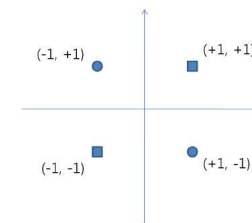


# Support Vector Machine

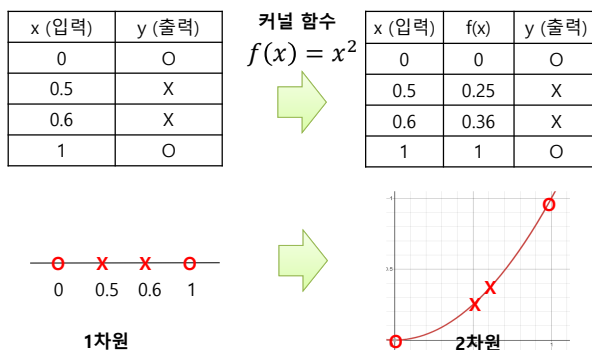
# SVM (Support Vector Machine)

- SVM
  - 데이터들을 커널함수(kernel function)를 이용하여 고차원 공간으로 사상시킨 후 support vector들로 이루어진 초평면을 이용하여 선형 분류하는 마진 기반 기계학습 모델
- 예제: XOR
  - 2차원 XOR 좌표를 이등분하는 초평면을 찾는 문제



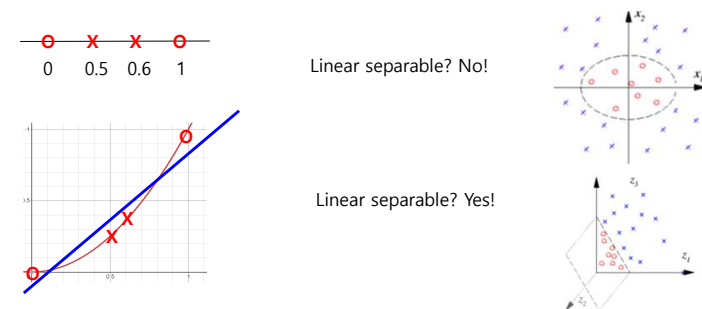
# 커널 함수 (Kernel Function)

- 커널 함수
  - 저차원 데이터를 고차원 공간으로 사상(mapping)시키는 함수
  - 예제



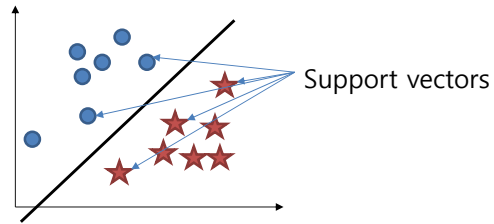
# 커널 함수 (Kernel Function)

- 커널 함수의 기능
  - 선형 분리 불가능 문제를 선형 분리 가능 문제로 변환
  - 선형 분리 가능 문제 (linear separable problem)
    - 한 개의 직선(초평면)으로 분리할 수 있는 문제



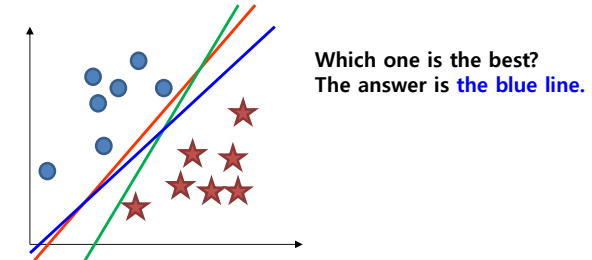
## 지지 벡터 (Support Vectors)

- SVM
  - 데이터들을 커널함수(Kernel function)를 이용하여 고차원 공간으로 사상시킨 후 **support vector들로 이루어진** 초평면을 이용하여 선형 분류하는 **마진 기반 기계학습 모델**
- 지지 벡터
  - 선형 분류(linear classification)의 경계 주변에 존재하는 데이터 포인트들
  - 이진 선형 분류 문제에서는 수많은 support vector들이 존재



## 최대 마진 (Maximum Margin)

- SVM
  - 데이터들을 커널함수(Kernel function)를 이용하여 고차원 공간으로 사상시킨 후 support vector들로 이루어진 초평면을 이용하여 선형 분류하는 **마진 기반 기계학습 모델**
- 최대 마진
  - 선형 분류를 가능하게 하는 직선은 무수히 많이 존재
  - 그 중에서 마진을 최대로 하는 직선이 가장 이상적 임



