SVM

(Support Vector Machine)

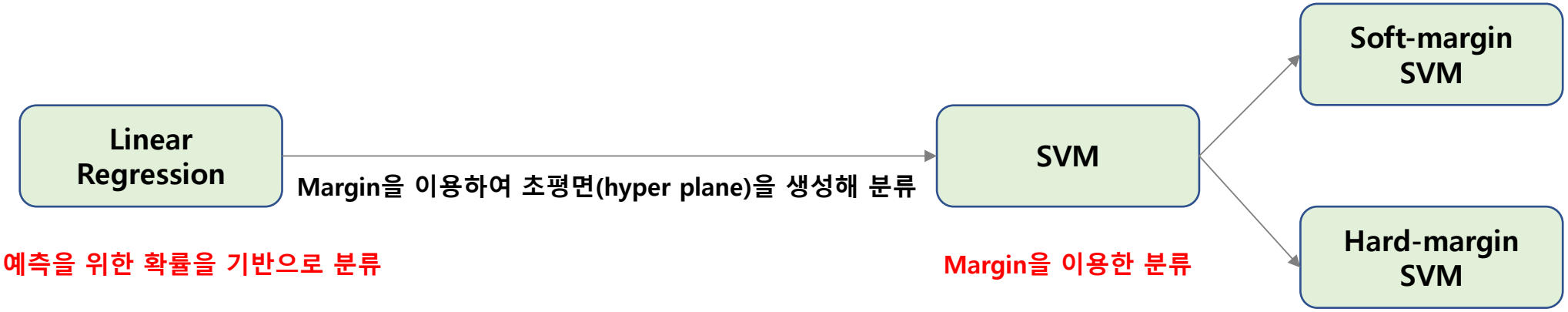
질문으로 이해하기

Coursera Machine Learning by Andrew NG 강의에서
명확하게 이해가 안되는 SVM만 따로 정리
(내가 궁금한것 중심으로...)

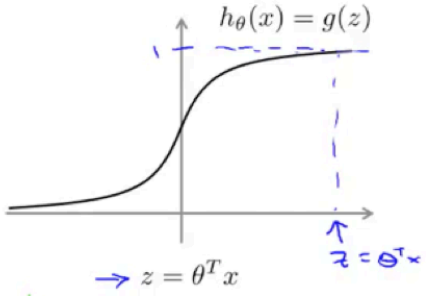
2017.06
freepsw

Week7. SVM 이해

데이터를 분류하는 최적(margin을 최대화) 선을 찾는 것



$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$



- $H_\theta(x)$가 정답의 확률을 의미함.
- 이 확률을 input값으로 Loss를 계산하고,
- Loss를 최소화하면서 분류를 진행함.

- 즉, 학습 데이터에서 loss를 최소화 하는 관점으로 접근
- ➔ 학습에 최적화되어 Test 데이터를 잘 분류하지 못하는 경우가 있음.

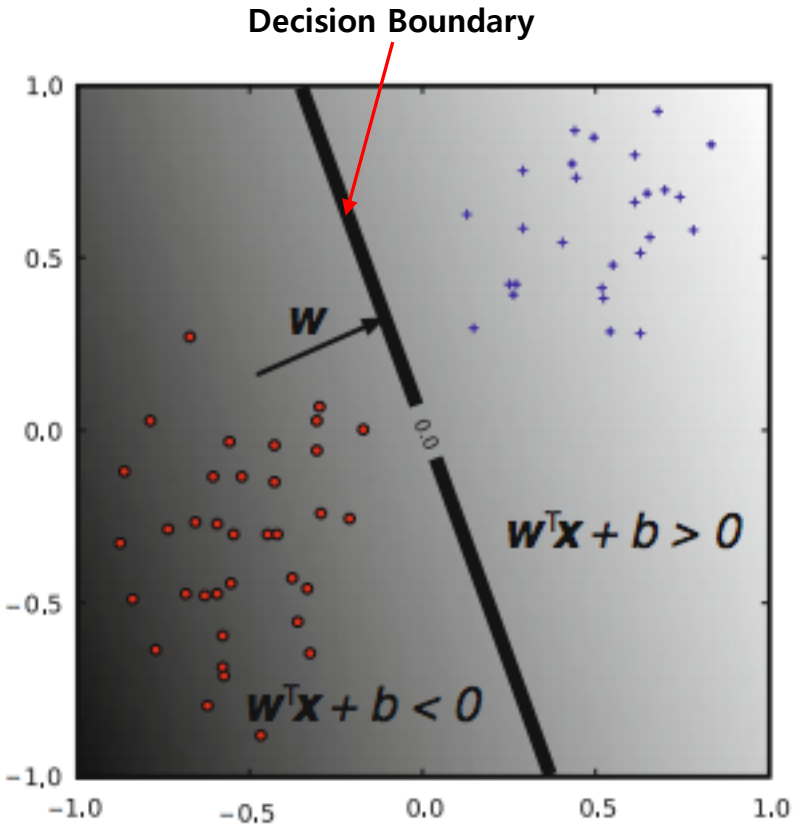
- Margin을 최대화하는 decision boundary를 제공함.

