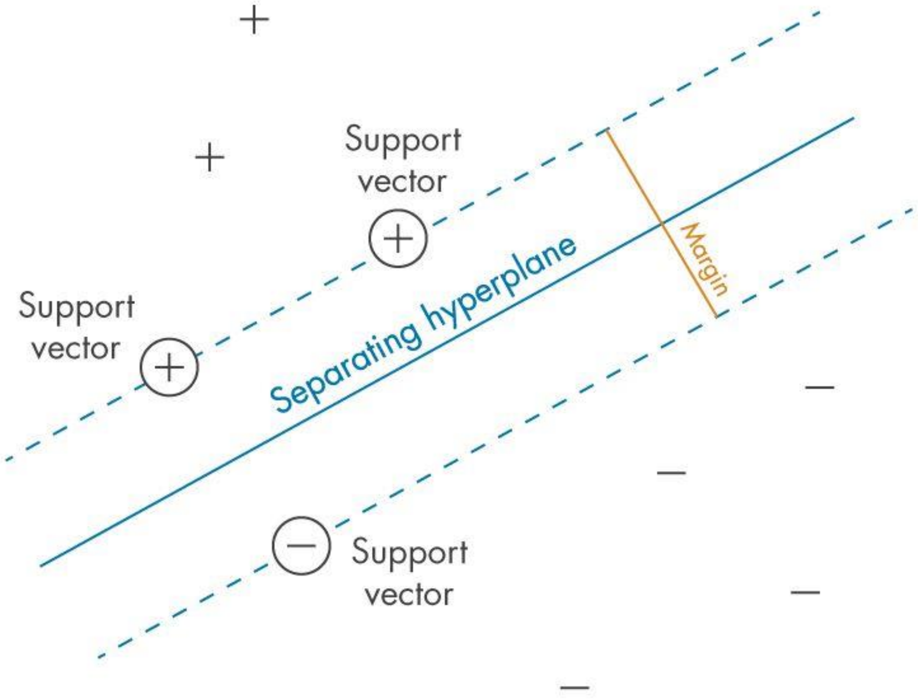
22기 정규세션

ToBig's 21기 김세민
ToBig's 22기 송진하

SVM
Dimensionality Reduction

Contents

Unit 01 | SVM 정의

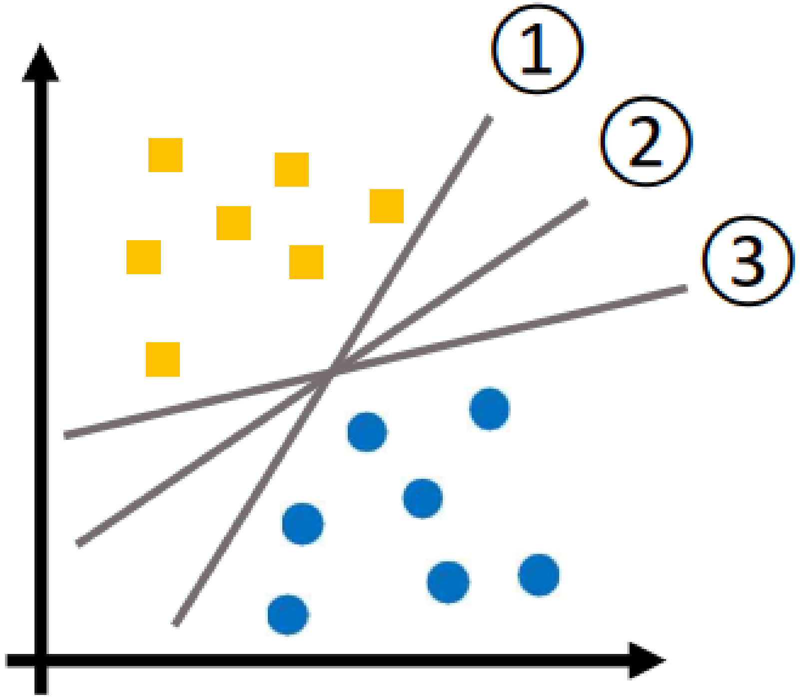


Support Vector Machine

주로 분류를 하기 위채 사용되는 기법이나 회귀에도 적용 가능.

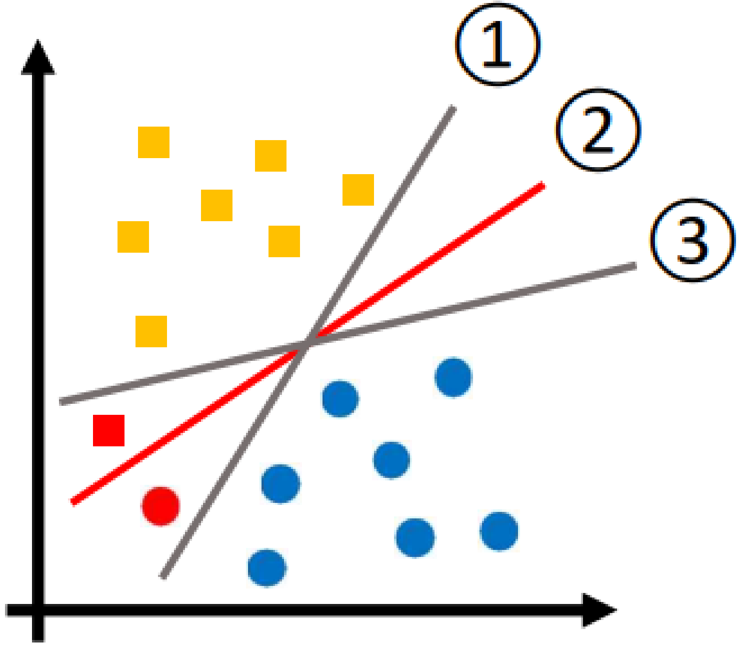
주요 목적은 데이터를 분류하는 최적의 초평면(hyperplane)을 찾는 것.

Unit 01 | SVM 정의



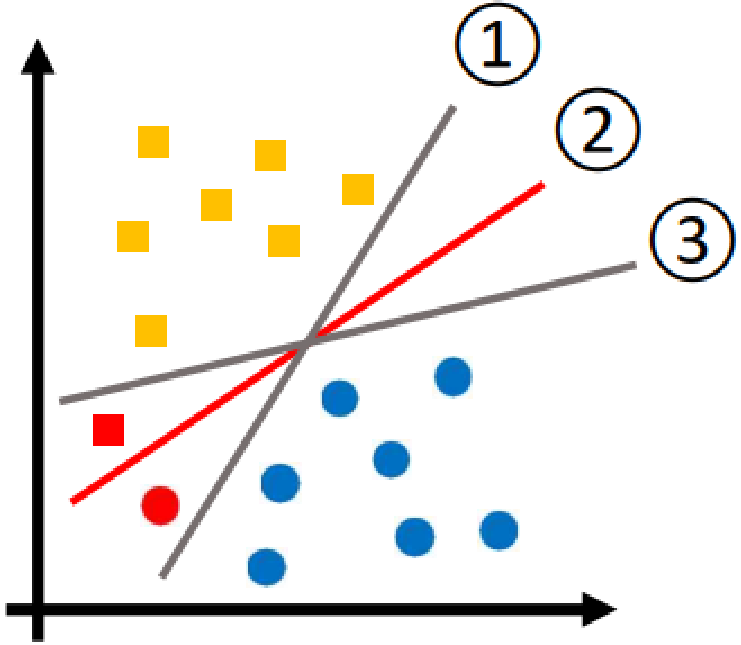
데이터를 가장 잘 나눈 선은?

Unit 01 | SVM 정의



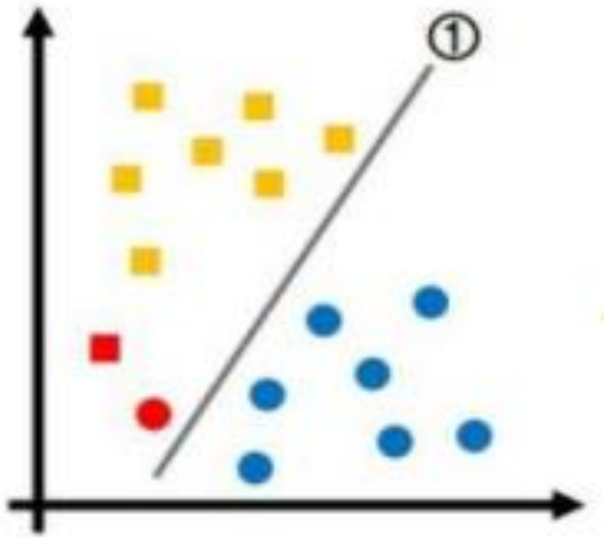
정답은 2번!