## 최대 마진 (Maximum Margin)

(1) 학습 데이터

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}^+ + b &= +1 \\ \mathbf{w} \cdot \mathbf{x}^- + b &= -1 \\ \mathbf{x}^+ &= \mathbf{x}^- + \lambda \mathbf{w} \\ |\mathbf{x}^+ - \mathbf{x}^-| &= M \end{aligned}$$

(2) 수식 유도

$$\begin{aligned} \mathbf{w}(\mathbf{x}^- + \lambda \mathbf{w}) + b &= 1 \quad M = |\mathbf{x}^+ - \mathbf{x}^-| = |\lambda \mathbf{w}| = \lambda |\mathbf{w}| \\ \mathbf{w} \cdot \mathbf{x}^- + b + \lambda \mathbf{w} \cdot \mathbf{w} &= 1 \\ -1 + \lambda \mathbf{w} \cdot \mathbf{w} &= 1 \\ \lambda &= \frac{2}{\mathbf{w} \cdot \mathbf{w}} \\ M &= \lambda |\mathbf{w}| = \frac{2}{\sqrt{\mathbf{w} \cdot \mathbf{w}}} = \frac{2}{\|\mathbf{w}\|} \end{aligned}$$

(3) 목적 함수

M을 최대화하는 것  $\rightarrow \|\mathbf{w}\|$ 를 최소화하는 것!

## 최적화 문제 (Optimization Problem)

- Primal Problem (주어진 원래 문제)

$$F(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b \quad \& \quad \begin{aligned} \mathbf{w} \cdot \mathbf{x}_i - b &> 0 \text{ if } y_i = 1, \text{ and} \\ \mathbf{w} \cdot \mathbf{x}_i - b &< 0 \text{ if } y_i = -1 \end{aligned}$$

$$\begin{aligned} \text{minimize: } Q(\mathbf{w}) &= \frac{1}{2} \|\mathbf{w}\|^2 &< \text{목적함수: Convex function of } \mathbf{w} \\ \text{subject to: } y_i(\mathbf{w} \cdot \mathbf{x}_i - b) &= 1, \forall (\mathbf{x}_i, y_i) \in D &< \text{제약조건: Linear in } \mathbf{w} \end{aligned}$$

$\mathbf{w}$ 와  $b$ 에 대한 최소값 = 정점  
 $\rightarrow$  미분을 했을 때 기울기가 0인 곳!

★ 주어진 제약하에서 목적함수의 최소값을 구하는 문제? **Lagrange function**

$$\text{Condition 1: } \frac{\partial J(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = 0$$

$$\text{Condition 2: } \frac{\partial J(\mathbf{w}, b, \alpha)}{\partial b} = 0$$

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^m \alpha_i \{y_i(\mathbf{w} \cdot \mathbf{x}_i - b) - 1\}$$

$\alpha_i$  Lagrange multiplier

$$\mathbf{w} = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i \quad \& \quad \sum_{i=1}^m \alpha_i y_i = 0$$

## 최적화 문제 (Optimization Problem)

### Dual Problem (쌍대 문제)

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^m \alpha_i \{y_i (\mathbf{w} \cdot \mathbf{x}_i - b) - 1\} \quad \& \quad \mathbf{w} = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i \quad \& \quad \sum_{i=1}^m \alpha_i y_i = 0$$

$$J(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} - \sum_{i=1}^m \alpha_i y_i \mathbf{w} \cdot \mathbf{x}_i + b \sum_{i=1}^m \alpha_i y_i - \sum_{i=1}^m \alpha_i$$

$$\mathbf{w} \cdot \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i = \mathbf{w} \cdot \mathbf{w}$$

Let  $J(\mathbf{w}, b, \alpha) = Q(\alpha)$

$$Q(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

★  $\mathbf{w}$ 와  $b$ 에 대한 최소값 =  $\alpha$ 에 대한 최대값!

$$\begin{aligned} \text{maximize: } Q(\alpha) &= \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \\ \text{subject to: } &\sum_{i=1}^m \alpha_i y_i = 0 \\ &\alpha_i \geq 0 \end{aligned}$$

$\mathbf{w}$ 가 없어지고 training data에서 얻을 수 있는 것으로 모두 바뀌었다!

