

# CUAI 스터디\_MTs

2022.05.03

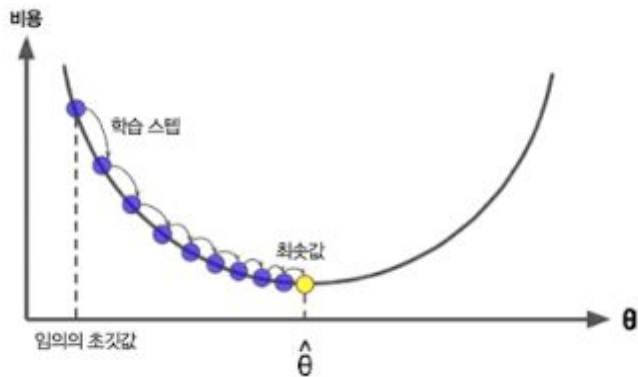
발표자 : 정 승 욱

## 스터디 모임 인증



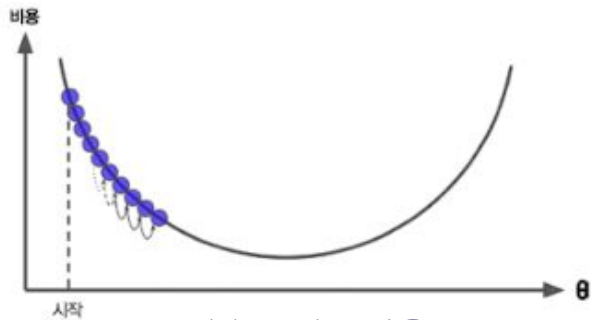
## Ch04. 모델 훈련 Regression

## 경사 하강법

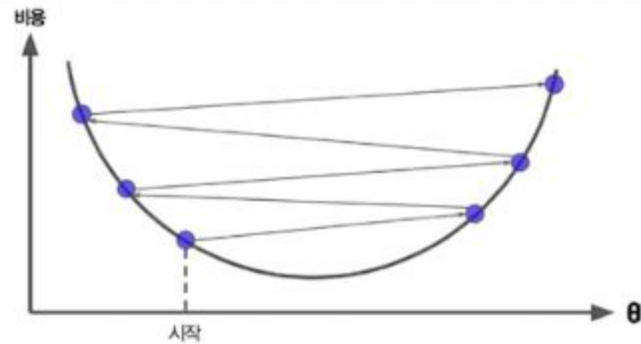


$$\theta^{(nextstep)} = \theta - \eta \nabla_{\theta} MSE(\theta)$$

$\eta = \text{Learning Rate}$



학습률 너무 작을  
때  
:



학습률 너무 클 때

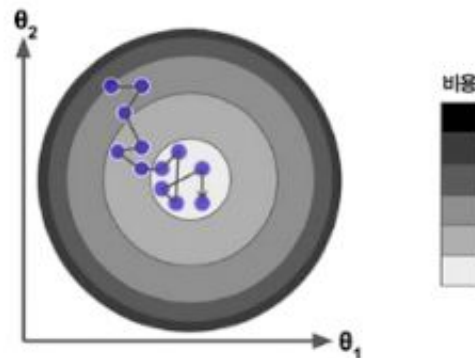
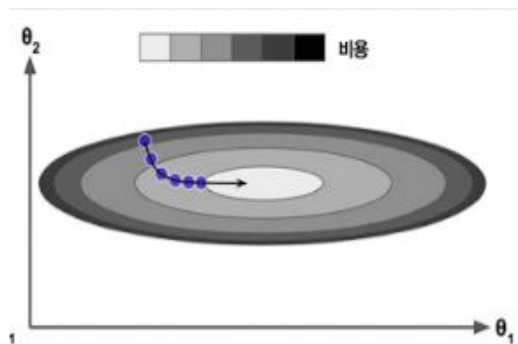
# THOHI