Unit 01 | SVM 정의

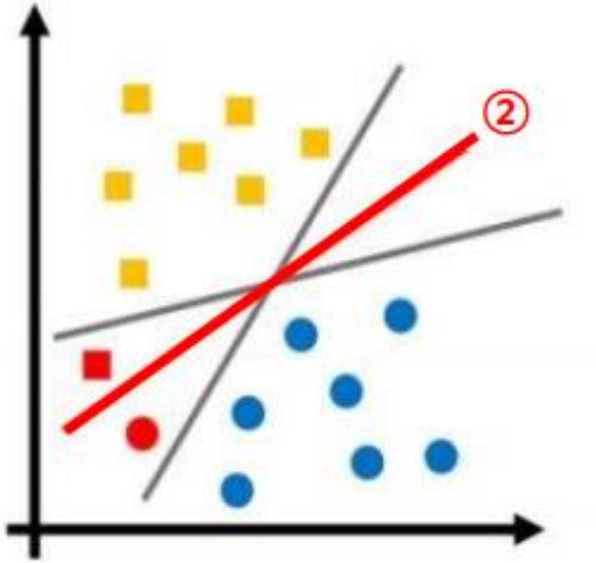


정답은 2번!

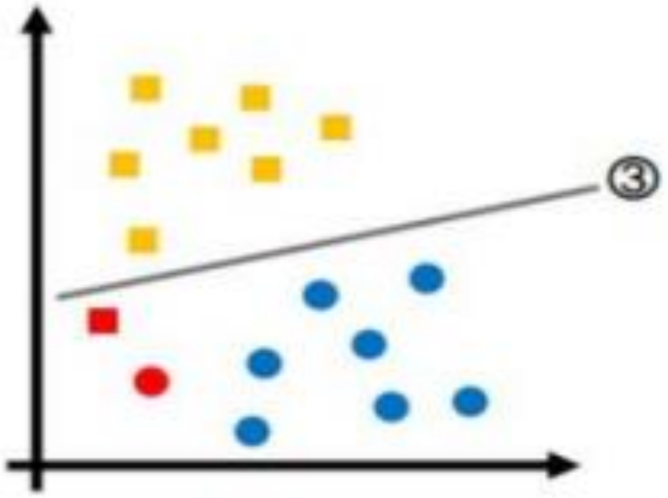
Unit 01 | SVM 정의



동그라미 분류X

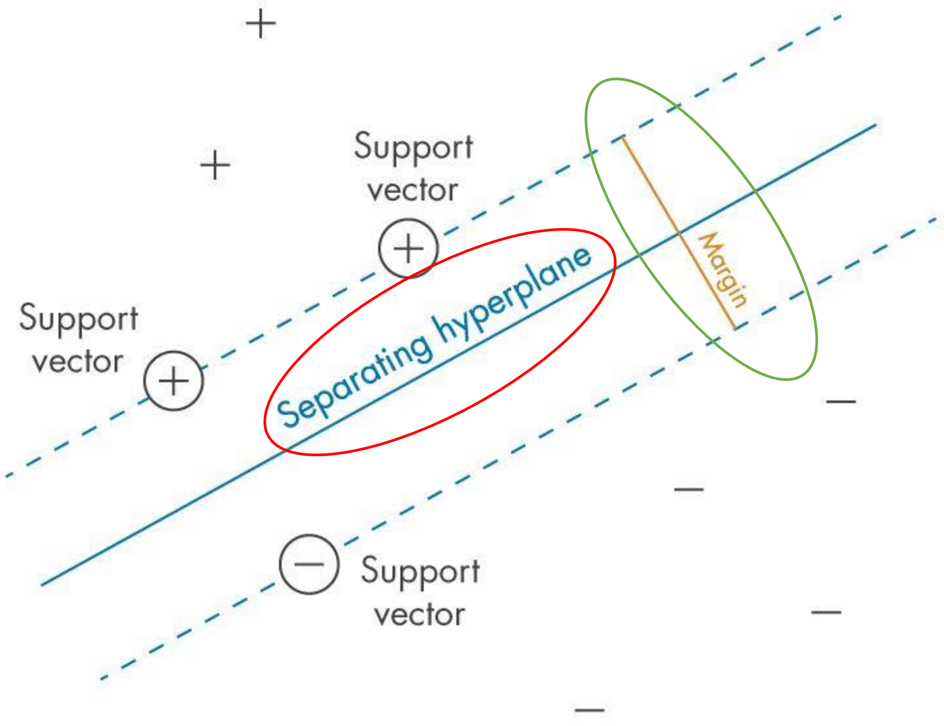


둘 다 분류O



네모 분류X

Unit 01 | SVM 정의



Hyperplane

데이터 포인트를 분리하는 기준이 되는 평면(초평면)

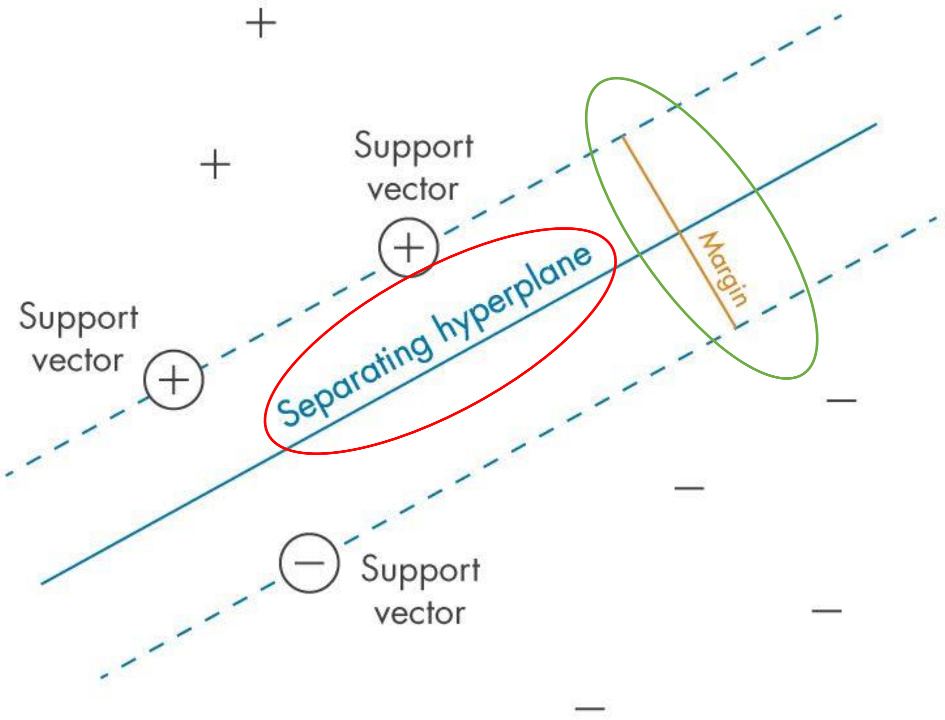
Support Vector

클래스별로 Hyperplane과 가장 가까운 데이터

Margin

결정경계와 서포터 벡터 사이의 거리 x 2

Unit 01 | SVM 정의



SVM 목표

마진을 최대화 하는 초평면을 찾는 것

Hyperplane

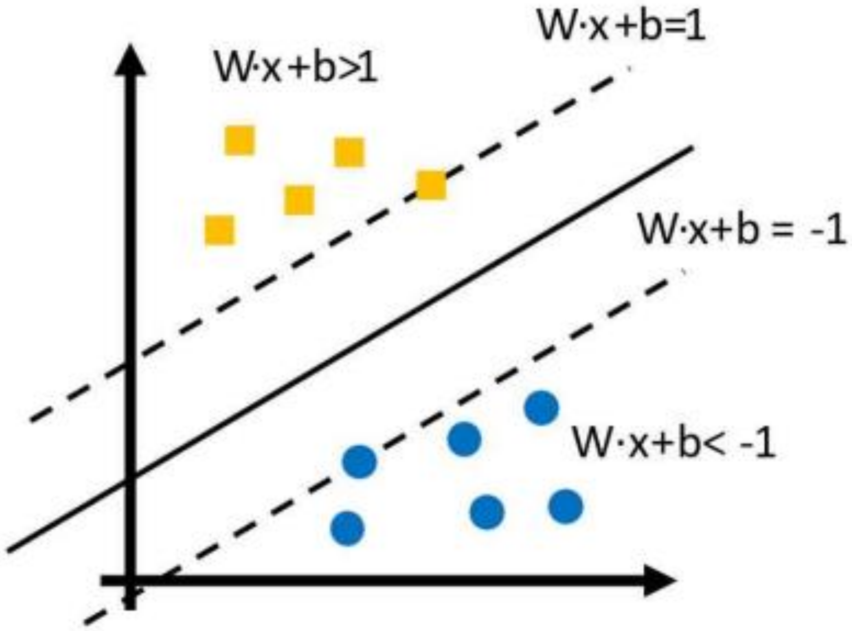
$$\boldsymbol{w} \cdot \boldsymbol{x} + \boldsymbol{b} = \mathbf{0}$$

