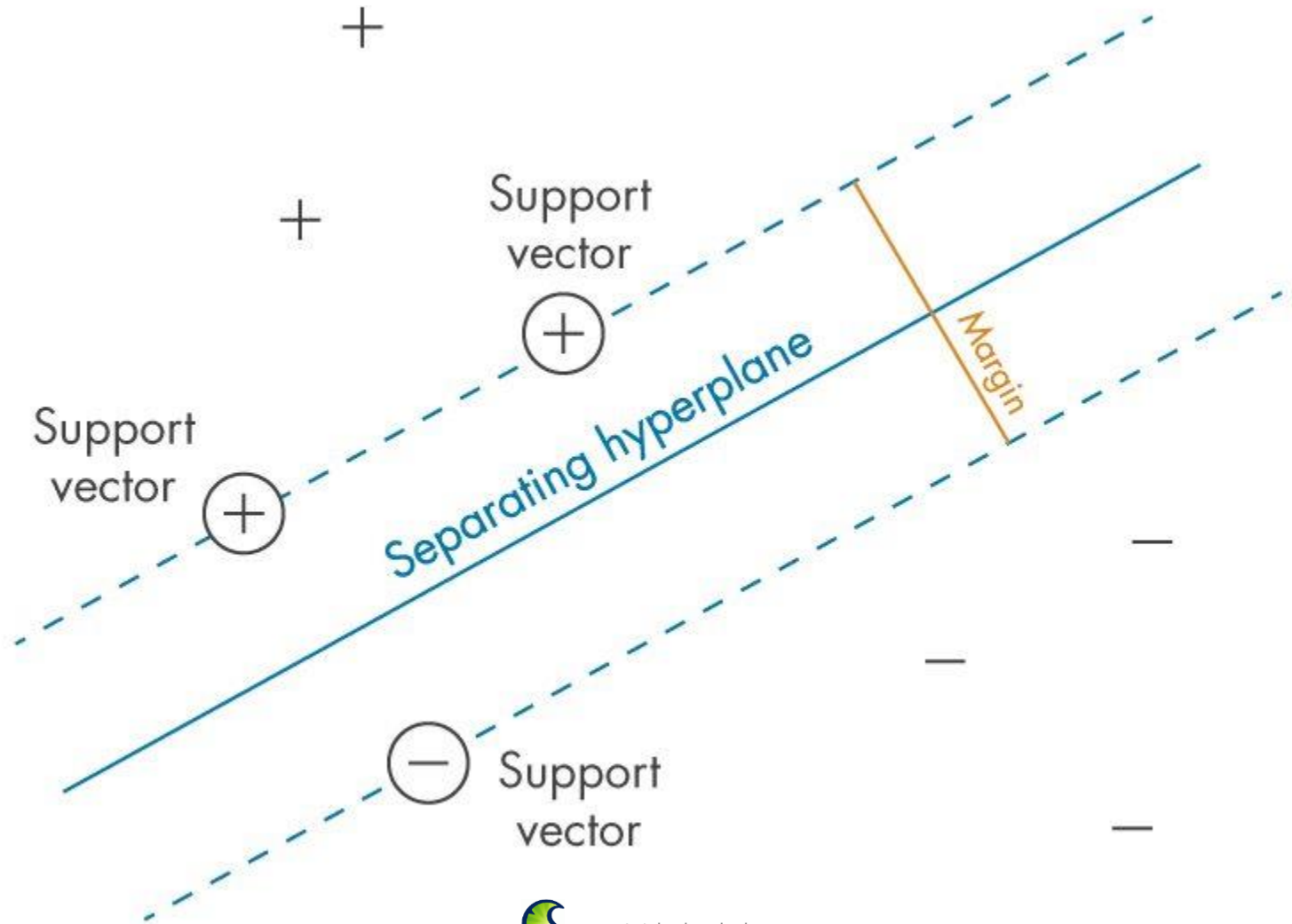


Support Vector Machine



Data Science School (by Dr. Dohyeong Kim, datascienceschool.net)

데이터 사이언스 스쿨
알파
강의
서비스
도서

웹사이트 소개
로그인

05.04 분산 분석과 모형 성능

source	degrees of freedom	mean square	F statistic
Regression	2	10.00	10.00
Residual	978	0.01	
Total	980		

08.06 VAE

04.03 적분

$$\int_a^b f(x) dx$$

10.02 조건부엔트로피

개발환경

12주 전

00.01 커맨드 라인 인터페이스

61주 전

02.04 도커 컨테이너와 파일을 공유하기

61주 전

02.03 도커 조건단 사용법

61주 전

02.02 도커 이미지 설치 및 실행

파이썬

9주 전

9주 전

11주 전

12주 전

덧글

tnsg*** 2020년 4월 30일 2:33 오후
다음 수식을 수렴할 때까지 반복 부분에서

gyuy*** 2020년 4월 30일 12:02 오후
최종 모델 산출에 관하여,,

osh9*** 2020년 4월 27일 9:20 오후
예측이 도움이 되는 경우에 관한 조건부 엔트로피 관련한 예시 관련한 질문입니다.

myna*** 2020년 4월 27일 2:19 오후
p140에서...

관리자 2020년 4월 24일 8:22 오후
답변: 식 4.5.14 부분과 4.5.15 부분이 이해가 되지 않습니다.

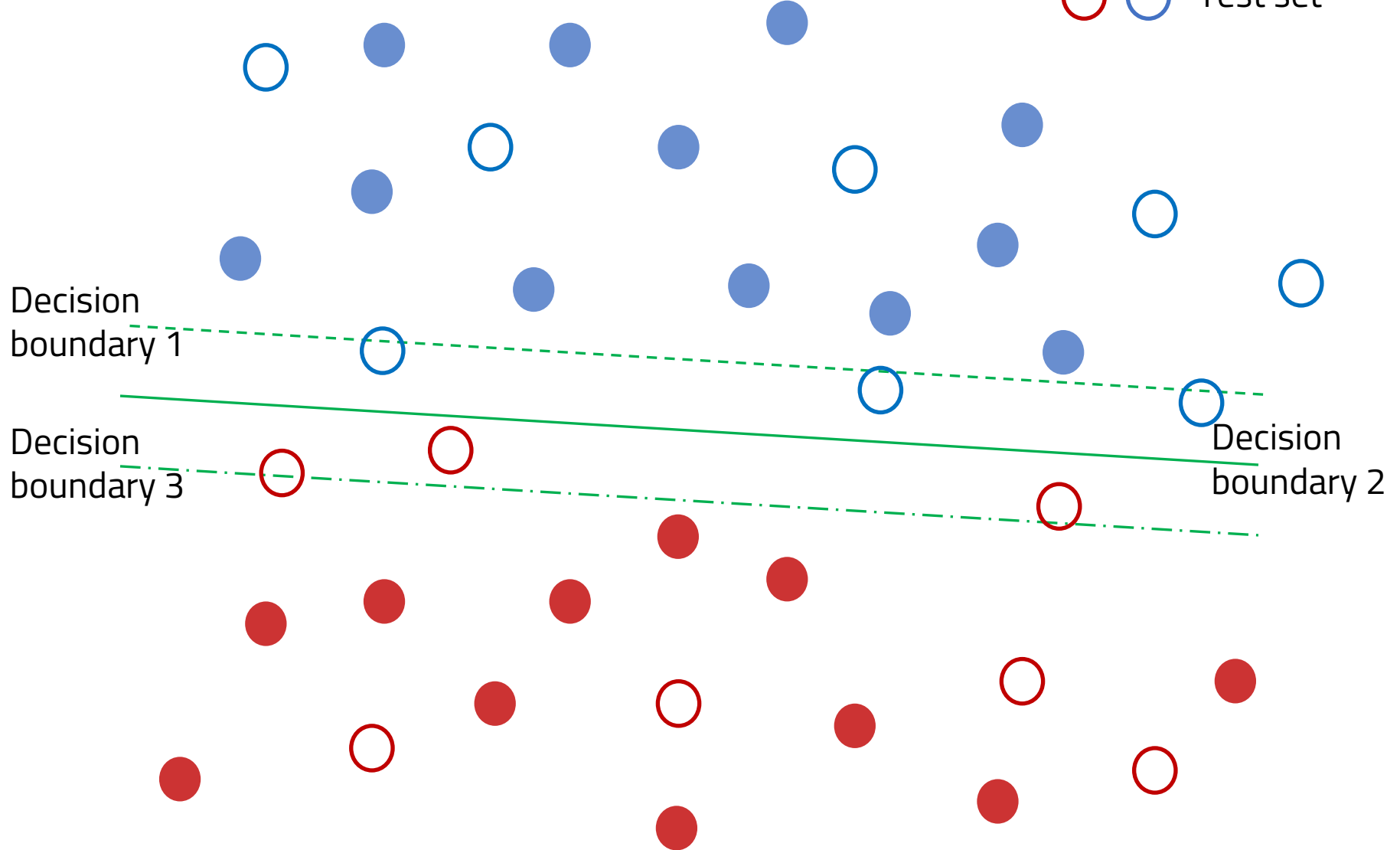
관리자 2020년 4월 24일 8:17 오후
답변: 전체 분산의 법칙을 증명하는 과정에서 질문이 생겼습니다.