- 장점
  - 학습 데이터에 over fitting 방지
  - 비선형 데이터 분류도 가능 (Kernel method활용)

- 단점
  - 학습 데이터의 margin이 적을 때 문제 발생가능
  - Support vector(margin) 근처의 데이터만 고려함.
  - 고차원 데이터에서 효율적 (비선형 → 고차원 변환)

Week7. SVM – Decision Boundary는 어떻게 결정하는가?

가중치 벡터(W)에 직교하면서, margin이 최대가 되는 선형을 찾는다



- W : 가중치 벡터
- X : 입력값
- b : 원점(origin)에서 이동한 거리
- 현재는 margin 없이 분류함 (SVM이 아님, Decision boundary 이해를 위한 그림)

Decision Boundary가 되려면?

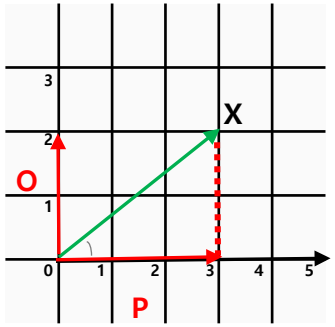
- $W^TX + b = 0$ 의 수식을 따르는 선형을 Decision Boundary 라고함
- 또한 가중치 벡터 W는 Decision Boundary와 직교(90 도)해야 한다.

왜 W 와 Decision Boundary는 직교해야 하나?

- 계산상의 편의를 위해서, b=0(원점을 지남)로 가정하면
- $W^TX = 0$ → Decision Boundary 로 정의할 수 있고,
- 2개 벡터 내적의 결과가 0이 되는 각도는 90 이므로 직교한다고 표현함.

왜 벡터 내적의 합이 0 이면 90도가 되는가?

- 2개 벡터(W, X)의 내적은 P가 된다.
- 이때 내적의 값이 가장 커지는 각도는 0도가 되고,
- 내적의 값이 0이 되는 각도는 90도가 된다.
- 즉, WX=0일 때, 벡터 X의 모든 점들은 벡터 W에 직교한다는 의미가 됨.



90도 → WX = 0　　0도 → WX = 최대

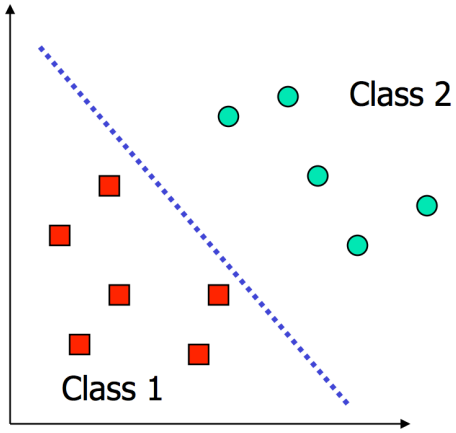
http://www.cs.colostate.edu/~asa/pdfs/howto.pdf

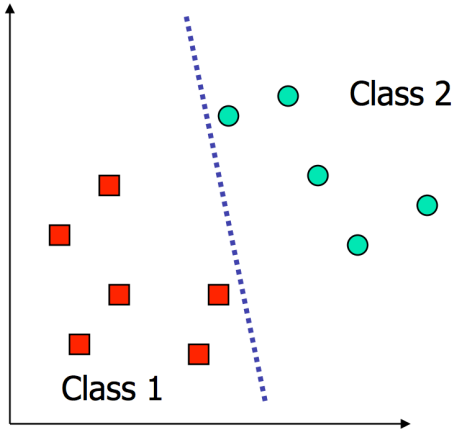
http://mrw0119.tistory.com/12

Week7. SVM – Margin을 최대로 하는 Decision Boundary?