## 배치 경사 하강법 & 확률적 경사 하강법 & 미니 배치 경사 하강법

배치 : 매 스텝 **모든** 샘플

확률 : 매 스텝 **1개** 샘플

미니 배치 : 매 스텝 **작은 랜덤** 샘플



<https://www.youtube.com/watch?v=sDv4f4s2SB8>

<https://www.youtube.com/watch?v=vMh0zPT0tLI>

## Epoch & Batch & Iteration



1 Epoch : 모든 데이터 셋을 한 번 학습

1 iteration : 1회 학습

minibatch : 데이터 셋을 batch size 크기로 쪼개서 학습

ex) 총 데이터가 100개, batch size가 10이면,  
1 iteration = 10개 데이터에 대해서 학습  
1 Epoch =  $100 / \text{batch size} = 10$  iteration

# 분산과 편향

편향: 잘못된 가정  
분산: 작은 변동에 모델이 민감

데이터 안의 진동: 엡실론으로 표기!

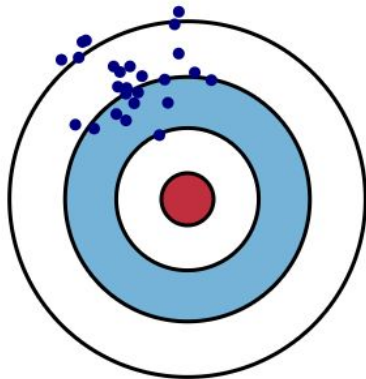
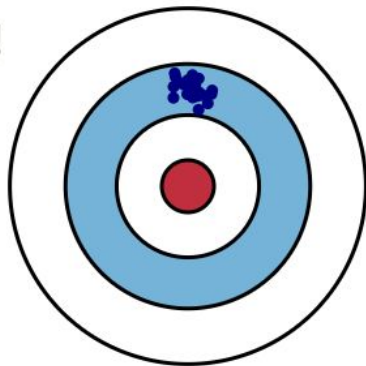
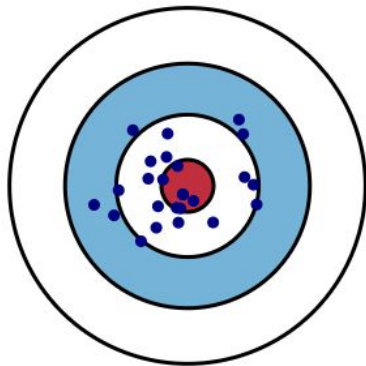
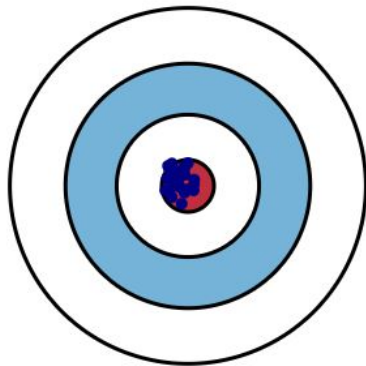
편향 종류!! bias vs variance & ' $\epsilon$ '인지!

Low Bias

High Bias

Low Variance

High Variance



## 규제 : 릿지, 라쏘, 엘라스틱넷

-> 가중치에 제한, 과대적합 방지

$$J(\theta) = \text{MSE}(\theta) + \alpha \frac{1}{2} \sum_{i=1}^n \theta_i^2$$

릿지: 덜 중요한 특성의 가중치 낮춤 **l2 규제!**

$$J(\theta) = \text{MSE}(\theta) + \alpha \sum_{i=1}^n |\theta_i|$$

라쏘: 덜 중요한 특성의 가중치 삭제 **l1 규제!**

$$J(\theta) = \text{MSE}(\theta) + r\alpha \sum_{i=1}^n |\theta_i| + \frac{1-r}{2}\alpha \sum_{i=1}^n \theta_i^2$$