관리자 2020년 4월 24일 8:09 오후
답변: ipython_config.py 설정 파일 관련

관리자 2020년 4월 24일 8:08 오후

What is the optimal decision boundary?

● ● Training set
○ ○ Test set



Support Vector Machine

Data

$$\mathcal{D} = \{(x_j, y_j)\}_{j=1, \dots, N}$$

$$y_i = \begin{cases} y_i^+ = +1 \\ y_i^- = -1 \end{cases}$$

$$x_i^+: x_i \in \{x_j\} \text{ such that } y_i = y_i^+$$

$$x_i^-: x_i \in \{x_j\} \text{ such that } y_i = y_i^-$$

Model

$$f(x) = w^T x - w_0$$

such that

$$f(x_i^+) = w^T x_i^+ - w_0 > 0$$

$$f(x_i^-) = w^T x_i^- - w_0 < 0$$

Decision boundary

$$x \text{ such that } f(x) = 0$$

Support vectors

$$x^+: x \in \{x_j^+\} \text{ at which } f(x) \text{ is the smallest of } \{f(x_j^+)\}$$

$$x^-: x \in \{x_j^-\} \text{ at which } f(x) \text{ is the largest of } \{f(x_j^-)\}$$

Constraints

$$f(x^+) = 1 \quad f(x^-) = -1$$

Model (SVM)

$$f(x) = w^T x - w_0$$

$$f(x) > +1$$

$$f(x^+) = +1$$

Decision
boundary

$$f(x) = w^T x - w_0 = 0$$

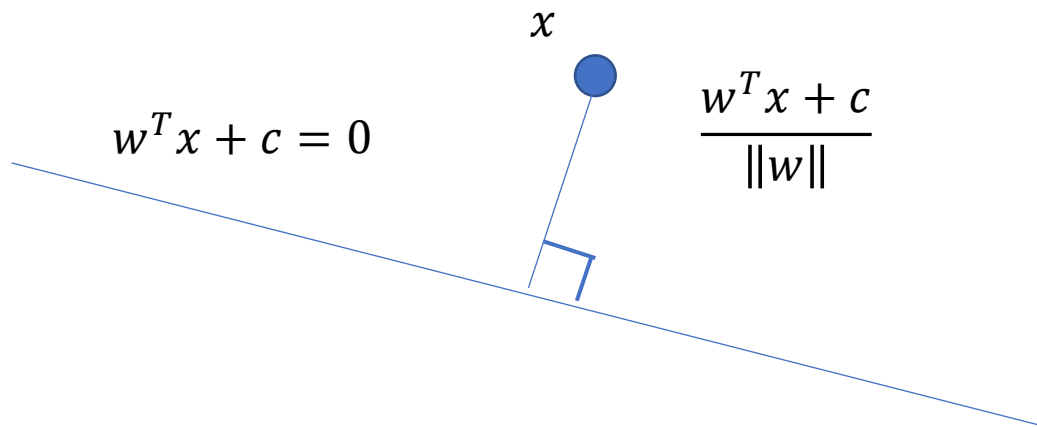
Support
vector x^+

$$f(x^-) = -1$$

Support
vector x^-

$$f(x) < -1$$

Distance from a point to a line



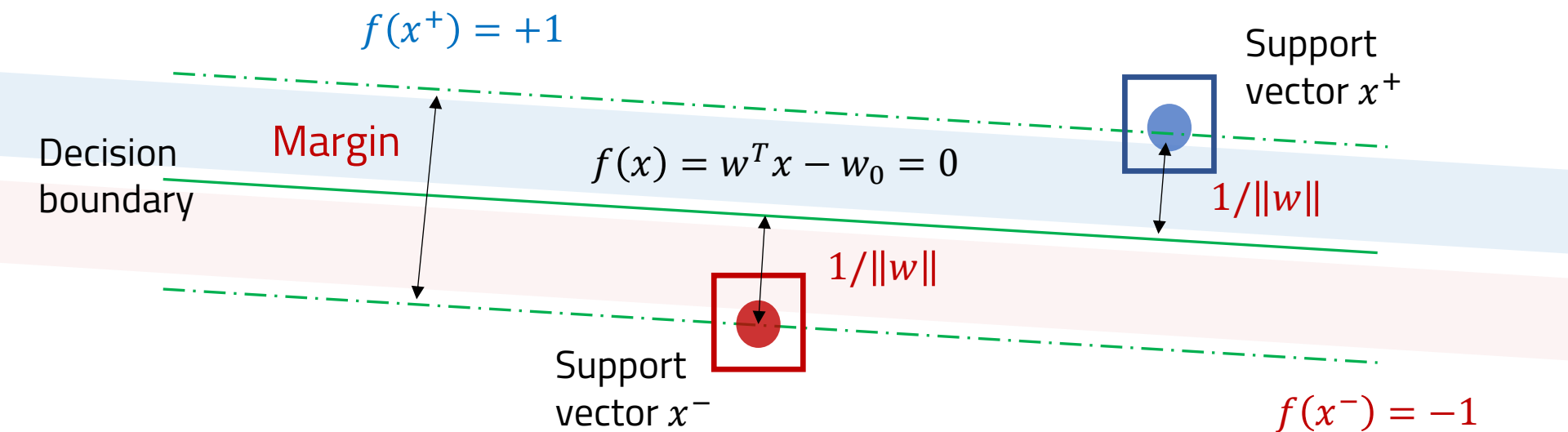
Margin maximization

$$w^* = \arg \max_w \frac{2}{\|w\|}$$

subject to

$$f(x^+) = +1$$

$$f(x^-) = -1$$



Margin maximization with inequality constraints

$$w^* = \arg \max_w \frac{2}{\|w\|}$$

subject to

$$f(x^+) = +1$$
$$f(x^-) = -1$$

$$w^* = \arg \min_w \|w\|^2$$

subject to

$$f(x_i^+) \geq +1$$
$$f(x_i^-) \leq -1$$

Model (SVM)