$\boldsymbol{w}$는 초평면의 법선 벡터 (초평면의 방향 결정)

$\boldsymbol{x}$는 데이터 포인트 벡터

$\boldsymbol{b}$는 초평면이 원점으로부터 얼마나 떨어져 있는지를 나타냄

Hyperplane



$$y_i(w \cdot x_i + b) \geq 1$$

$$y_i = \begin{cases} +1 \text{ for } ■ \text{ sample} \\ -1 \text{ for } ● \text{ sample} \end{cases}$$

Margin (soft vs hard)

Hard Margin SVM

데이터가 선형적으로 완벽하게 분리 될 수 있는 경우에 사용

오분류 허용 X

Soft Margin SVM

데이터가 완벽하게 선형적으로 분리되지 않는 경우에 사용

현실 세계의 데이터는 노이즈와 이상치가 존재해 모든 데이터가 깔끔하게 선형적으로 나뉘지 않음

오분류 허용 O

Margin (soft vs hard)

Soft Margin SVM

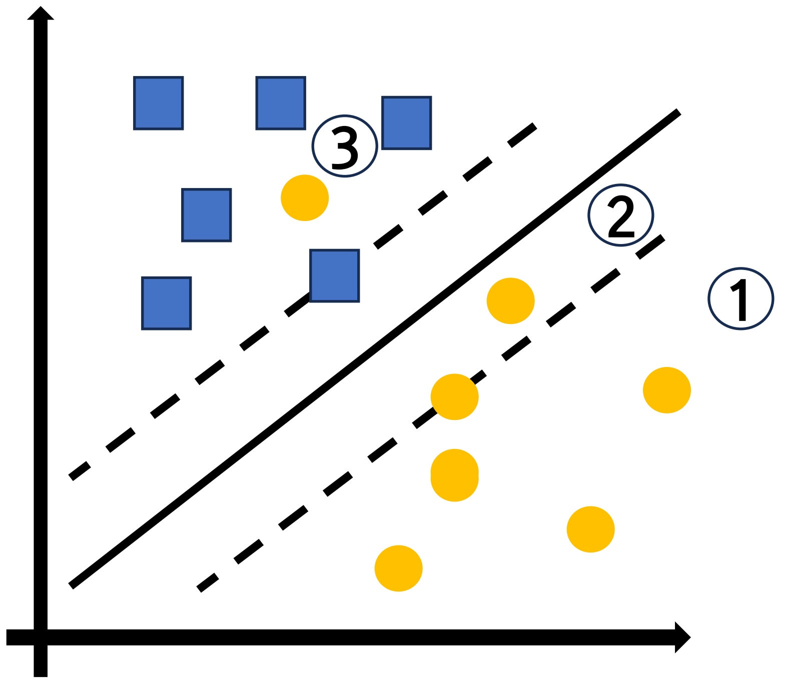
분류기의 각 영역에 있을 때

① $(\boldsymbol{w} \cdot \boldsymbol{x_i} + \boldsymbol{b}) \geq \mathbf{1}$ Correct

② $\mathbf{0} < (\boldsymbol{w} \cdot \boldsymbol{x_i} + \boldsymbol{b}) < \mathbf{1}$ Incorrect

③ $(\boldsymbol{w} \cdot \boldsymbol{x_i} + \boldsymbol{b}) < \mathbf{0}$ Incorrect

Slack variable(ξ_i) 사용



Margin (soft vs hard)

Soft Margin SVM

$$\boldsymbol{y_i}(\boldsymbol{w} \cdot \boldsymbol{x_i} + \boldsymbol{b}) \geq \mathbf{1} - \xi_i$$

$$\xi_i \geq 0$$

Soft Margin SVM 목적 함수

$$min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

C↑ 오류 허용↓ 모델의 복잡성 ↑

C↓ 오류 허용↑ 모델의 복잡성 ↓

Lagrange Multiplier Method

라그랑주 승수법이란 ?

최적화 문제를 제약 조건 하에서 해결하는 방법

SVM에서 최적의 초평면을 찾는 방법

목적 함수 $min \frac{1}{2} \|w\|^2$

제약 조건 $subject\ to\ y_i(w^T \cdot x_i + b) \geq 1$

Unit 01 | SVM 정의

$$L(w, b, \alpha) = \tfrac{1}{2}||w||^2 - \textstyle\sum_i \alpha_i [y_i(w \cdot x_i + b) - 1]$$

$$\frac{\partial L}{\partial w} = w - \sum_i \alpha_i y_i x_i = 0 \quad \Rightarrow \quad w = \sum_i \alpha_i y_i x_i$$

$$\frac{\partial L}{\partial b} = \sum_i \alpha_i y_i = 0$$

$$\begin{aligned}
\frac{1}{2}\|w\|^2 &= \frac{1}{2} w^T w \\
&= \frac{1}{2} w^T \Sigma_{j=1}^{n} a_j y_j x_j \\
&= \frac{1}{2} \Sigma_{j=1}^{n} a_j y_j (w^T x_j) \\
&= \frac{1}{2} \Sigma_{j=1}^{n} a_j y_j (\Sigma_{i=1}^{n} a_i y_i x_i^T + x_j) \\
&= \frac{1}{2} \Sigma_{j=1}^{n} \Sigma_{i=1}^{n} a_i a_j y_i y_j x_i^T x_j
\end{aligned}$$

$$\begin{aligned}
-\sum_{i=1}^{n} a_i (y_i (w^T x_i + b) - 1) &= -\sum_{i=1}^{n} a_i y_i (w^T x_i + b) + \sum_{i=1}^{n} a_i \\
&= -\sum_{i=1}^{n} a_i y_i w^T x_i - b \sum_{i=1}^{n} a_i y_i + \sum_{i=1}^{n} a_i \\
&= -\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j x_i^T x_j + \sum_{i=1}^{n} a_i
\end{aligned}$$

Unit 01 | SVM 정의

$$\min_{w,b} \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(w^T x_i + b) - 1) \longrightarrow \underset{\alpha}{\text{maximize}} \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$\textit{subject to} \sum_{i=1}^{n} \alpha_i y_i = 0,$$