릿지와 라쏘의 절충: 엘라스틱 넷

혼합 비율  $r, 1-r$

- 특성 수가 훈련 샘플 수 보다 많거나
- 특성 몇 개가 강하게 연관되어 있을 때 라쏘가 문제를 일으킴



# 로지스틱 회귀: 범주형 자료 회귀, 분류 모델

이진 분류 하는 문제! 1

또는 0

시그모이드 함수를 이용

$$\log(\text{Odds}(p)) = wx + b$$

오즈 Odds

실패 비율 대비 성공 비율

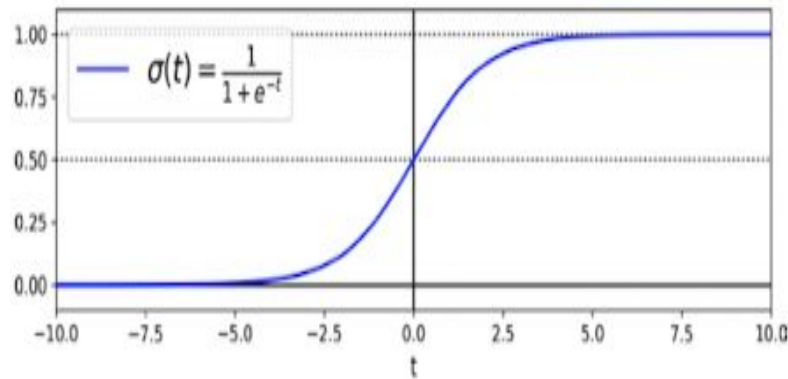
$$\frac{p}{1-p}$$

로짓 Logit Log+Odds

오즈에 자연로그!

$$L = \ln \frac{p}{1-p}$$

$$p = \frac{e^{-L}}{e^{-L} + 1}$$



<https://www.youtube.com/watch?v=yIYKR4sgzI8>

## Ch05. SVM

## SVM(Support Vector Machine)

Decision Boundary(결정경계)를 결정하기 위한  
이진분류모델

=> Margin이 최대화 된 초평면(hyperplane)을 찾는 모델

# Margin??

결정경계와 서포트벡터 까지의 거리

# SVM(Support Vector Machine)

```
from sklearn import datasets  
  
iris = datasets.load_iris()  
...
```

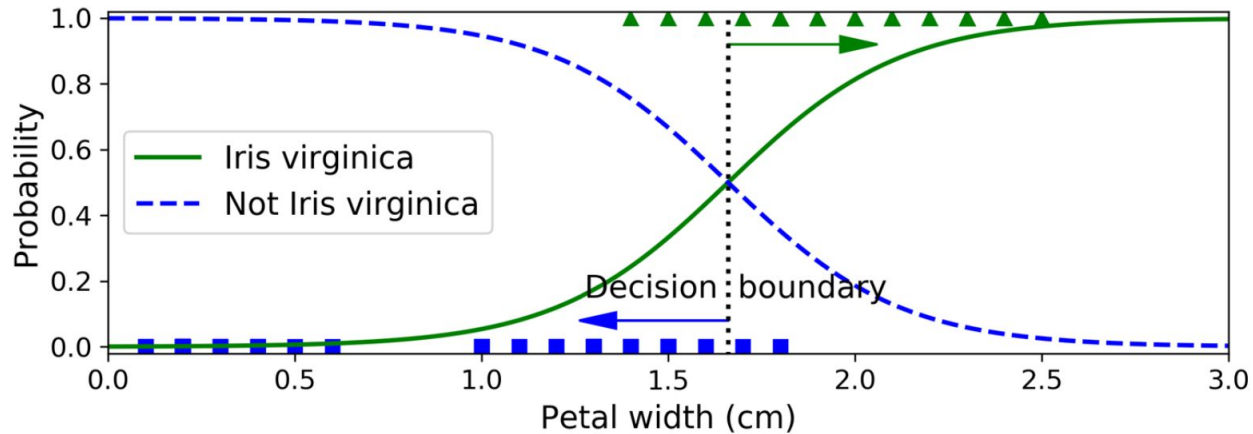
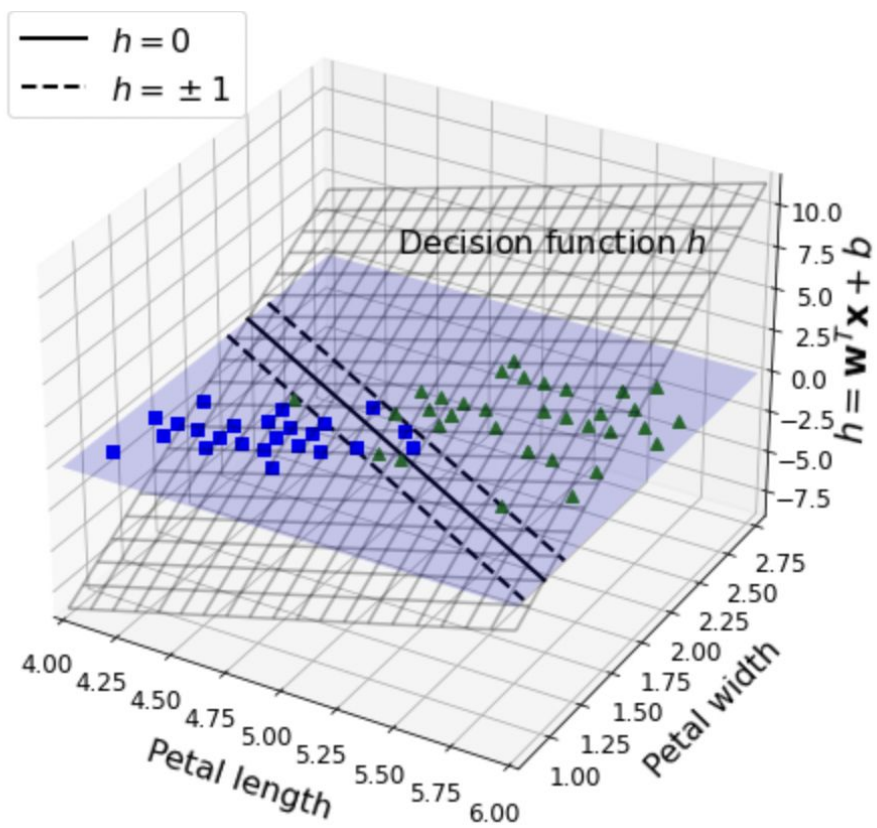