$$f(x) = w^T x - w_0$$

$$w^* = \arg \min_w \frac{1}{2} w^T w$$

subject to

$$f(x_i^+) y_i^+ \geq +1$$
$$f(x_i^-) y_i^- \geq +1$$

$$w^* = \arg \min_w \frac{1}{2} w^T w$$

subject to

$$1 - f(x_i) y_i \leq 0$$

$J(w)$

Primal Problem

computationally challenging

$g_i(w; x_i, y_i)$

Transformation of the primal problem to its dual problem

$$L(w, \lambda) = J(w) + \sum_{i=1}^N \lambda_i g(w; x_i, y_i)$$

$$= \frac{1}{2} w^T w + \sum_{i=1}^N \lambda_i (1 - y_i (w^T x_i - w_0)) \leq J(w)$$

$$\min_w J(w) \longleftrightarrow \max_{\lambda} \min_w L(w, \lambda)$$

subject to $1 - f(x_i) y_i \leq 0$ subject to KKT condition

substitution

From KKT condition

$$(1) \quad \frac{\partial L}{\partial w} = w - \sum_{i=1}^N \lambda_i y_i x_i = 0 \quad \longrightarrow \quad w = \sum_{i=1}^N \lambda_i y_i x_i$$

$$(2) \quad \frac{\partial L}{\partial w_0} = \sum_{i=1}^N \lambda_i y_i = 0$$

$$(3) \quad \lambda_i \frac{\partial L}{\partial \lambda_i} = 0 \quad \longrightarrow \quad \lambda_i \neq 0 \quad \text{for } x = x^+, x^- , \quad \lambda_i = 0 \quad \text{otherwise}$$

$$(4) \quad \lambda_i \geq 0$$

Constructing a dual problem from the primal problem

$$\max_{\lambda} L(\lambda; x_i, y_i)$$

subject to

$$L(\lambda; x_i, y_i) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j x_i^T x_j$$

$$(2) \quad \sum_{i=1}^N \lambda_i y_i = 0 \quad (4) \quad \lambda_i \geq 0$$



Dual Problem

computationally easier



Matrix-vector form

$$L(\lambda; x_i, y_i) = c^T \lambda + \frac{1}{2} \lambda^T Q \lambda \quad \text{where } c^T = [1, 1, \dots, 1], Q = -q^T q, \\ q = [q_1 \quad q_2 \quad \dots \quad q_N], q_i = y_i x_i$$

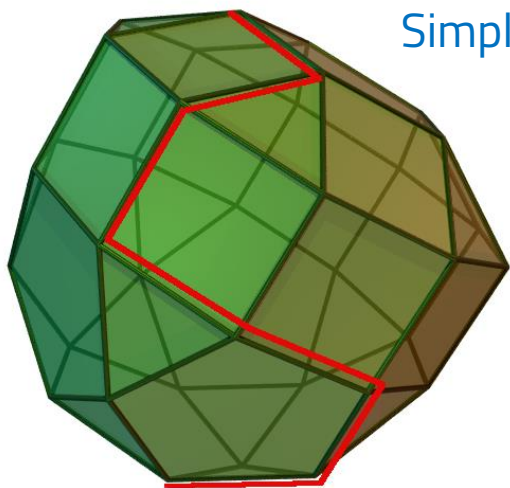
$$(2) \quad A \lambda = 0 \quad \text{where } A = \text{diag}([y_1, y_2, \dots, y_N]) \quad (4) \quad \lambda \geq 0$$

Quadratic programming

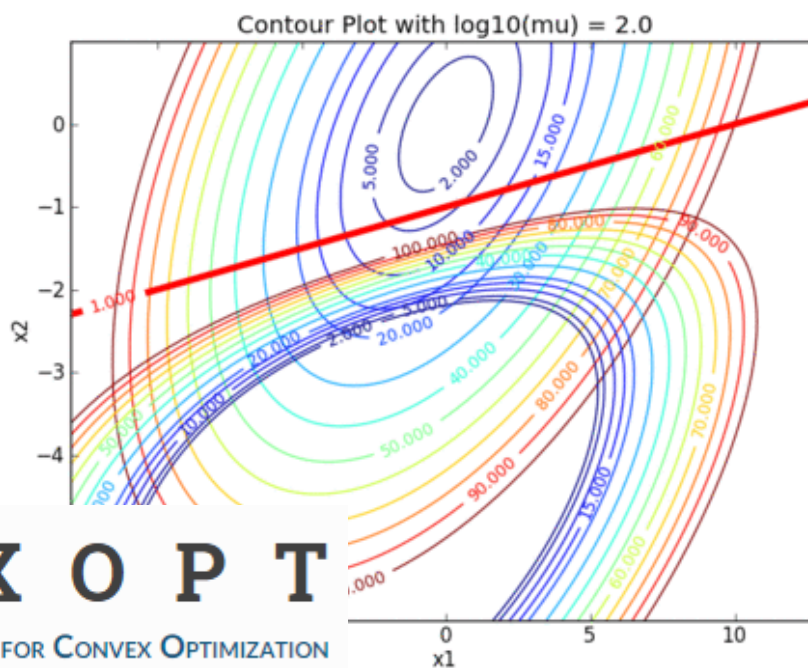
$$L(\{(x_i, y_i)\}, \lambda) = c^T \lambda + \frac{1}{2} \lambda^T Q \lambda \quad \text{where } c^T = [1, 1, \dots, 1], Q = -q^T q, \\ q = [q_1 \quad q_2 \quad \dots \quad q_N], q_i = y_i x_i$$

subject to

$$A\lambda = 0 \quad \text{where } A = \text{diag}([y_1, y_2, \dots, y_N]) \quad , \quad \lambda \geq 0$$