Margin이 커지면 학습 데이터에 최적화 되지 않고, 실제 데이터의 분류정확도 향상

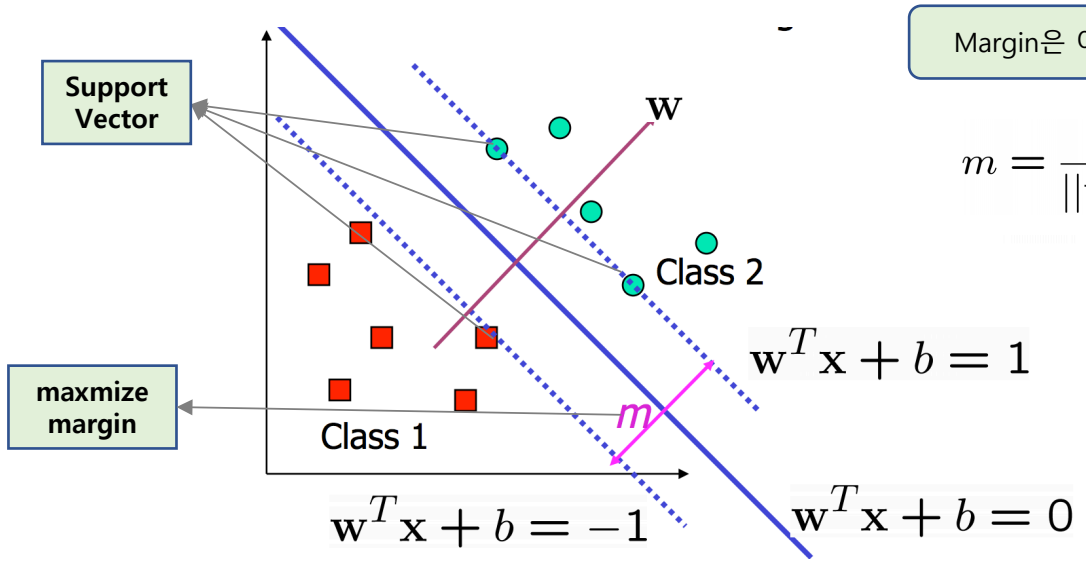
Margin이 없는 경우 Decision Boundary





- 직관적으로 봤을 때,
- 분류하는 선이 향후 데이터가 추가되었을때 제대로 분류하지 못할 가능성이 커보임

Margin을 활용한 Decision Boundary



- Decision boundary와 평행하고, 가중치 벡터(W)와 직교하며,
- Decision boundary와 가장 가까운 좌표와의 거리가 최대가 되는(margin)
- 3개의 벡터 (support vector)를 기준으로
- Decision boundary 결정

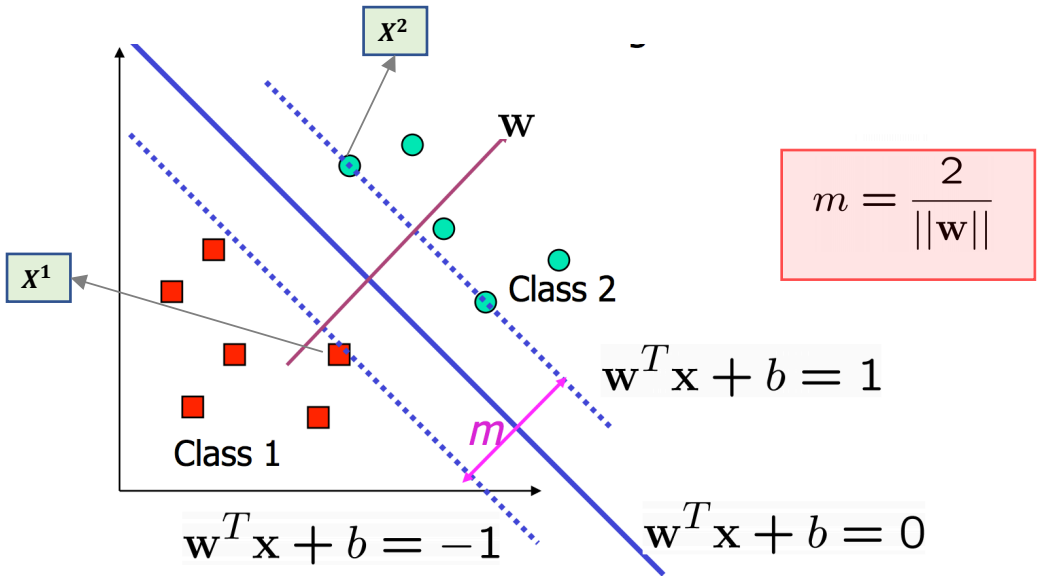
- $W^TX + b = 1$ (b=0으로 가정했을때, $W^TX > 1$ 경우 Class2로 분류)
- $W^TX + b = -1$ (b=0으로 가정했을때, $W^TX < -1$ 경우 Class1로 분류)

https://www.cise.ufl.edu/class/cis4930sp11dtm/notes/intro_svm_new.pdf

Week7. SVM – Margin은 어떻게 계산할까?

