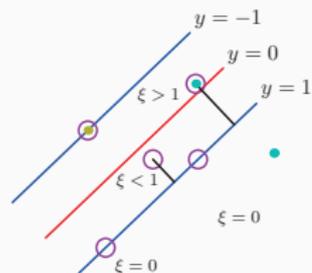
---

# 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  $\xi_n \geq 0$ . 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  $n = 1, 2, \dots, N$ , 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

$$t_n y(\mathbf{x}_n) \geq 1 - \xi_n, \quad n = 1, \dots, N \quad (19)$$

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

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

$$\text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n \quad (20)$$

$$\text{subject to } t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1 - \xi_n \quad (21)$$

$$\xi_n \geq 0. \quad (22)$$

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

$$\text{minimize } \frac{1}{2} \|\mathbf{w}\|^2 \quad (23)$$

$$\text{subject to } t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1 \quad (24)$$

## Soft margin SVM: Lagrangian dual problem

The corresponding Lagrangian is given by

$$L(\mathbf{w}, b, \mathbf{a}, \mu_n) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n - \sum_{n=1}^N a_n \{t_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^N \mu_n \xi_n \quad (25)$$

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

$$a_n \geq 0 \quad (26) \qquad \mu_n \geq 0 \quad (29)$$

$$t_n y(\mathbf{x}_n) - 1 + \xi_n \geq 0 \quad (27) \qquad \xi_n \geq 0 \quad (30)$$

$$a_n (t_n y(\mathbf{x}_n) - 1 + \xi_n) = 0 \quad (28) \qquad \mu_n \xi_n = 0 \quad (31)$$

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

## 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

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \implies \mathbf{w} = \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n) \quad (32)$$

$$\frac{\partial L}{\partial b} = 0 \implies \sum_{n=1}^N a_n t_n = 0 \quad (33)$$

$$\frac{\partial L}{\partial \xi_n} = 0 \implies a_n = C - \mu_n \quad (34)$$

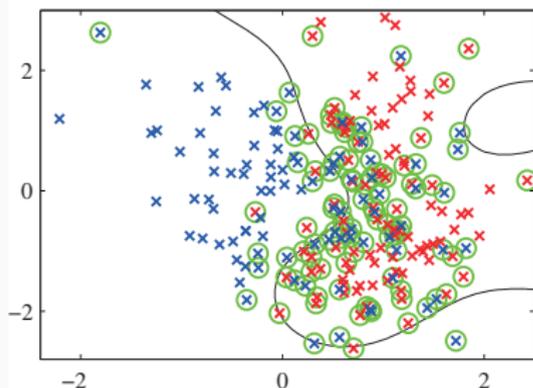
We obtain the dual Lagrangian in the form

$$\text{maximize } \tilde{L}(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \quad (35)$$

$$\text{subject to } 0 \geq a_n \geq C \quad (36)$$

$$\sum_{n=1}^N a_n t_n = 0 \quad (37)$$

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



# Kernel Method

---

# Kernel method

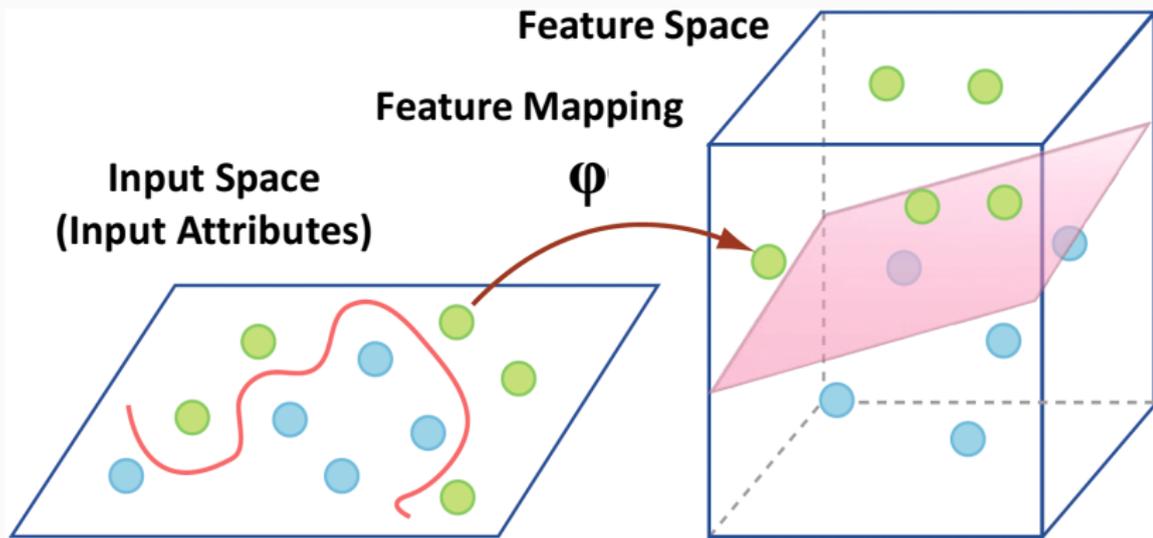


Image by MIT OpenCourseWare

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

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') \quad (38)$$

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

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

$$k(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T \mathbf{x}')^2 \quad (39)$$

$$\phi(\mathbf{x}) = [ 1 \quad x_1^2 \quad \sqrt{2}x_1x_2 \quad x_2^2 \quad \sqrt{2}x_1 \quad \sqrt{2}x_2 ] \quad (40)$$

# Construction of kernel

임의의 kernel function 대해서  $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ 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  $K$ 가 positive semi-definite symmetric matrix면 function  $k$ 는 kernel function임.