Simplex method



C V X O P T

PYTHON SOFTWARE FOR CONVEX OPTIMIZATION

<https://cvxopt.org/userguide/index.html>

Interior point method

Support Vector Machine

$$f(x) = w^T x - w_0$$



$$w = \sum_{i=1}^N \lambda_i y_i x_i$$

$$f(x) = w^T x - w_0 = \lambda^+(x^+)^T x - \lambda^-(x^-)^T x - w_0$$

$$\Rightarrow w = \lambda^+ x^+ - \lambda^- x^-$$

$$f(x^+) = w^T x^+ - w_0 = +1$$

$$f(x^-) = w^T x^- - w_0 = -1$$

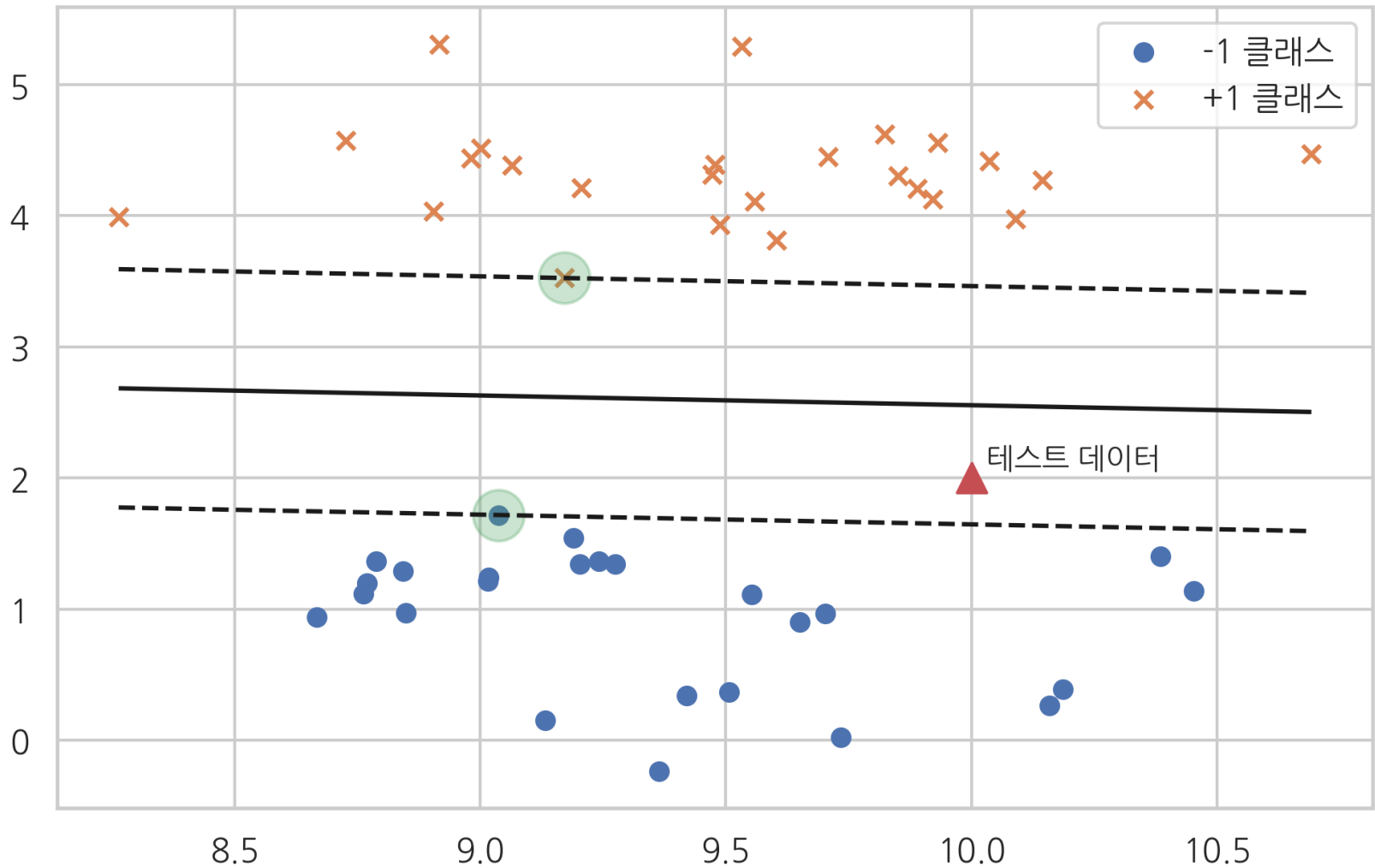
$$\Rightarrow w_0 = \frac{1}{2} w^T (x^+ + x^-)$$

$$f(x) = \lambda^+ \langle x, x^+ \rangle - \lambda^- \langle x, x^- \rangle - w_0$$

$\langle \cdot, \cdot \rangle$: inner product

similarity
measure

SVM 예측 결과



```
from sklearn.datasets import make_blobs
X, y = make_blobs(n_samples=50, centers=2, cluster_std=0.5, random_state=4)
y = 2 * y - 1
```

Support Vector Machine (SVM) with Scikit-Learn



```
from sklearn.svm import SVC
```

```
model = SVC(kernel='linear', C=1e10).fit(X, y)
```

SVC : support vector classifier

```
model.support_
```

the index of the support vector for each class

```
y[model.support_]
```

the target values y^+ , y^- of the support vectors

```
model.support_vectors_
```

the feature values x^+ , x^- of the support vectors

```
x_new = [10, 2]
```

```
y_pred = model.decision_function(x_new)
```

the predicted value y_{pred} for a new feature value x_{new}

Exercise 1

Let's solve the iris problem with a support vector machine. Let's solve it by changing it to a binary classification problem **using only the species 'versicolor', 'virginica'**. Set the kernel type and the slack variable C as 'linear' and $1e10$, respectively.

Not Linearly Separable Data and Slack Variable ξ_i

Model (SVM)

$$f(x) = w^T x - w_0$$

$$f(x) > +1 - \boxed{\xi_i}$$

$$\geq 0$$

$$f(x^+) = +1$$

$$f(x) = 0$$

Support
vector x^+

$$f(x^-) = -1$$

Support
vector x^-

$$f(x) < -1 + \boxed{\xi_i}$$

$$\geq 0$$

Lagrangian multiplier approach with inequality constraints

Model (SVM)

$$f(x) = w^T x - w_0$$

$$\min_w \left(J(w) + C \sum_{i=1}^N \xi_i \right)$$

subject to

$$g_i(w; x_i, y_i) \leq \xi_i$$

slack variable

$$\xi_i \geq 0$$

where

$$J(w) = \frac{1}{2} w^T w$$

$$g_i(w; x_i, y_i) = 1 - f(x_i) y_i$$



$$L(\lambda; x_i, y_i) = J(w) + \sum_{i=1}^N \lambda_i (g(w; x_i, y_i) - \xi_i) \\ - \sum_{i=1}^N \mu_i \xi_i + C \sum_{i=1}^N \xi_i$$

$$(1) \quad \frac{\partial L}{\partial w} = w - \sum_{i=1}^N \lambda_i y_i x_i = 0$$

$$(2) \quad \frac{\partial L}{\partial w_0} = \sum_{i=1}^N \lambda_i y_i = 0$$

$$(3) \quad \lambda_i \frac{\partial L}{\partial \lambda_i} = 0, \quad \mu_i \frac{\partial L}{\partial \mu_i} = 0$$