## 최적화 문제 (Optimization Problem)

maximize:  $Q(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$  ★ Lagrange multiplier → 편미분 ?

subject to:  $\sum_{i=1}^m \alpha_i y_i = 0$   
 $\alpha_i \geq 0$

제약사항이 부등식이면? 쿤-터커 정리 적용

번호	조건식	풀이
1	$\frac{\partial L}{\partial x_i} = f_i - \sum_{k=1}^m \lambda_k g_i^k \leq 0$	라그랑지안 함수를 각 변수 $x_i$ 로 편미분한 결과는 극값에서 음이거나 0이어야 한다.
2	$\frac{\partial L}{\partial \lambda_i} = r_i - g^k \geq 0$	라그랑지안 함수를 라그랑지 승수로 편미분한 결과는 비음이어야 한다.
3	$x_i (\frac{\partial L}{\partial x_i}) = x_i (f_i - \sum_{k=1}^m \lambda_k g_i^k) = 0$	최적해에서 변수 $x_i$ 와 $\frac{\partial L}{\partial x_i}$ 를 곱한 값이 0이어야 한다.
4	$\lambda_i (\frac{\partial L}{\partial \lambda_i}) = \lambda_i (r_i - g^k) = 0$	최적해에서 변수 $\lambda_i$ 와 $\frac{\partial L}{\partial \lambda_i}$ 를 곱한 값이 0이어야 한다.
5	$x_i \geq 0$	최적해에서 변수는 비음이어야 한다.
6	$\lambda_i \geq 0$	최적해에서 라그랑지안 승수는 비음이어야 한다.

Finally  $F(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} - b \rightarrow F(\mathbf{x}) = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} - b$  즉,  $Q(\mathbf{w}) = Q(\alpha)$

## 커널 트릭 (Kernel Trick)

### 커널 트릭

- SVM 함수는 고차원 공간에서의 내적을 통해서 구현

$$F(\mathbf{x}) = \sum_{i=1}^m \alpha_i y_i \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}) - b$$

고차원 공간으로 사상한 후 내적 수행

- 고차원 공간으로 사상하고 내적을 구하는 것은 매우 복잡
- 현재 차원에서 동일한 효과를 거두는 커널 함수를 사용 → 트릭

Mercer의 이론에 따르면 다음 조건을 만족하면 대체 가능

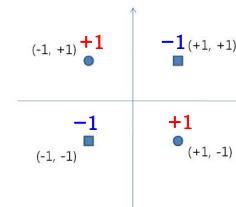
$$\int K(\mathbf{u}, \mathbf{v}) \psi(\mathbf{u}) \psi(\mathbf{v}) d\mathbf{x} d\mathbf{y} \leq 0$$

where  $\int \psi(\mathbf{x})^2 d\mathbf{x} \leq 0$

- Polynomial:  $K(a, b) = (a \cdot b + 1)^d$
- Radial Basis Function (RBF):  $K(a, b) = \exp(-\gamma \|a - b\|^2)$
- Sigmoid:  $K(a, b) = \tanh(\kappa a \cdot b + c)$

잘 알려진 커널 트릭 함수들!

## 확인 예제



XOR problem  
(선형 분리 불가능)

$x_1$	$x_2$	$y$
-1	-1	-1
-1	+1	+1
+1	-1	+1
+1	+1	-1

커널 트릭 함수

$$K(\mathbf{x}, \mathbf{x}_i) = (1 + \mathbf{x} \cdot \mathbf{x}_i)^2$$

$$K(\mathbf{x}, \mathbf{x}_i) = 1 + x_1^2 x_{i1}^2 + 2x_1 x_2 x_{i1} x_{i2} + x_2^2 x_{i2}^2 + 2x_1 x_{i1} + 2x_2 x_{i2}$$

고차원 사상 효과

$$\varphi(\mathbf{x}) = (1, x_1^2, \sqrt{2}x_1 x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2)$$

## 확인 예제

$$\text{maximize: } Q_2(\alpha) = \sum_i \alpha_i - \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to: } \sum_i \alpha_i y_i = 0 \\ C \geq \alpha \geq 0$$

$$Q(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 \\ - \frac{1}{2} (9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4 \\ + 9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4 + \alpha_4^2)$$