$$\alpha_i \geq 0, i = 1,2, \ldots \ldots, n$$

듀얼 문제로 변환하면 계산 복잡성이 감소

후에 커널 트릭 사용하기에 용이

$$w = \sum_i \alpha_i y_i x_i \qquad y(w^T x + b) - 1 = 0$$

Unit 01 | SVM 정의

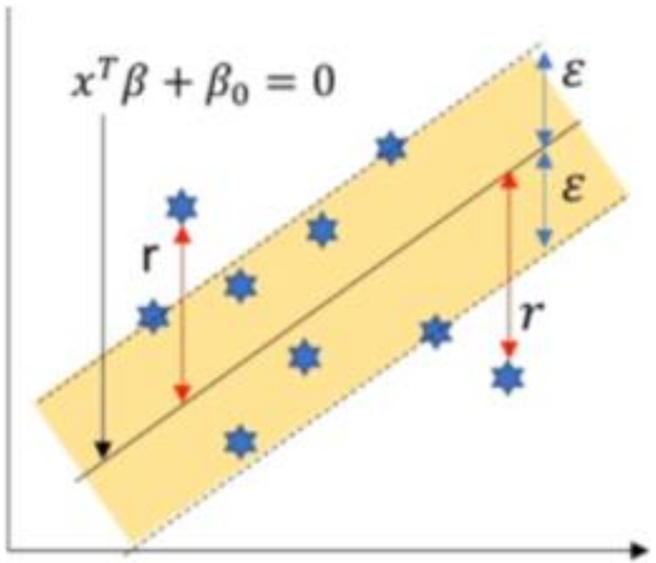
SVR

SVM의 회귀 버전

제한된 마진 오류 안에서 가능한 많은 샘플이 들어가도록 학습

폭은 하이퍼 파라미터 ε로 조절

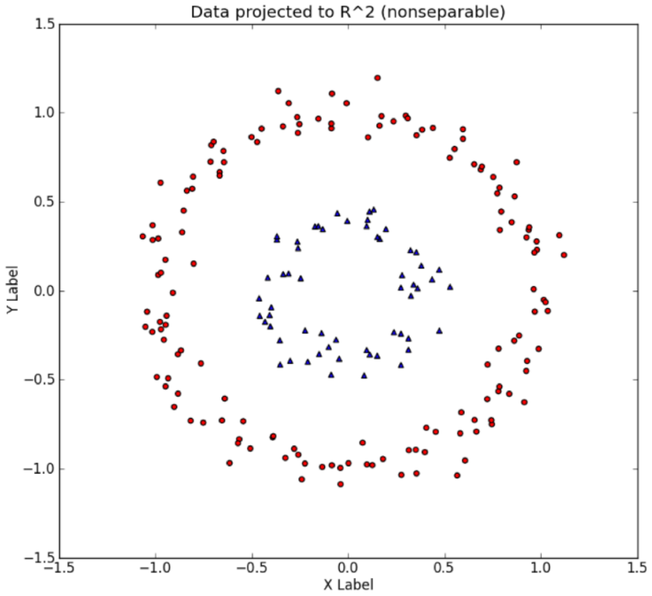
$\varepsilon\uparrow$ 폭$\uparrow$ $\varepsilon\downarrow$ 폭$\downarrow$



Unit 02 | Non-Linear SVM

실제 데이터 중 선형적으로 분류할 수 없는 데이터셋이 많음

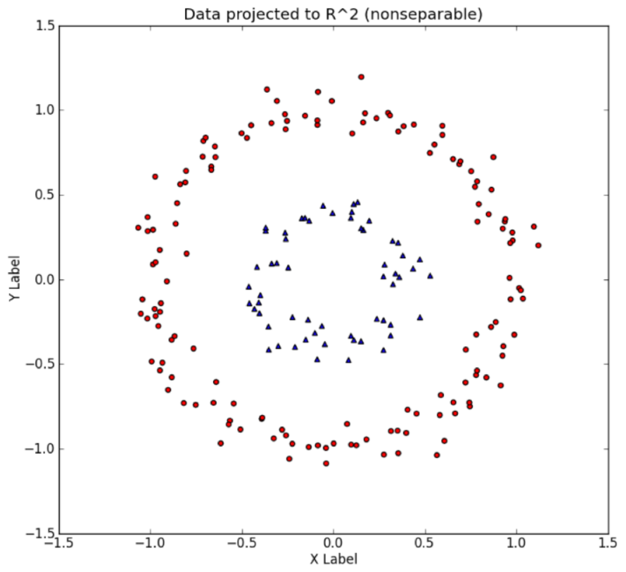
해결책으로 등장한 것이 커널 트릭

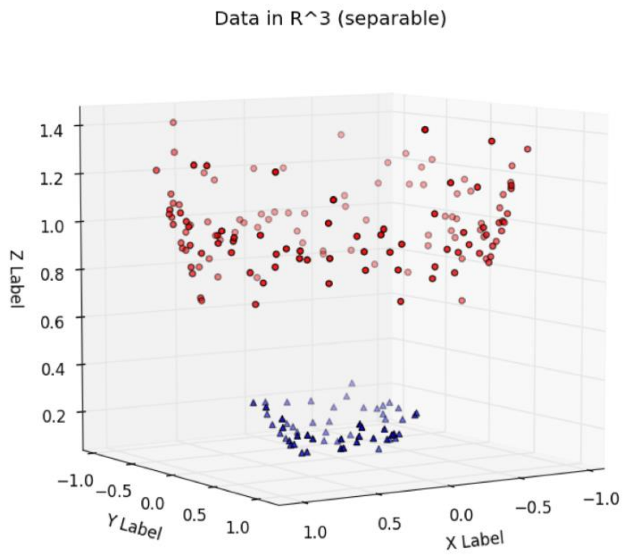


선형으로 분리 어려움

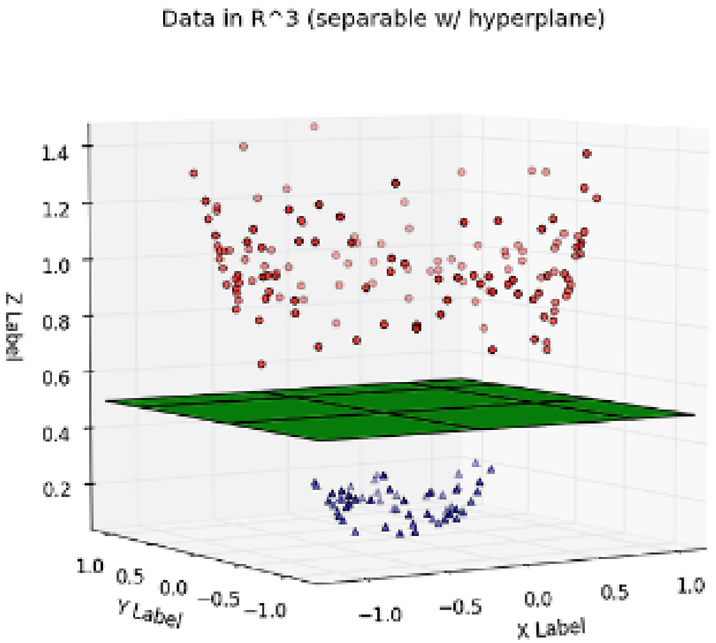
Kernel Trick 이란?

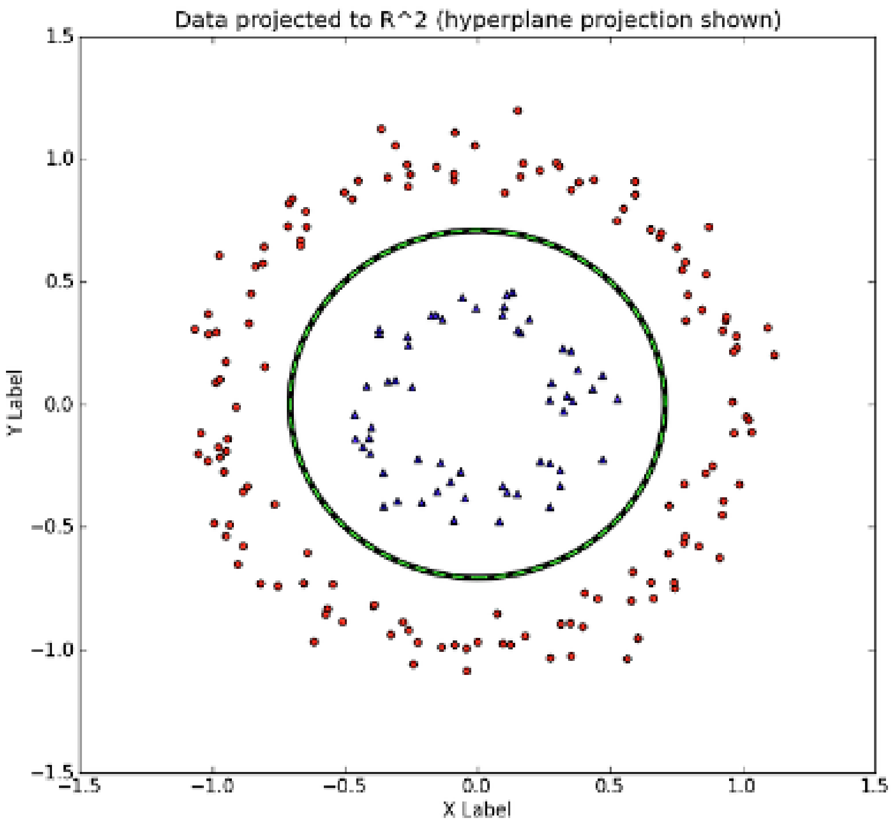
원래 공간에서는 선형적으로 분리할 수 없는 데이터를 고차원 특성 공간으로 변환하여
선형적으로 분리할 수 있도록 하는 기술