$$(4) \quad \lambda_i \geq 0, \quad \mu_i \geq 0$$

■ small ξ_i

■ large ξ_i

C=10

슬랙변수 가중치 C의 영향

C=0.1

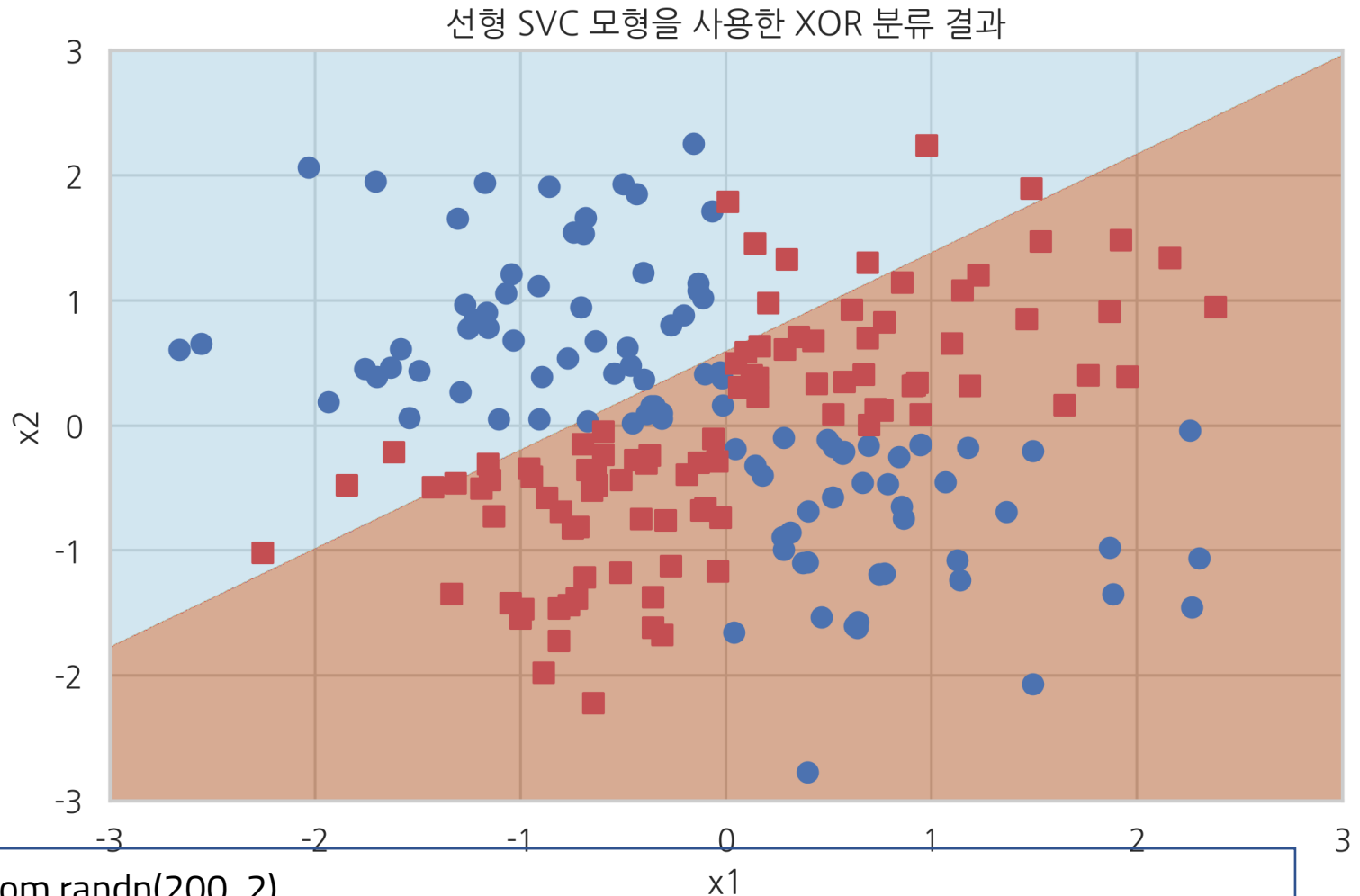
small margin

large margin

Exercise 2

Let's solve the iris problem with a support vector machine. Let's solve it by changing it to a binary classification problem **using only the species 'versicolor', 'virginica'**. Fix the kernel type as 'linear', **but find the optimal slack variable C while changing the value.**

Linear SVM cannot solve the XOR problem



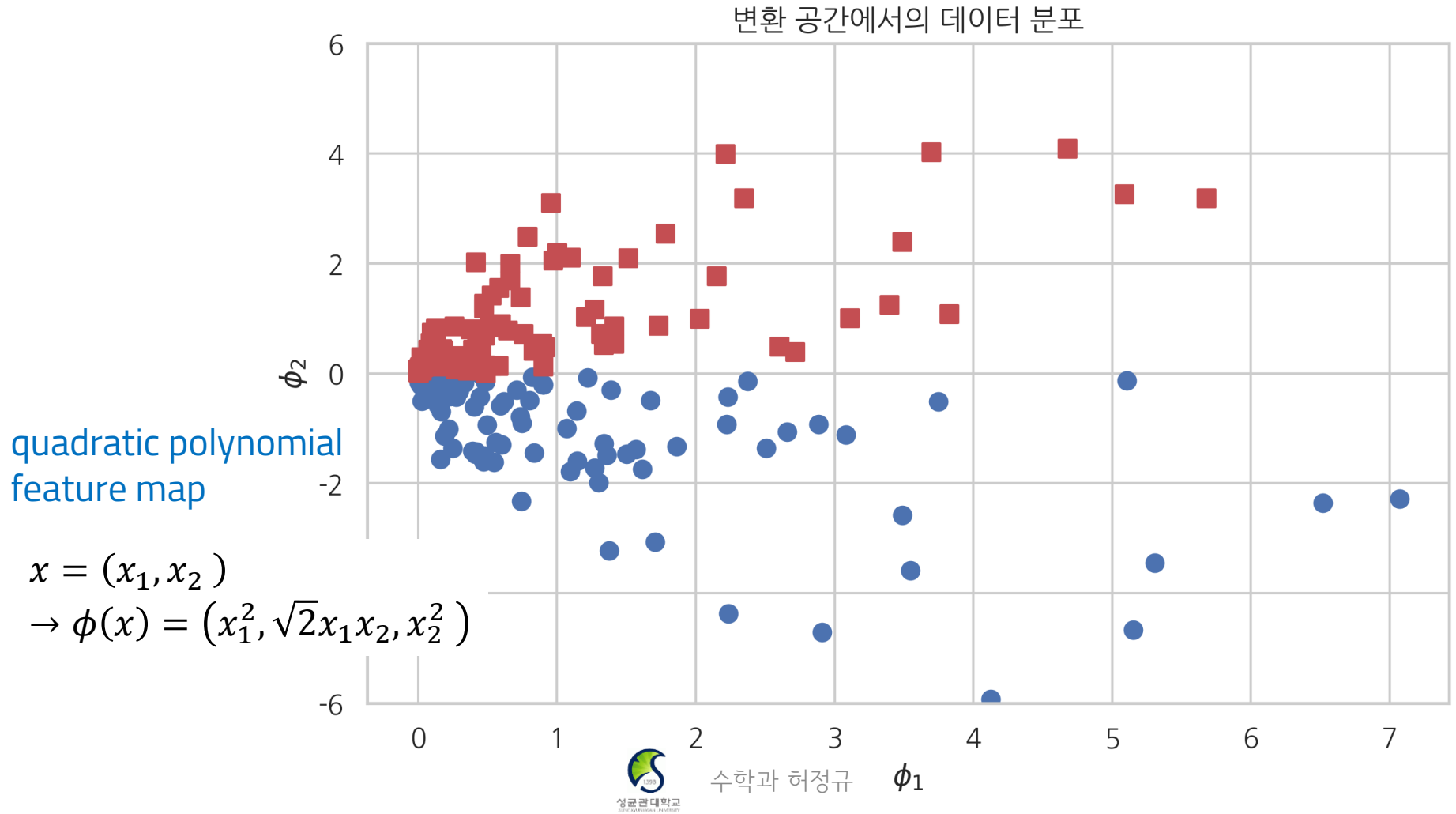
```
X_xor = np.random.randn(200, 2)
y_xor = np.logical_xor(X_xor[:, 0] > 0, X_xor[:, 1] > 0)
y_xor = np.where(y_xor, 1, 0)
```