Lagrange multipliers (각  $\alpha$ 의 미분=0)

$$9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 = 1 \\ -\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 = 1 \\ -\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 = 1 \\ \alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 = 1$$

$$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{1}{8}$$

$$Q(\alpha) = \frac{1}{4} \\ \frac{1}{2} \|w\|^2 = \frac{1}{4}, \text{ or } \|w\| = \frac{1}{\sqrt{2}}$$

## 확인 예제

$$w = \sum \alpha_i y_i \phi(\mathbf{x}_i) \& \phi(\mathbf{x}) = (1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2) \& \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{1}{8}$$

$$w = \frac{1}{8} [-\phi(\mathbf{x}_1) + \phi(\mathbf{x}_2) + \phi(\mathbf{x}_3) - \phi(\mathbf{x}_4)] \\ = \frac{1}{8} \left[ \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ -\sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ \sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ \sqrt{2} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ \sqrt{2} \end{bmatrix} \right] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{matrix} (-1, +1), (+1, -1) \\ \bullet +1 \\ -x_1x_2 = 0 \\ \blacksquare -1 \\ (-1, -1), (+1, +1) \end{matrix}$$

SVM 함수

$$w \cdot \phi(\mathbf{x}) - b = 0 \Rightarrow w \cdot \phi(\mathbf{x}) = [0 \ 0 \ \frac{-1}{\sqrt{2}} \ 0 \ 0 \ 0] \begin{bmatrix} 1 \\ x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix} = 0 \Rightarrow -x_1x_2 = 0$$

## SVM에 적합한 문제

- 선형 분류 문제: DT = NN = SVM
- 비선형 분류 문제: NN >= SVM
- 비선형 분류 문제 → 고차원에서의 선형 분류 문제: SVM >>= DT, NN (속도 & 성능)

## 실습

실습 코드 다운로드:  
<https://github.com/KUNLP/Lecture>

- SMSSpamCollection 데이터를 입력으로 하는 SVM 기반 스팸 메일 필터링 프로그램을 작성하시오.
  - SMSSpamCollection 데이터 형식
    - ham/spam Wt 문장

```
ham Go until jurong point, crazy.. Available only in bugis
ham Ok lar... Joking wif u oni...
spam Free entry in 2 a wkly comp to win FA Cup final tkt
ham U dun say so early hor... U c already then say...
ham Nah I don't think he goes to usf, he lives around here
spam FreeMsg Hey there darling it's been 3 week's now ar
ham Even my brother is not like to speak with me. They tree
ham As per your request 'Melle Melle' (Oru Minnaminunginte N
spam WINNER!! As a valued network customer you have been
spam Had your mobile 11 months or more? U R entitled to
ham I'm gonna be home soon and i don't want to talk about t
spam SIX chances to win CASH! From 100 to 20,000 pounds
```

## 실습

### 구글 colab 연결

```
from google.colab import drive
drive.mount("/gdrive", force_remount=True)
```

### 데이터 읽기 (처음부터 100개)

```
import numpy as np

file_path = "/gdrive/My Drive/colab/svm/SMSSpamCollection"

# 파일 읽기
x_data, y_data = [], []
with open(file_path, 'r', encoding='utf8') as inFile:
    lines = inFile.readlines()

lines = lines[:100]

for line in lines:
    line = line.strip().split(' ')
    sentence, label = line[1], line[0]
    x_data.append(sentence)
    y_data.append(label)

print("x_data의 개수 : " + str(len(x_data)))
print("y_data의 개수 : " + str(len(y_data)))
```

```
x_data의 개수 : 100
y_data의 개수 : 100
```