Margin이 커지면 학습 데이터에 최적화 되지 않고, 실제 데이터의 분류정확도 향상

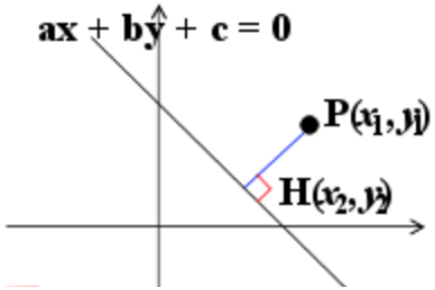
Margin 계산방식



$$m = \frac{2}{||w||}$$

- 각 클래스에서 decision boundary와 가장 가까운 point를 $X^2(Class\ 2), X^1(Class1)$ 이라고 가정하면, (b=0로 가정)
- $W^T X^1 = -1\,, W^T X^2 = 1$
- $m = W^T X^2 - W^T X^1$ ➔ 이를 직선($W^T X = 0$) 과 X^1, X^2간 거리로 분리
- $= \frac{|1|}{||W||} + \frac{|-1|}{||W||}$ 거리의 개념으로 절대값을 사용하고, 2개의 거리를 더함
- $= \frac{|2|}{||W||}$

직선의 방정식을 이용한 margin 계산



직선 PH 계산하기

- 직선 PH와 점 P 사이의 거리를 구하는 공식
- 직선 (3x + 4y -3 =0)과 점(2,3)의 거리 ➔

$$d = \frac{|ax_1 + by_1 + c|}{\sqrt{a^2 + b^2}}$$

$$= \frac{|3 \times 2 + 4 \times 3 - 3|}{\sqrt{3^2 + 4^2}} = \frac{15}{5} = 3$$

- 위 공식을 이용하여 왼쪽의 방정식을 대입해 보면, (b=0로 가정)
- $W^T X^1 = 1$: $W^T X + b = 1$ 인 직선에서 X^1 포인트를 지나는 선
- $W^T X^2 = -1$: $W^T X + b = -1$ 인 직선에서 X^2 포인트를 지나는 선
- $W^T X = 0$ 와 X^1과의 거리는 ➔ $\frac{|W^T X^1|}{\sqrt{W^2}} = \frac{|W^T X^1|}{||W||} = \frac{|-1|}{||W||}$
- $W^T X = 0$ 와 X^1과의 거리는 ➔ $\frac{|W^T X^2|}{\sqrt{W^2}} = \frac{|W^T X^2|}{||W||} = \frac{|1|}{||W||}$

https://www.cise.ufl.edu/class/cis4930sp11dtm/notes/intro_svm_new.pdf

http://mathbang.net/453

An Idiot's guide to Support vector machines

Week7. SVM – Margin은 어떻게 최대화 하는가?

먼저 SVM은 어떻게 모델을 최적화 하는지 이해해 보자

SVM Cost Function

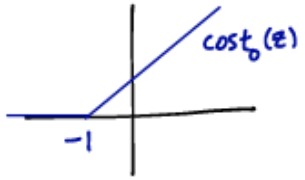
- 정답의 유형(0 or 1)에 따라서 Cost가 증가하는 방향이 다르고,
- Logistic Regression의 Log기반의 cost함수(cross-entropy)와 다르게, hinge loss 함수를 사용함.
-
- Hinge loss : $\ell(y) = \max(0, 1 - t \cdot y)$

$$\min_{\theta} \; C\left[\sum_i y^{(i)} \text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)})\,\text{cost}_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

If $y=1$,

We want $\theta^T x \geq +1$

(not just ≥ 0)



If $y=0$,

We want $\theta^T x \leq -1$

(not just < 0)

https://wikidocs.net/4280

Hinge loss 란?