Data Generation Code

Transform of data using a feature map

$\phi(\cdot): R^D \rightarrow R^M$ feature map

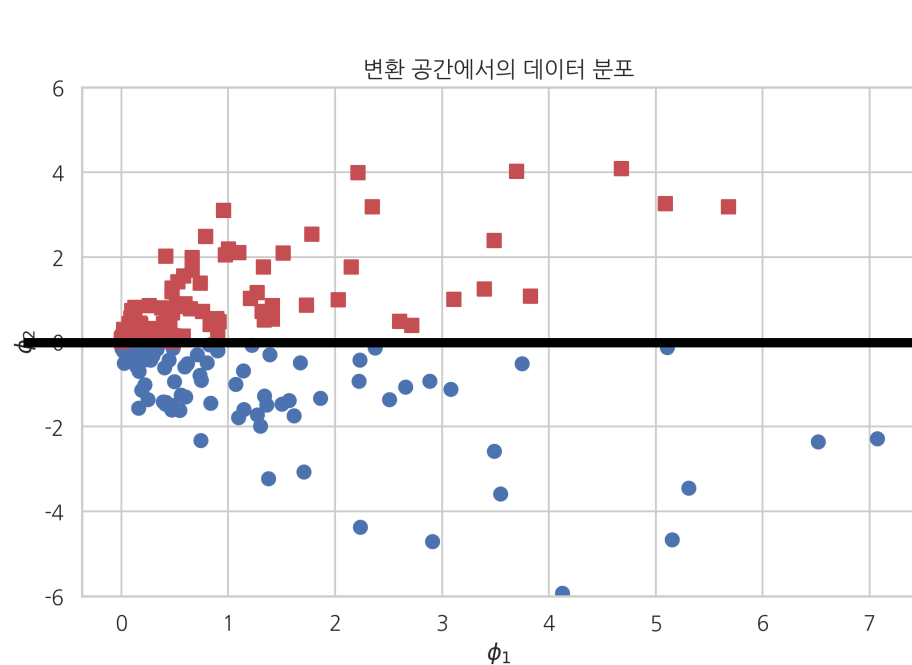
s.t. $x = (x_1, x_2, \dots, x_D) \rightarrow \phi(x) = (\phi_1(x), \phi_2(x), \dots, \phi_M(x))$



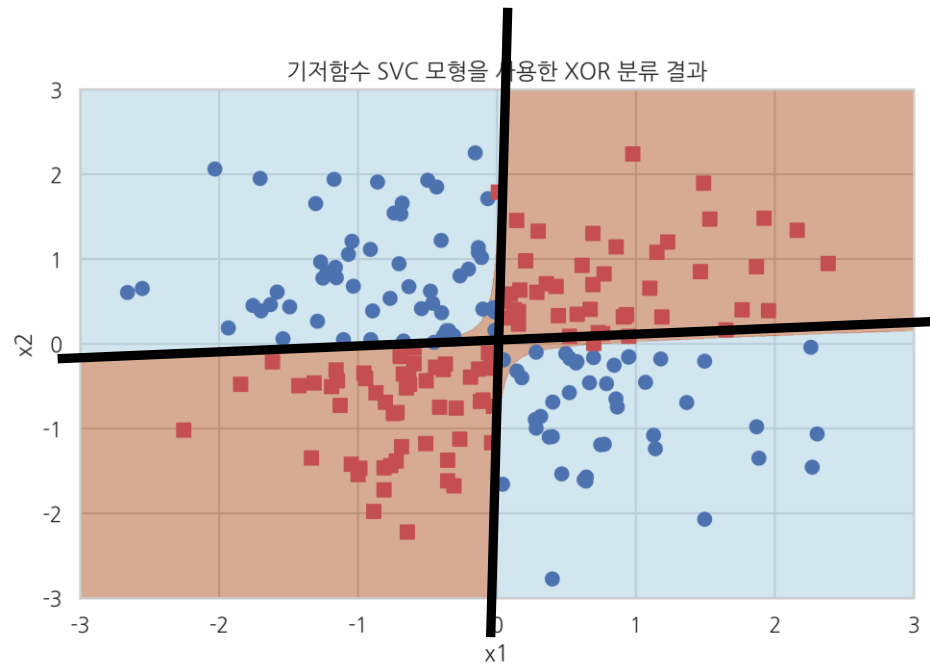
"Linear SVM + Quadratic polynomial kernel" solve the XOR problem

quadratic polynomial
feature map

$$x = (x_1, x_2)$$
$$\rightarrow \phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$



kernel feature space



original feature space

Support Vector Machine (SVM) with Scikit-Learn



```
from sklearn.preprocessing import FunctionTransformer

def kernel(X):
    return np.vstack([X[:, 0]**2, np.sqrt(2)*X[:, 0]*X[:, 1], X[:, 1]**2]).T

X_xor2 = FunctionTransformer(kernel).fit_transform(X_xor)
```

quadratic polynomial
feature map

$$x = (x_1, x_2)$$
$$\rightarrow \phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

```
from sklearn.pipeline import Pipeline

basismodel = Pipeline([("kernel", FunctionTransformer(kernel)),
    ("svc", SVC(kernel="linear"))]).fit(X_xor, y_xor)
```

pipeline construction
using the polynomial feature map and a support vector machine

Kernel Trick

* Model (SVM)

$$f(x) = w^T x - w_0$$

* Feature map

$$\phi(\cdot): R^D \rightarrow R^M \text{ s.t. } x \rightarrow \phi(x)$$

$$\min_w J(w)$$

subject to

$$g_i(w; x_i, y_i) \leq 0$$

where

$$J(w) = \frac{1}{2} w^T w$$

$$g_i(w; x_i, y_i) = 1 - f(\phi(x_i))y_i$$



$$\max_{\lambda} L(\lambda; \phi(x_i), y_i)$$

subject to

$$\sum_{i=1}^N \lambda_i y_i = 0, \quad \lambda_i \geq 0$$

where

$$L(\lambda; \phi(x_i), y_i) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j \phi(x_i)^T \phi(x_j)$$

kernel

$$k(u, v) = \phi(u)^T \phi(v)$$

Kernel Support Vector Machine

$$f_{\phi}(x) = w^T \phi(x) - w_0$$

$$w = \sum_{i=1}^N \lambda_i y_i \phi(x_i)$$



$$f_{\phi}(x) = w^T \phi(x) - w_0 = \lambda^+ (\phi(x^+))^T \phi(x) - \lambda^- (\phi(x^-))^T \phi(x) - w_0$$

$$\Rightarrow w = \lambda^+ \phi(x^+) - \lambda^- \phi(x^-)$$

$$f_{\phi}(x^+) = w^T \phi(x^+) - w_0 = +1$$

$$f_{\phi}(x^-) = w^T \phi(x^-) - w_0 = -1$$

$$\Rightarrow w_0 = \frac{1}{2} w^T (\phi(x^+) + \phi(x^-))$$

$$f_{\phi}(x) = \lambda^+ \langle x, x^+ \rangle_{\phi} - \lambda^- \langle x, x^- \rangle_{\phi} - w_0$$