Unit 02 | Non-Linear SVM





Unit 02 | Non-Linear SVM

일반적인 커널 함수들

Linear Kernel: 단순한 선형 관계 사용 $K(x_i, x_j) = x_i \cdot x_j$

Polynomial Kernel: 다항식 관계 모델링 $K(x_i, x_j) = (x_i \cdot x_j + c)^d$

RBF or Gaussian Kernel: 데이터 간의 거리를 측정해 유사도 계산 $K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$

Sigmoid Kernel: 뉴럴 네트워크의 활성화 함수와 유사한 형태 $K(x_i, x_j) = \tanh(\alpha x_i \cdot x_j + c)$

Kernel Trick

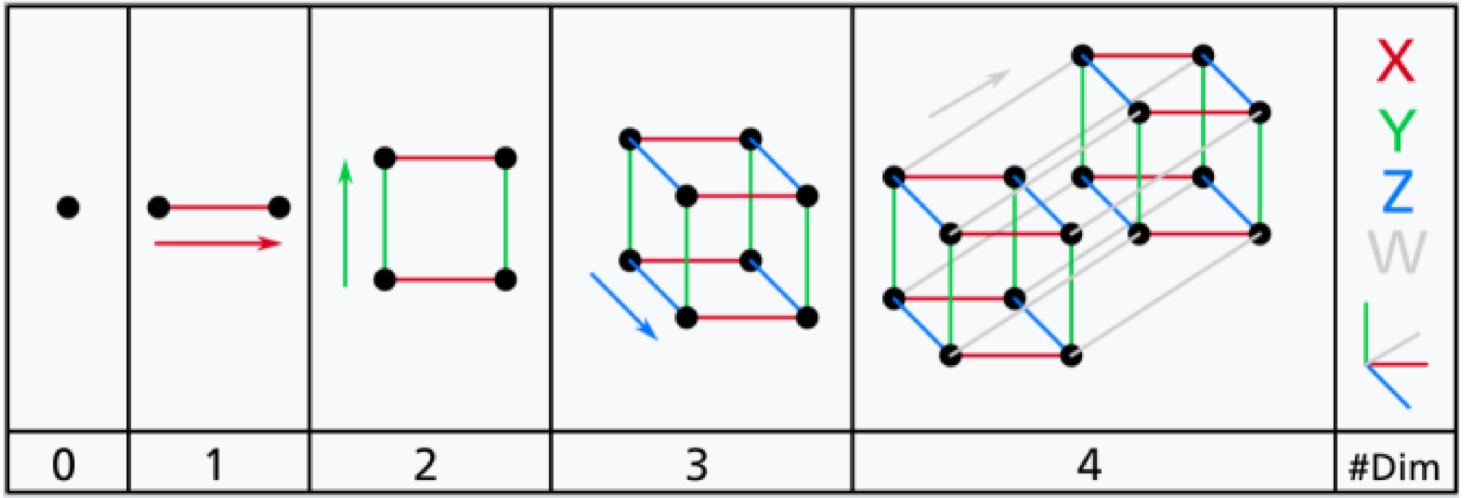
$$\max L_D(\alpha_i) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \Phi(x_i)^T \Phi(x_j)$$

SVM의 듀얼 문제는 데이터 포인트 간의 내적만을 포함

커널 함수를 사용해 직접 고차원 공간으로 데이터를 변환하지 않고도 고차원 공간에서의 연산을 원래 공간에서 수행 가능

차원이란?

공간의 성질을 나타내는 수로 공간에서 독립적으로 움직일 수 있는 방향의 개수



차원이란?

데이터를 구성하고 있는 속성의 개수

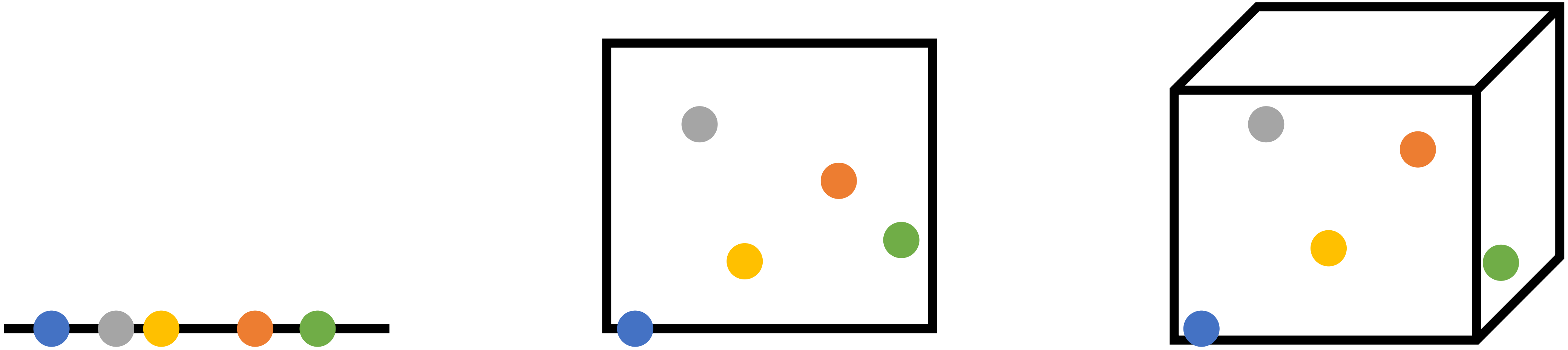
번호	수학1	확.통	물리	물리실험	미술	체육
1	80	85	75	80	40	60
2	30	32	25	30	90	55
3	70	75	90	92	80	50
4	40	32	54	40	90	70
5	95	90	100	95	60	90

Unit 03 | Introduction of Dimensionality Reduction

차원의 저주란?

차원이 증가함에 따라 데이터 분석 및 알고리즘 성능에 악영향을 미치는 다양한 현상

데이터 희소성, 계산 복잡도 증가, 과적합, 거리 측정의 신뢰성 감소 등



차원의 저주 극복 방법

차원 축소: 데이터의 차원을 줄임

특징 선택: 모델 성능에 중요한 영향을 미치는 주요 특징만 선택해 사용해 불필요한 차원 제거

정규화: 모델이 복잡해지는 것을 방지하기 위해 정규화 기법 사용해 과적합 방지

Unit 03 | Introduction of Dimensionality Reduction

차원 축소

번호	수학1	확.통	물리학	물리실험	미술	체육
1	80	85	75	80	40	60
2	30	32	25	30	90	55
3	70	75	90	92	80	50
4	40	32	54	40	90	70
5	95	90	100	95	60	90

특징 선택

번호	수학1	물리학	미술
1	80	75	40
2	30	25	90
3	70	90	80
4	40	54	90
5	95	100	60

특징 추출

번호	이과	미술
1	77.5	40
2	27.5	90
3	80	80
4	47	90
5	97.5	60

이과 = (수학1 + 물리학) /2

차원 축소

번호	수학1	확.통	물리학	물리실험	미술	체육
1	80	85	75	80	40	60
2	30	32	25	30	90	55
3	70	75	90	92	80	50
4	40	32	54	40	90	70
5	95	90	100	95	60	90

특징 선택

번호	수학1	물리학	미술
1	80	75	40
2	30	25	90
3	70	90	80
4	40	54	90
5	95	100	60

특징 추출

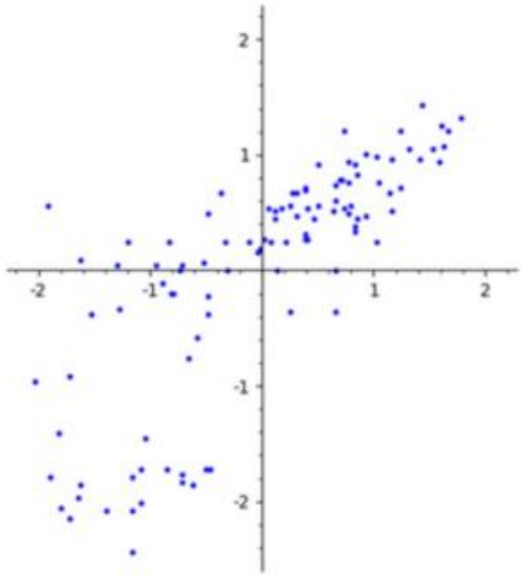
번호	이과	미술
1	77.5	40
2	27.5	90
3	80	80
4	47	90
5	97.5	60