Figure 4-23. Estimated probabilities and decision boundary



# SVM(Support Vector Machine)

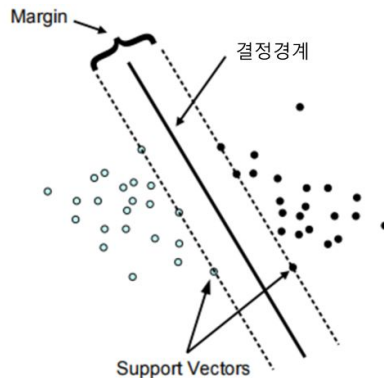
## 하드 마진 분류, 소프트 마진 분류

### Hard margin Classification

- linear separable 할 때, 잘 작동한다
- support vector를 잘 골라내야하기 때문에 이상치에 민감하다.
- margin 값을 작게 잡아, 개별적인 데이터에 대해 신경쓴다.  $\Rightarrow$  overfitting의 문제가 생긴다.

### Soft margin Classification

- margin 값을 크게 잡아, decision boundary와 서포트벡터들이 멀어진다.  $\Rightarrow$  under fitting의 문제



C(cost)

# SVM(Support Vector Machine)

```
from sklearn.datasets import make_moons
X, y = make_moons(n_samples=100, noise=0.15, random_state=42)

def plot_dataset(X, y, axes):
    plt.plot(X[:, 0][y==0], X[:, 1][y==0], "bs")
    plt.plot(X[:, 0][y==1], X[:, 1][y==1], "g^")
    plt.axis(axes)
    plt.grid(True, which='both')
    plt.xlabel(r"$x_1$", fontsize=20)
    plt.ylabel(r"$x_2$", fontsize=20, rotation=0)

plot_dataset(X, y, [-1.5, 2.5, -1, 1.5])
plt.show()
```