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \cdots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \quad (41)$$

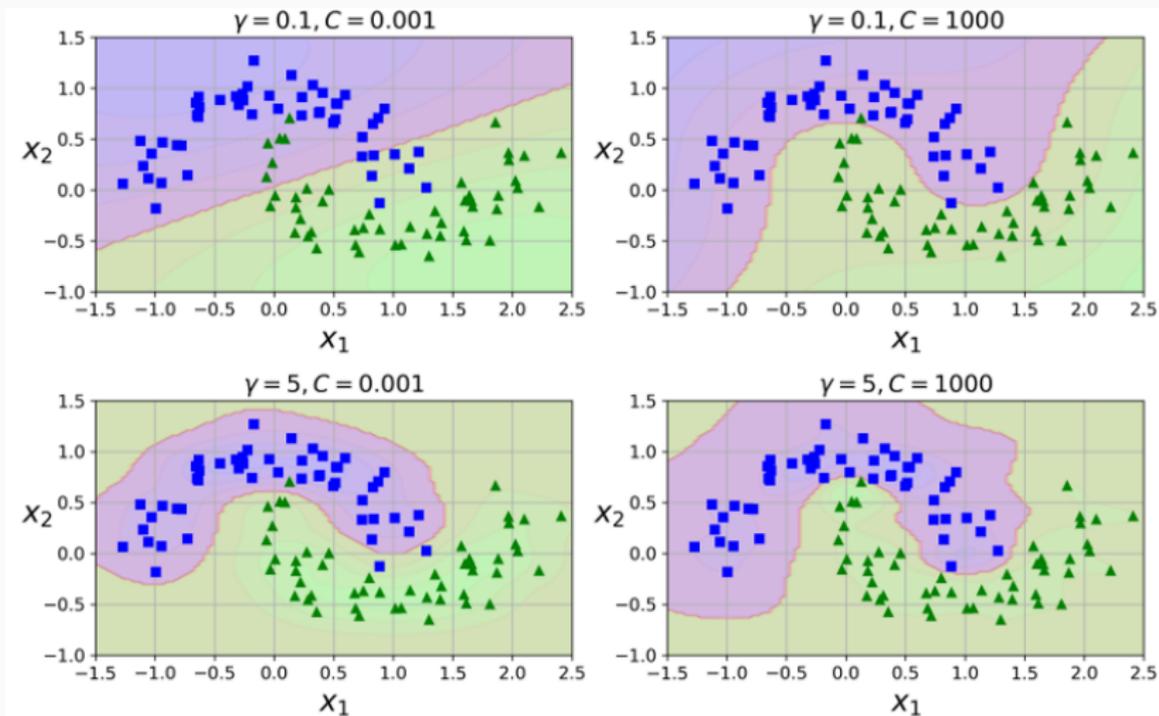
# Gaussian radial basis function (RBF) kernel

Gaussian RBF kernel

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2) \quad (42)$$

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

# Hyperparameter of soft SVM-RBF



## 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

## 단점

- 노이즈에 민감: 비교적 적은 수의 잘못된 label로 성능이 심각하게 악화
- 커널 함수의 선택/구축하는 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)

Due: 10월 20일 화요일, 23시 59분까지

- 컴퓨터로 작성(latex, word, ppt, 한글 등)해서 pdf로 업로드해주세요.
- 또는 손으로 작성한 파일을 스캔앱(Adobe scan, Office lens 등)을 써서 pdf로 업로드해주세요.

1. 행렬의 singular value decomposition (SVD)에 대해 공부해보고 다음에 대해 설명하세요.

- 1.1 Linear transformation 관점에서 행렬  $A$ 를  $A = U\Sigma V^T$ 로 SVD 했을 때, 주어진 벡터  $x$ 를 어떤 단계를 거쳐  $y = Ax$ 로 변환하는지 (10점)
- 1.2  $A = U\Sigma V^T$ 에서 각 행렬  $U, V$ 이 행렬의 column space 및 row space와 어떤 물리적인 관계를 갖는지 (5점)
- 1.3 행렬의 singular value와  $\Sigma$  행렬의 관계 (5점)
- 1.4 행렬의 positive definiteness와 singular value 사이의 관계 (10점)