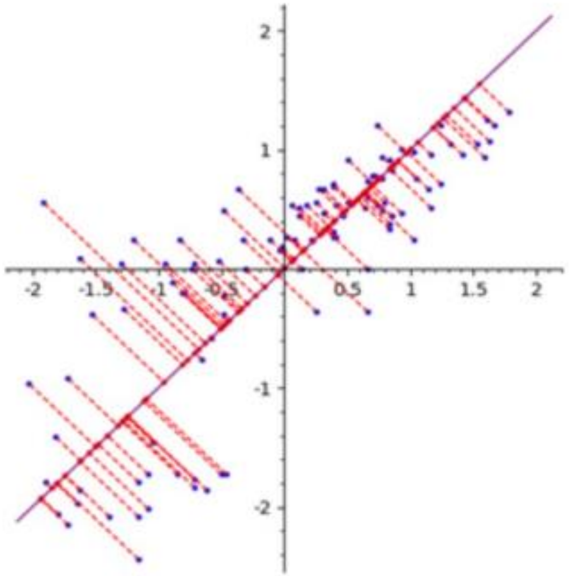
이과 = (수학1 + 물리학) /2

PCA
LDA

Principal Component Analysis (주성분 분석)

데이터의 변동성을 최대한 보존하면서 고차원 데이터를 저차원으로 축소하는 것

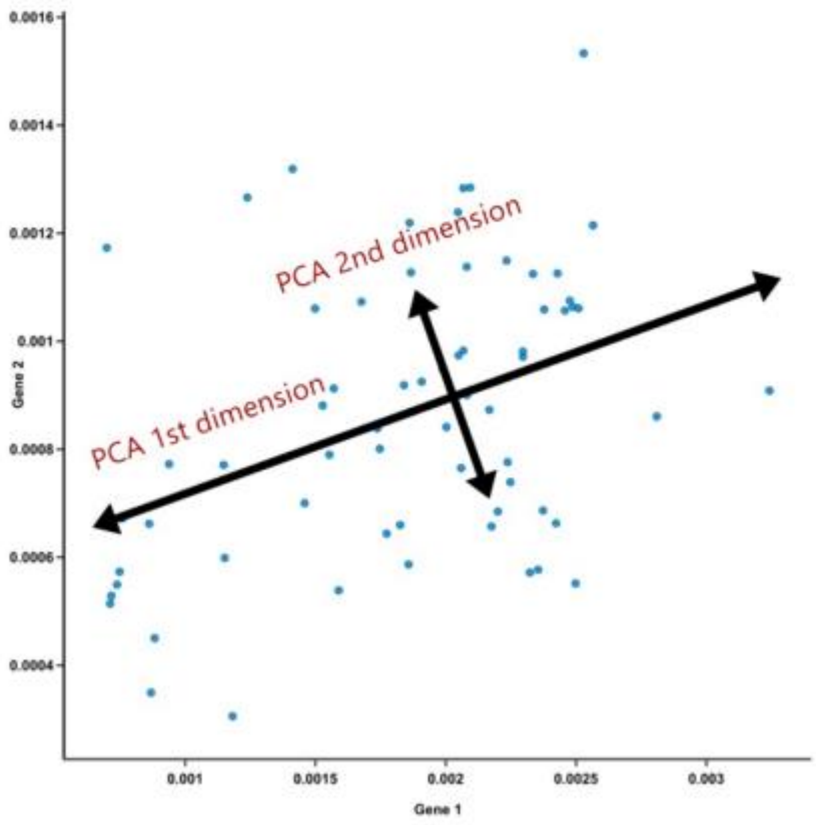




Principal Component Analysis 과정

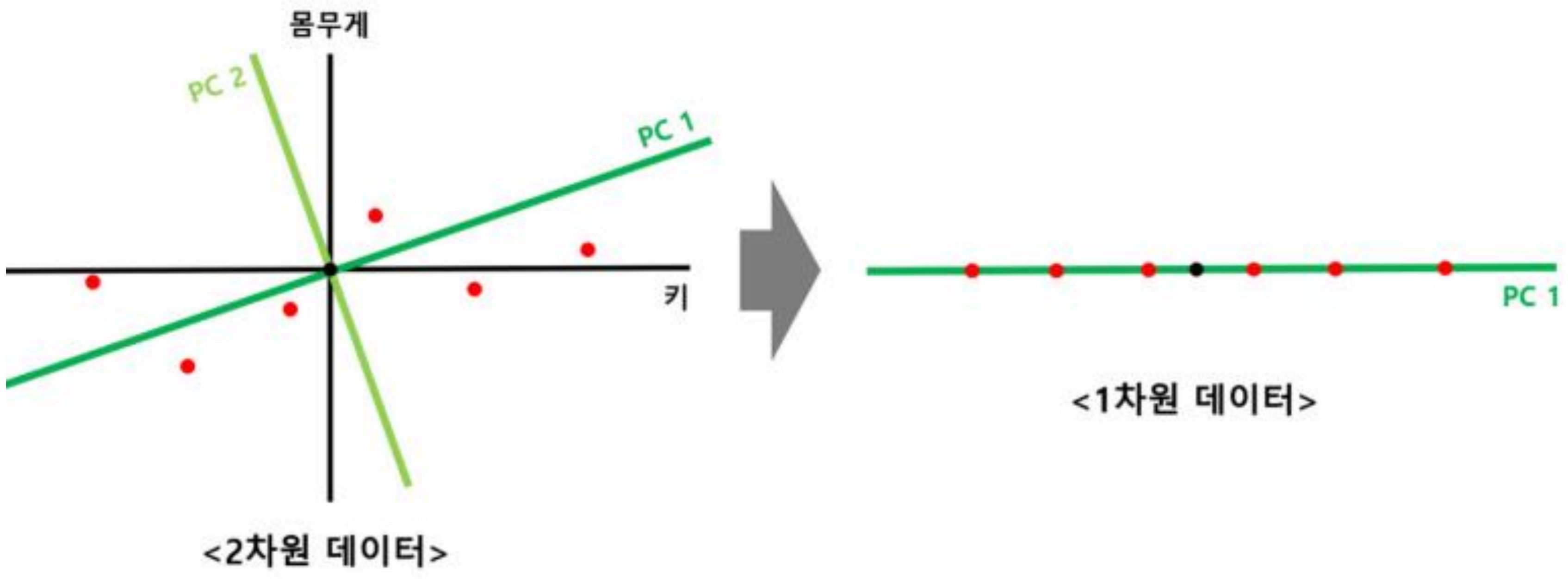
데이터의 분산을 최대화하는 방향으로 새로운 축(주성분)을 찾는다.

1. 분산이 가장 큰 벡터 선택
2. 첫 번째 벡터와 직교하는 벡터 중 분산이 가장 큰 벡터 선택
3. 두 번째 벡터와 직교하는 벡터 중 분산이 가장 큰 벡터 선택
4. 위와 같은 과정을 반복