$\langle \cdot, \cdot \rangle_{\phi}$: inner product kernel induced by ϕ
which is defined as $\langle u, v \rangle_{\phi} = \phi(u)^T \phi(v)$

similarity
measure

Commonly Used Kernels

- Linear Kernel

$$k(u, v) = u^T v$$

components ϕ_j of
the feature map
 $\phi = (\phi_1, \dots, \phi_M)$

- Polynomial Kernel

$$k(u, v) = (\gamma(u^T v) + \theta)^d$$

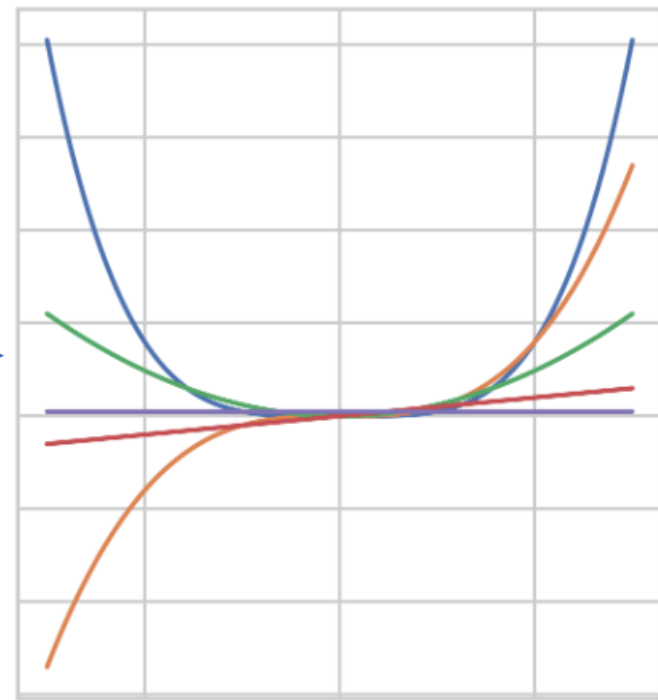
$$k(u, v) = \phi(u)^T \phi(v)$$

- Gaussian Kernel (or called Radial Basis Function)

$$k(u, v) = \exp(-\gamma \|u - v\|^2)$$

- Sigmoid Kernel

$$k(u, v) = \tanh((\gamma(u^T v) + \theta))$$



Kernel Support Vector Machine



* polynomial kernel

```
polysvc = SVC(kernel="poly", degree=2, gamma=1, coef0=0).fit(X_xor, y_xor)
```

* Gaussian kernel

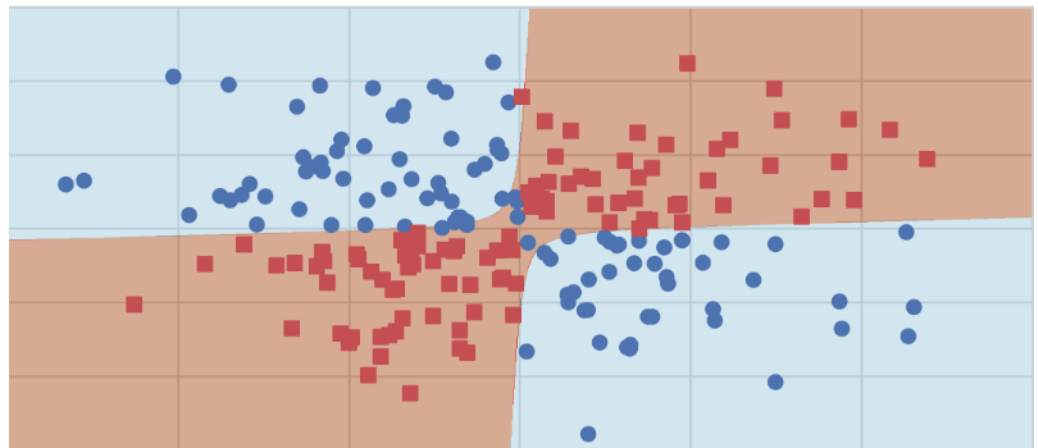
```
rbfsvc = SVC(kernel="rbf").fit(X_xor, y_xor)
```

* sigmoid kernel

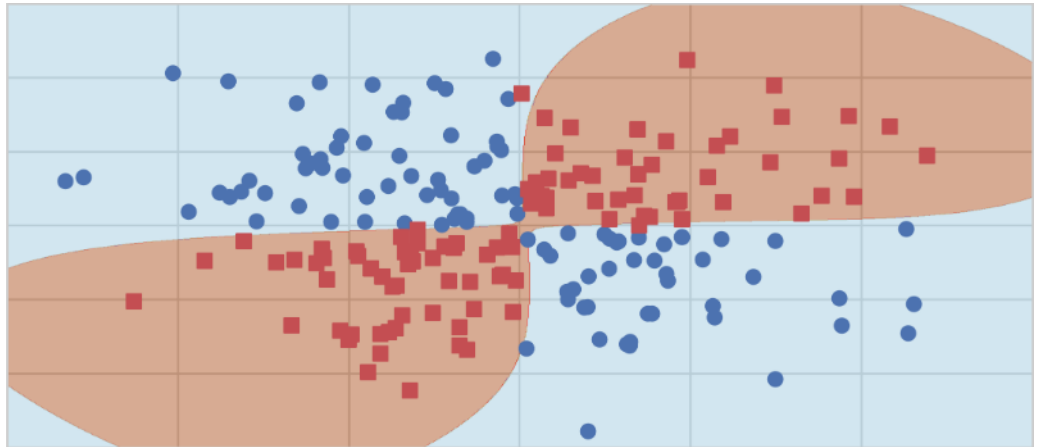
```
sigmoidsvc = SVC(kernel="sigmoid", gamma=2, coef0=2).fit(X_xor, y_xor)
```

- `kernel = "linear"`: 선형 SVM. $k(x_1, x_2) = x_1^T x_2$
- `kernel = "poly"`: 다항 커널. $k(x_1, x_2) = (\gamma(x_1^T x_2) + \theta)^d$
 - `gamma`: γ ◦ `coef0`: θ ◦ `degree`: d
- `kernel = "rbf"` 또는 `kernel = None`: RBF 커널. $k(x_1, x_2) = \exp(-\gamma \|x_1 - x_2\|^2)$
 - `gamma`: γ
- `kernel = "sigmoid"`: 시그모이드 커널. $k(x_1, x_2) = \tanh(\gamma(x_1^T x_2) + \theta)$
 - `gamma`: γ ◦ `coef0`: θ