Edited by Harksoo Kim

## 실습

### 데이터 변환 (문자열 → 숫자)

```
from keras.preprocessing.text import Tokenizer
tokenizer = Tokenizer()
```

```
# spam, ham 라벨을 대응하는 index로 치환하기 위한 딕셔너리
label2index_dict = {'spam':0, 'ham':1}
```

```
# indexing 한 데이터를 넣을 리스트 선언
indexing_x_data, indexing_y_data = [], []
```

```
for label in y_data:
    indexing_y_data.append(label2index_dict[label])
```

```
# x_data를 사용하여 딕셔너리 생성
tokenizer.fit_on_texts(x_data)
```

```
# x_data에 있는 각 문장의 단어들을 대응하는 index로 치환하고 그 결과값을 indexing_x_data에 저장
indexing_x_data = tokenizer.texts_to_sequences(x_data)
```

```
print("x_data indexing 하기 전 : " + str(x_data[0]))
print("x_data indexing 하기 후 : " + str(indexing_x_data[0]))
print("y_data indexing 하기 전 : " + str(y_data[0]))
print("y_data indexing 하기 후 : " + str(indexing_y_data[0]))
```

```
x_data indexing 하기 전 : 60 until Jurong point, crazy.. Available only in bugis n great world ia e buffet... Cine there got amore wat...
x_data indexing 하기 후 : [38, 93, 239, 240, 241, 242, 53, 11, 243, 72, 94, 244, 245, 126, 246, 247, 73, 74, 248, 127]
y_data indexing 하기 전 : ham
y_data indexing 하기 후 : 1
```



Edited by Harksoo Kim

## 실습

### SVM 학습

```
from sklearn.svm import SVC

# 문장의 길이를 max_length으로 맞춰 변환
max_length = 60
for index in range(len(indexing_x_data)):
    length = len(indexing_x_data[index])

    if(length > max_length):
        indexing_x_data[index] = indexing_x_data[index][:max_length]
    elif(length < max_length):
        indexing_x_data[index] = indexing_x_data[index] + [0]*(max_length-length)

# 전체 데이터를 9:1의 비율로 나누어 학습 및 평가 데이터로 사용
number_of_train = int(len(indexing_x_data)*0.9)

train_x = indexing_x_data[:number_of_train]
train_y = indexing_y_data[:number_of_train]
test_x = indexing_x_data[number_of_train:]
test_y = indexing_y_data[number_of_train:]

print("train_x의 개수 : " + str(len(train_x)))
print("train_y의 개수 : " + str(len(train_y)))
print("test_x의 개수 : " + str(len(test_x)))
print("test_y의 개수 : " + str(len(test_y)))

svm = SVC(kernel='linear', C=1e10)
svm.fit(train_x, train_y)
```

```
train_x의 개수 : 90
train_y의 개수 : 90
test_x의 개수 : 10
test_y의 개수 : 10
SVC(C=10000000000.0, break_ties=False, cache_size=200, class_weight=None,
coef0=0.0, decision_function_shape='ovr', degree=3, gamma='scale',
kernel='linear', max_iter=1, probability=False, random_state=None,
shrinking=True, tol=0.001, verbose=False)
```



Edited by Harksoo Kim

## 실습

### SVM 평가

```
predict = svm.predict(test_x)
```

```
correct_count = 0
for index in range(len(predict)):
    if(test_y[index] == predict[index]):
        correct_count += 1
```

```
accuracy = 100.0*correct_count/len(test_y)
```

```
print("Accuracy: " + str(accuracy))
```

```
index2label = {0:"spam", 1:"ham"}
```

```
test_x_word = tokenizer.sequences_to_texts(test_x)
```

```
for index in range(len(test_x_word)):
    print()
    print("문장 : ", test_x_word[index])
    print("정답 : ", index2label[test_y[index]])
    print("모델 출력 : ", index2label[predict[index]])
```

```
Accuracy: 80.0
```

```
문장 : yeah do don 't stand to close tho you 'll catch so
정답 : ham
모델 출력 : spam

문장 : sorry to be a pain is it ok if we meet another nig
정답 : ham
모델 출력 : spam

문장 : smile in pleasure smile in pain smile when trouble
정답 : ham
모델 출력 : ham

문장 : please call our customer service representative on
정답 : spam
모델 출력 : spam

문장 : havent planning to buy later i check already lido
정답 : ham
모델 출력 : ham

문장 : your free ringtone is waiting to be collected simp
정답 : spam
모델 출력 : spam
```



Edited by Harksoo Kim

# 질의응답

---

Q & A

Homepage: <http://nlp.konkuk.ac.kr>  
E-mail: [nlpdrkim@konkuk.ac.kr](mailto:nlpdrkim@konkuk.ac.kr)



Edited by Harksoo Kim