## SVM(Support Vector Machine) - 다항식 커널

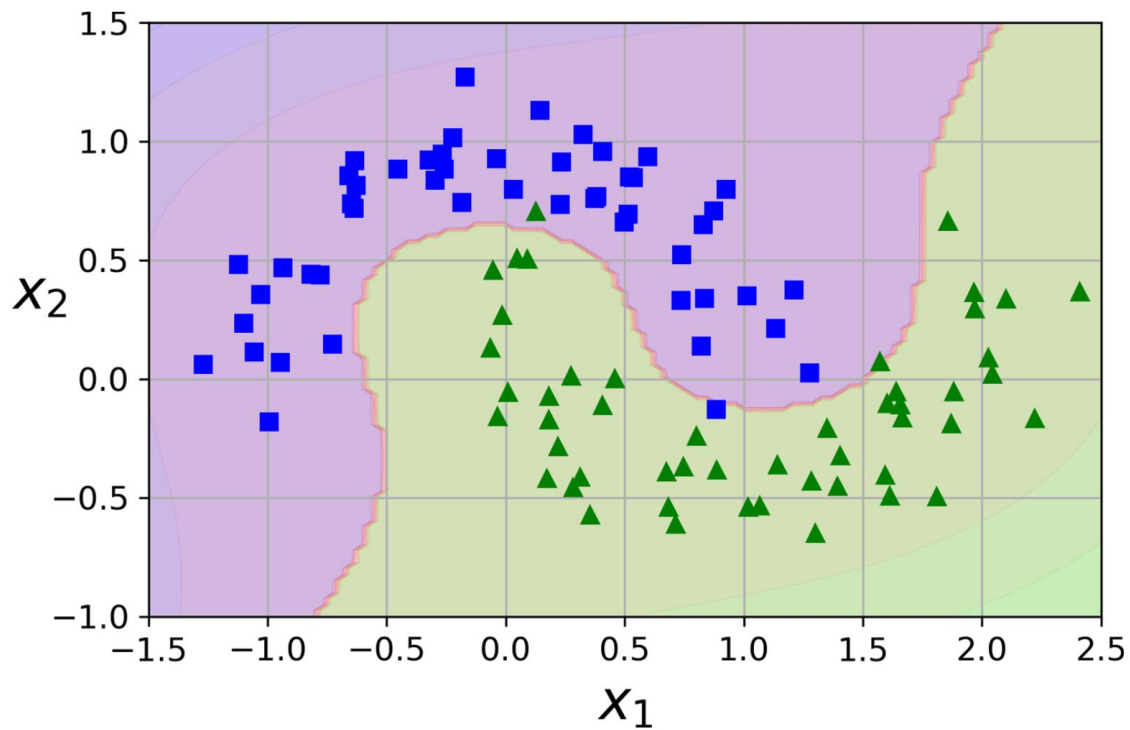
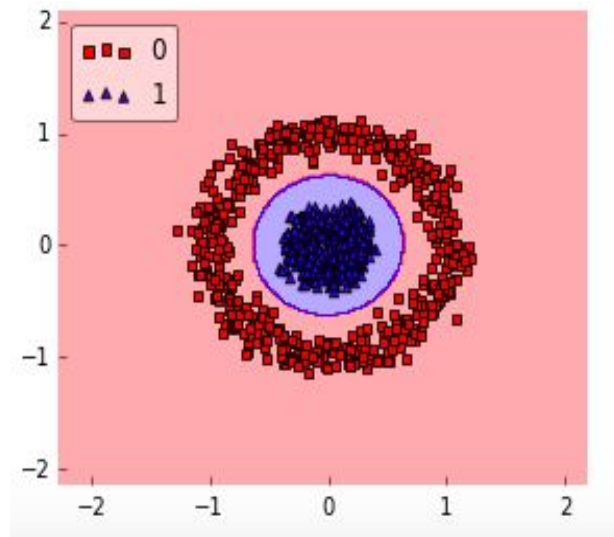
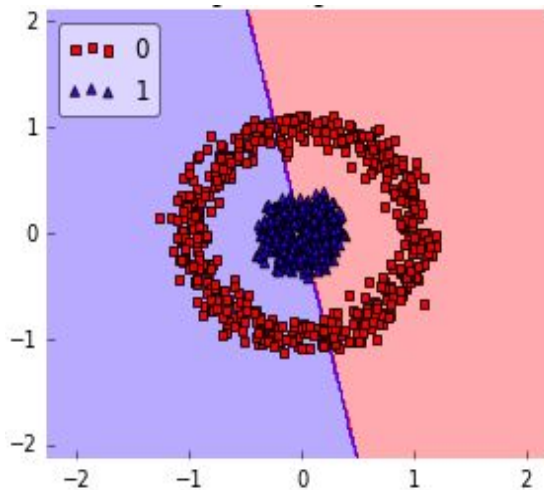