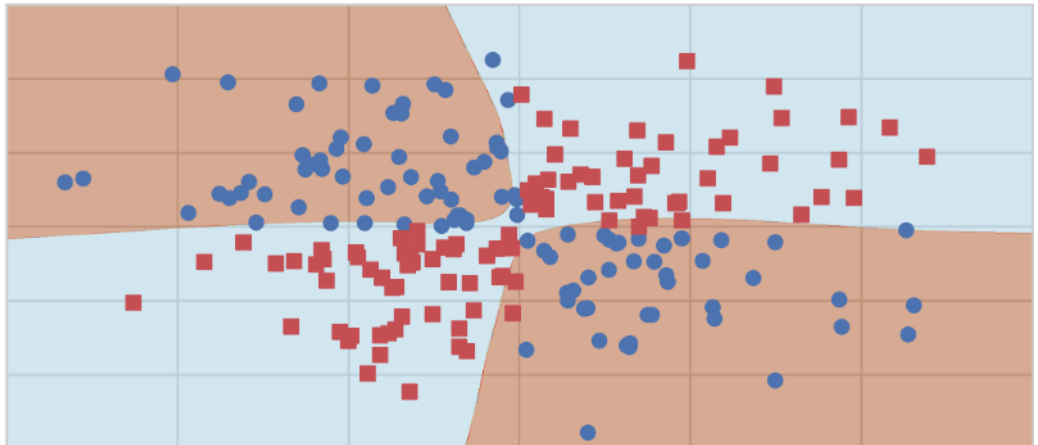
Polynomial
kernel



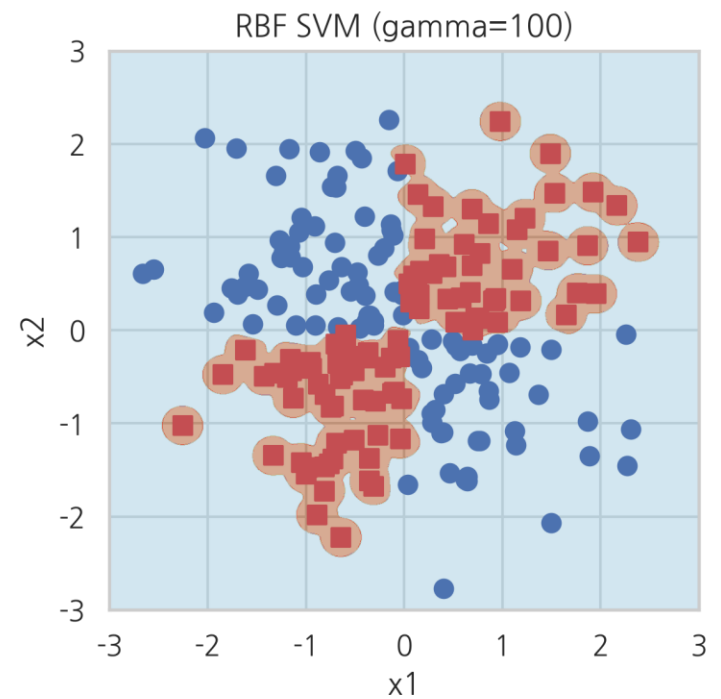
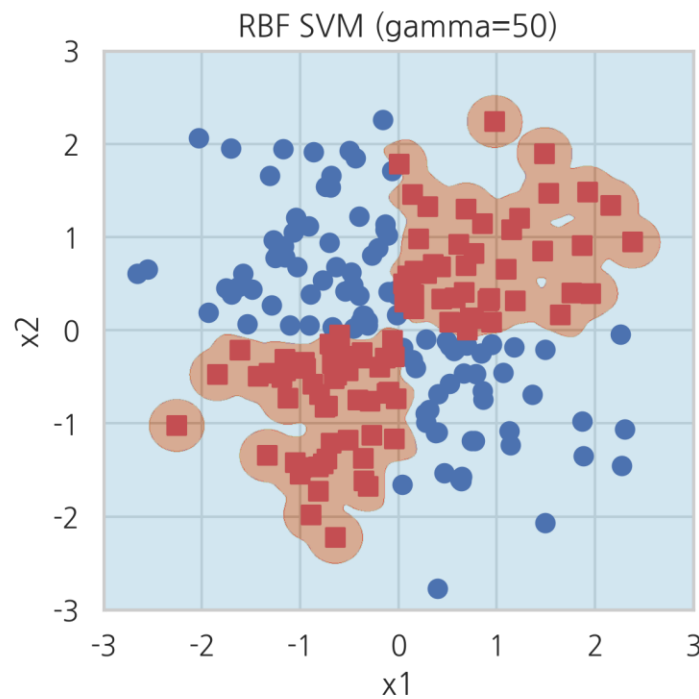
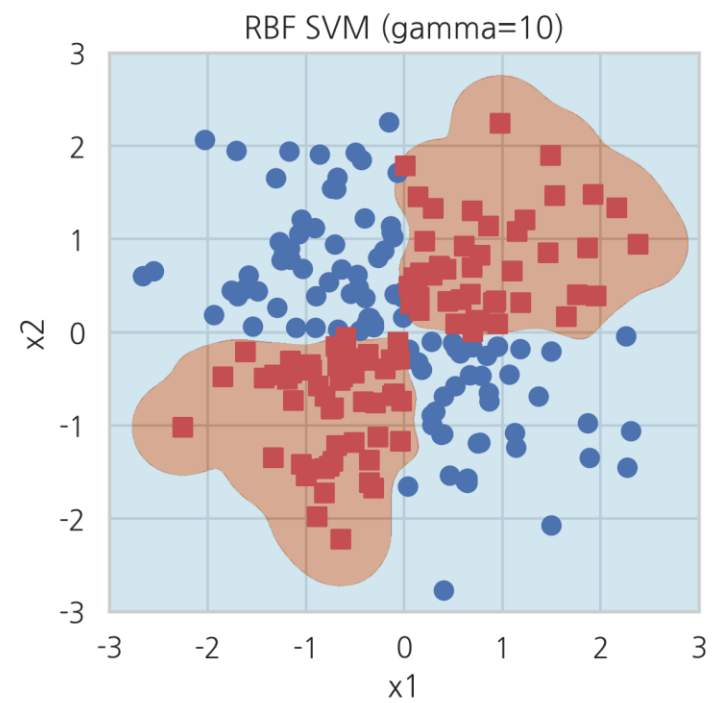
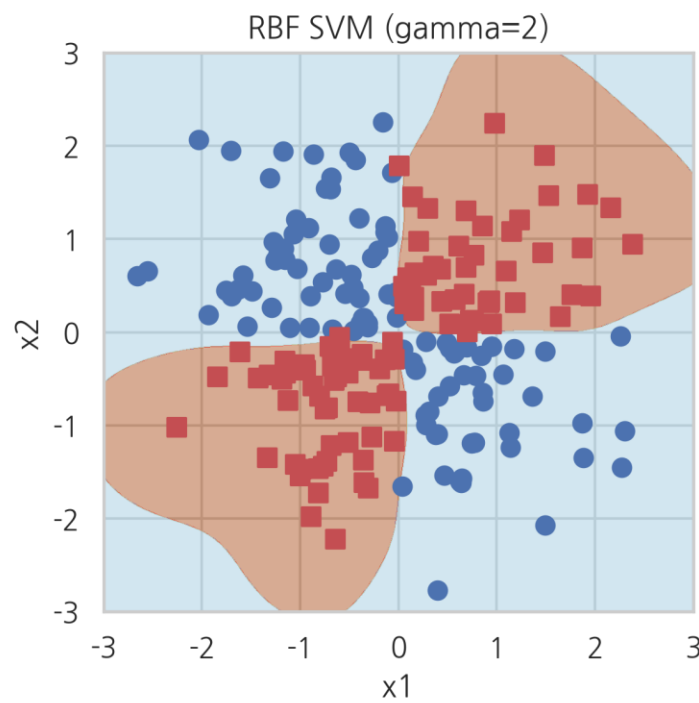
Gaussian
kernel



Sigmoid
kernel



Kernel Parameter Effect



Exercise 3

Let's solve the iris problem with a support vector machine. Let's solve it by changing it to a binary classification problem using only the species 'versicolor', 'virginica'. Find the optimal kernel type and slack variable C while changing the values.