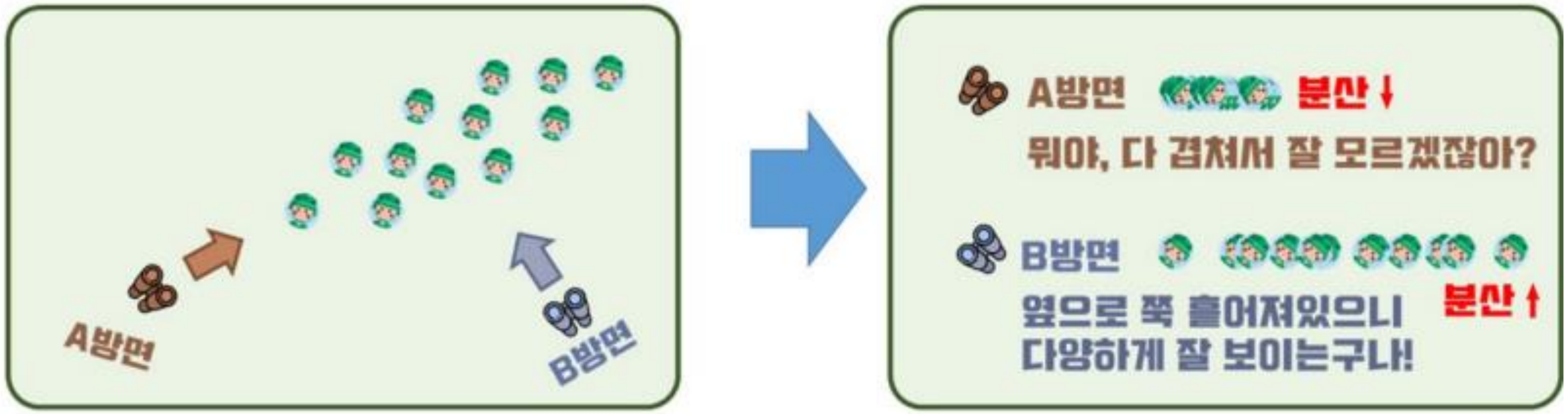
Principal Component Analysis 결과

사전에 정의한 N개의 벡터를 선택한 후, N개의 벡터가 나타내는 공간에 투영



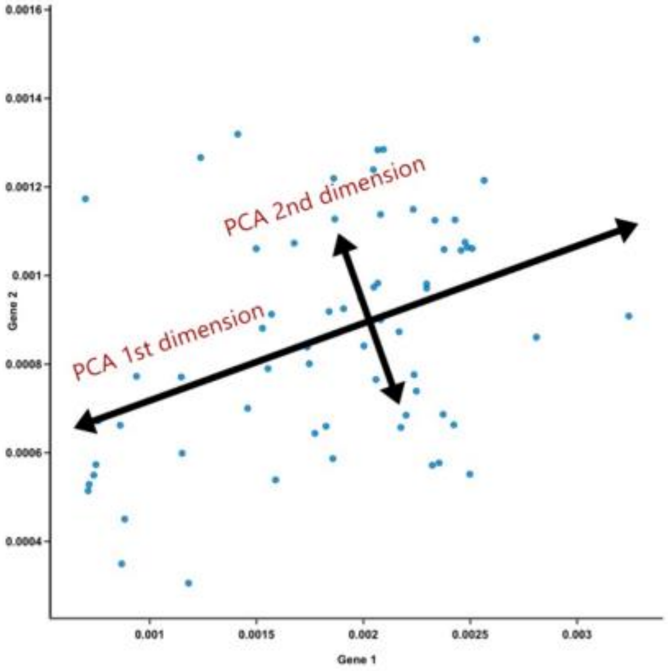
최대 분산을 선택하는 이유

분산이 큰 벡터가 중요한 정보를 담고 있다고 가정



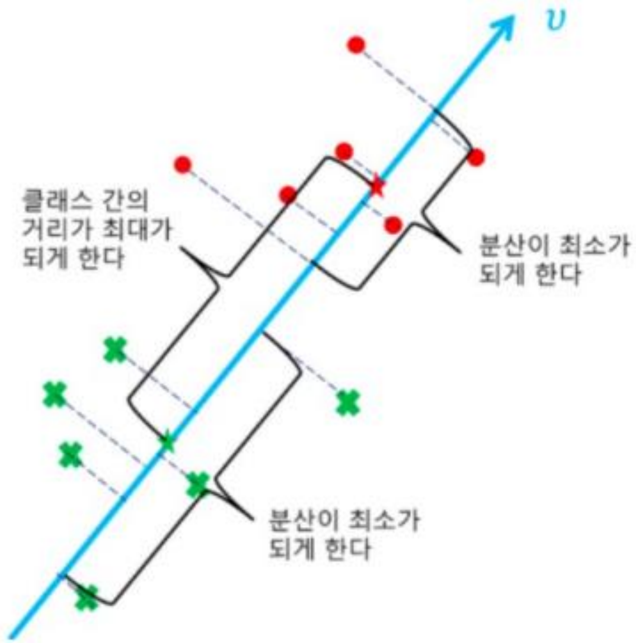
직교 벡터를 선택하는 이유

직교 벡터는 상호 독립적이어서 데이터의 중요한 정보를 중복 없이 독립적으로 설명 가능



Linear Discriminant Analysis (선형 판별 분석)

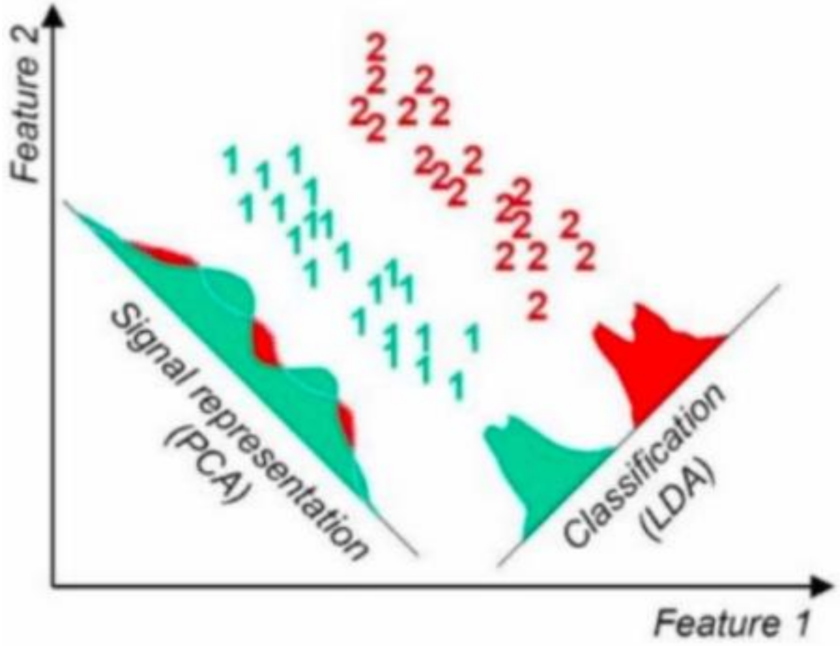
클래스 간의 거리가 최대가 되게 하면서
클래스 내의 분산이 최소가 되게 하는 벡터를 찾음



LDA와 PCA의 비교

PCA: 데이터의 전체 변동성을 최대화하는 방향으로 데이터 변환

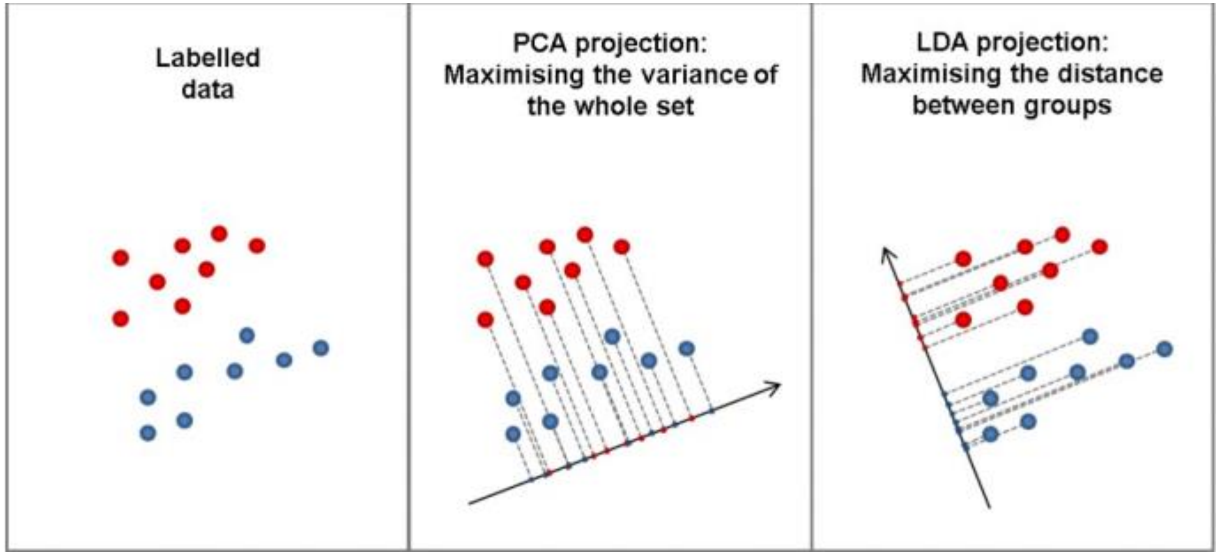
LDA: 클래스 간 분리를 최대화 하는 방향으로 데이터 변환



LDA와 PCA의 비교

PCA: 데이터의 분산이 가장 큰 벡터 선택, 비지도 학습 (클래스 레이블 정보 사용X)