Figure 5-6. Linear SVM classifier using polynomial features



## SVM(Support Vector Machine)

### RBF 커널(가우시안 커널)



## SVM(Support Vector Machine)

### RBF 커널(가우시안 커널)

#### Kernel

- SVM에서 목적함수와 예측 모델을 dual form으로 변형가능하다.
  - 두개의 변환된 독립변수 벡터를 inner product한 값을 하나의 함수로 나타낼 수 있다.

$$k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

저차원 => 고차원

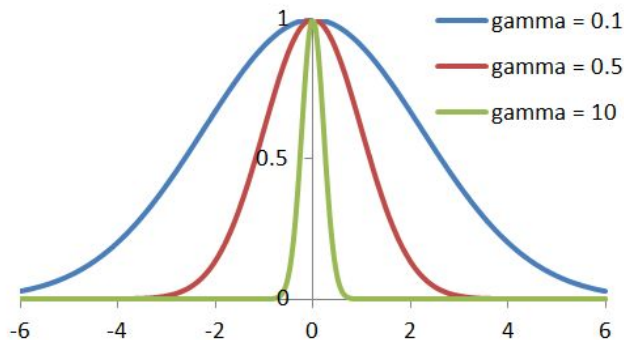
커널이란? 저차원에서 분리되지 않는 데이터들을 추가적인 연산을 통해서 고차원에서 분리

## SVM(Support Vector Machine)

### RBF 커널(가우시안 커널)

## Gamma??

영향력을 행사하는 거리  
 -> 규제의 역할  
 (가우시안 함수에서 표준편차와 관련)



Gamma r값에 따른  
정규분포 그래프

Gamma가 클수록 규제가 심하다.

- ✓ 각 데이터 샘플의 영향력이 높아진다.
- ✓ 결정경계의 곡률을 결정한다.
- ✓ 클수록 Overfitting

# SVM(Support Vector Machine) 사용하기

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

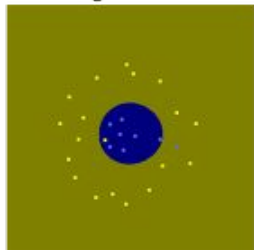
```
from sklearn.svm import SVC

gamma1, gamma2 = 0.1, 5
C1, C2 = 0.001, 1000
hyperparams = (gamma1, C1), (gamma1, C2), (gamma2, C1), (gamma2, C2)

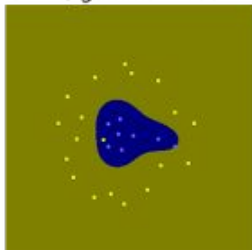
svm_clfs = []
for gamma, C in hyperparams:
    rbf_kernel_svm_clf = Pipeline([
        ("scaler", StandardScaler()),
        ("svm_clf", SVC(kernel="rbf", gamma=gamma, C=C))
    ])
    rbf_kernel_svm_clf.fit(X, y)
    svm_clfs.append(rbf_kernel_svm_clf)
```

gamma 커짐

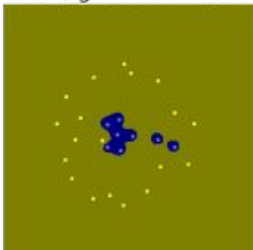
C = 1, gamma = 10



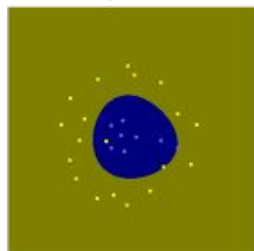
C = 1, gamma = 100



C = 1, gamma = 1000



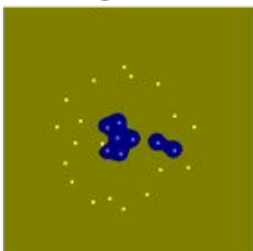
C = 10, gamma = 10



C = 10, gamma = 100



C = 10, gamma = 1000



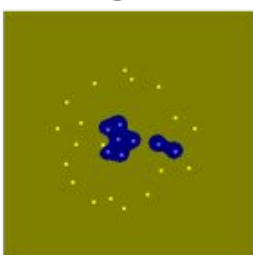
C = 100, gamma = 10



C = 100, gamma = 100



C = 100, gamma = 1000



C 커짐

THOHI

감사합니다!