LDA: 클래스 간 분리를 최대화 하는 방향으로 데이터 변환, 지도 학습(클래스 레이블 정보 사용O)

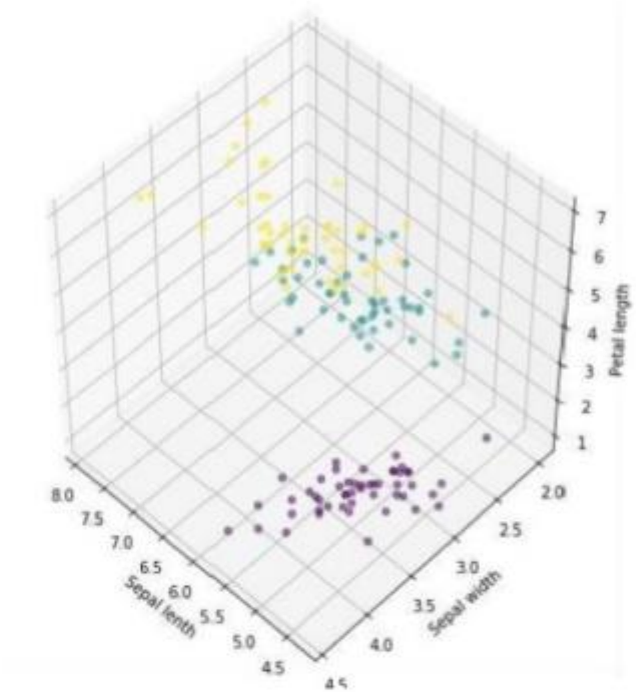


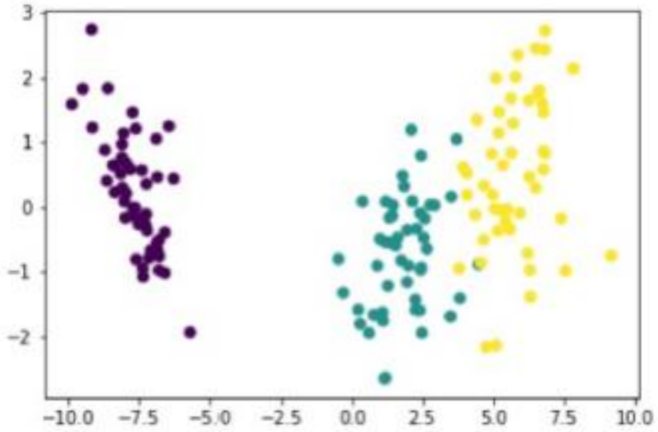
Linear Discriminant Analysis 과정

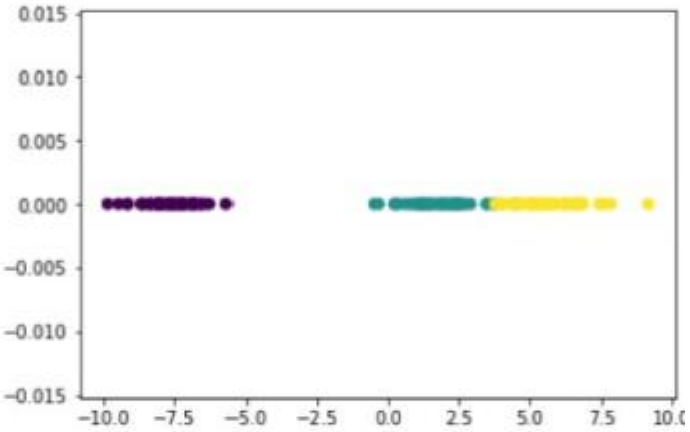
1. 데이터 준비
2. 클래스별 평균 벡터 계산
3. 클래스 내 공분산 행렬(S_W)과 클래스 간 공분산 행렬(S_B) 계산
4. 클래스 간 분산을 최대화 하고 클래스 내 분산을 최소화하는 변환 행렬 계산
5. S_W의 역행렬과 S_B를 곱한 행렬의 고유값을 계산해 가장 큰 고유값에 해당하는 고유벡터 선택
6. 원본 데이터를 변환 행렬로 변환해 새로운 저차원 공간에 투영

Unit 04 | PCA, LDA

Linear Discriminant Analysis 결과







Unit 00 | 실습

SVM & Dimensionality Reduction

- Iris 데이터셋에 대해 svm 분류 수행 및 커널 별 시각화
- Pca와 lda를 활용하여 분류 수행 및 결과 시각화

Unit 00 | 과제

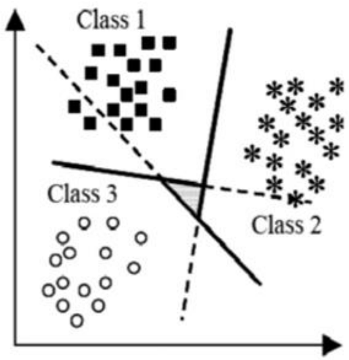
Multiclass SVM

- 이제껏 공부한 SVM 모델은 hyperplane을 기준으로 데이터를 이진 분류하는 데 초점이 맞춰져 있음
  - Class가 여러 개인 경우에는?

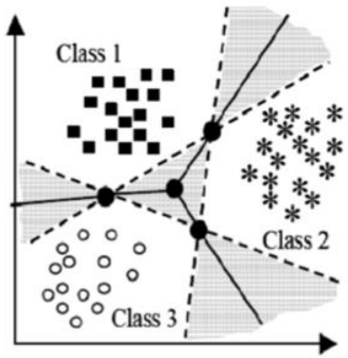
- One vs One (OvO)
  - N개의 클래스가 있을 때, 모든 클래스에 대해 1:1로 이진 분류를 진행
    ➔ Voting 기법으로 최종 예측 클래스 결정

- One vs Rest (OvR)
  - N개의 클래스가 있을 때, 모든 클래스에 대해 1 : N-1로 이진 분류를 진행
    ➔ 가장 높은 점수를 내는 클래스를 최종 예측 클래스로 결정

OvO SVM



OvR SVM



Unit 00 | 과제

Multiclass SVM

과제: 내 손으로 만드는 DIY SVM★

<u>(1) SVM from scratch</u>

- Numpy만을 사용하여 SVM Classifier 구현
- 참고 사이트 (과제 파일 참조)에 있는 코드 및 구현 방법을 참고하여 SVM의 가중치 업데이트 식과 predict 함수를 올바르게 작성해주시면 됩니다.

<u>(2) Dimensionality Reduction</u>

- Pca와 lda를 사용하여 MNIST 데이터셋의 차원을 축소시킨 뒤, 앞서 구현한 SVM Classifier로 분류 작업을 수행해주세요.

Unit 00 | 과제

Multiclass SVM

과제: 내 손으로 만드는 DIY SVM★

(3) Multiclass SVM

- 기본적인 사이킷런 SVM 클래스는 다중 분류 기능을 지원하지만 여기서는 사용하시면 안됩니다. 1번 과제에서 구현한 SVM Classifier을 사용하셔야 합니다.
- OvO 방법론을 기반으로 하여 구현하고, 결과를 출력해주세요. (accuracy score, classification report, etc.)
- 사이킷런의 공식 문서를 참고하여 구현하여도 무방합니다.
- 만약 voting 결과가 동점으로 나왔을 시에는 임의로 처리 방식을 결정해주세요.
  - Ex) decision function 이용, 가장 개수가 많은 클래스 활용, 랜덤으로 선택 등등

Unit 00 | 출처

https://kr.mathworks.com/discovery/support-vector-machine.html

https://m.blog.naver.com/wooy0ng/222667389075

https://velog.io/@cyeongy/SVM-Support-Vector-Machine

https://m.blog.naver.com/jaehong7719/221928401297

ToBig's 20기 박준 님 강의자료

ToBig's 20기 양승빈 님 강의자료

https://velog.io/@gangjoo/ML-%EC%B0%A8%EC%9B%90-%EC%B6%95%EC%86%8C-LDA-Linear-Discriminant-Analysis

https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

https://velog.io/@claude_ssim/%EA%B8%B0%EA%B3%84%ED%95%99%EC%8A%B5-Multi-Class-Classification-and-Kernel-Method

Q & A

들어주셔서 감사합니다.