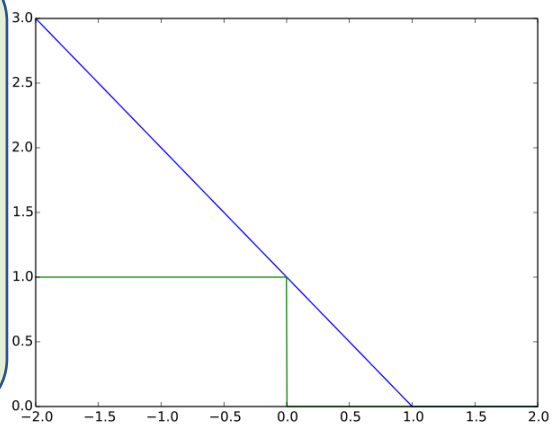
- Hinge loss란 margin을 최대화 하는 분류모델(SVM)에서 사용하는 loss function
- 공식 : $\ell(y) = \max(0, 1 - t \cdot y)$
-
- t = 의도한 결과 (예를 들면, +1, -1)
- Y = 분류 결과 ($y = WX + b$), x = 분류할 *input point*
-
- 여기서 기존 loss와 다른점이 y가 예측된 class label의 확률이 아니라, wx +b의 결과값이라는 것이다. (Logistic regression은 y에 sigmoid함수의 결과인 확률값을 입력함)
-
- 만약 t와 y가 동일한 sign(y가 분류한 값이 정답인라는 의미)을 가지면,
- |y| > 0, hinge loss = 0가 된다

Sign은 동일한데, y가 소수점으로 나면?

Max (0, 1 – 1*0.5) = 0.5 ???

➔ margin 영역에 포함된다고 판단하고, 페널티(0.5)를 부여

➔ -이면 maring에서 떨어졌다고 판단하여 패널티 부여 안함



- 정답은 1인 경우
-
- 파란 선 : hinge loss
- 녹색 선 : zero-one loss
- t = 1 일때,
-
- Y > 1 (예측이 맞음) → loss = 0
- Y < 1 (예측이 틀림) → loss 증가

Week7. SVM – Margin은 어떻게 최대화 하는가?

먼저 SVM은 어떻게 모델을 최적화 하는지 이해해 보자

Logistic Regression 과 비교한 SVM 최적화

Support vector machine

Logistic regression:

$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\underbrace{\left(-\log h_\theta(x^{(i)})\right)}_{\text{cost}_1(\theta^T x^{(i)})} + (1-y^{(i)})\underbrace{\left(-\log(1-h_\theta(x^{(i)}))\right)}_{\text{cost}_0(\theta^T x^{(n)})}\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Support vector machine:

$$\min_\theta \; C\sum_{i=1}^{m} y^{(i)}\text{cost}_1(\theta^T x^{(i)}) + (1-y^{(i)})\text{cost}_0(\theta^T x^{(i)}) + \frac{1}{2}\sum_{i=0}^{n}\theta_j^2$$

$\min_u (u-5)^2+1 \rightarrow u=5$ ×10 $\rightarrow \min_u 10(u-5)^2+10 \rightarrow u=5$

$A + \lambda B \Leftarrow$ $C = \frac{1}{\lambda}$

$\rightarrow C A + B \Leftarrow$

$$\rightarrow \min_\theta C\sum_{i=1}^{m}\left[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$

- 첫번째로, cost 최적화에 영향이 없는 상수인 m을 제거하고,
- 두번째로, 새로운 cost 함수(hinge loss)를 이용하였으며,
- 세번째로, 새로운 매개변수인 C를 이용하여 A함수를 최적화하는 방식을 사용한다.
- C는 $\frac{1}{\lambda}$ 의 관계로 정의하고, regularization 단계에서 λ를 제거하였다.
- 결국 $A + \lambda B(\text{logistic regression}) = CA + B(\text{SVM})$은 동일한 결과를 제공한다.

왜 SVM에서는 최적화 함수를 CA + B로 변환했을까?

- C가 아주 큰 값이라고 가정해 보자. (C = 100,000) ➔ 엄격한 분류
- SVM에서 CA + B → 최소화 하려면 A가 0이되어야 한다.
- A가 0이 되기 위해서는 hinge loss의 값이 0이 되어야 하며,
- 즉 아래와 같은 조건이 필요하다

$$\min_\theta C\sum_{i=1}^{m}\left[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$

A

If $y=1$, we want $\theta^T x \geq 1$ (not just ≥ 0)

If $y=0$, we want $\theta^T x \leq -1$ (not just < 0)

C가 크다고 가정한 이유는. $C = \frac{1}{\lambda}$ 로 가정한 경우,
λ 의 값이 적다는 의미임.
즉, regularization에 영향이 적어서, 모델의 overfitting이 높아짐.

→ 결국 Training data에 대하여 오분류가 없도록 분류하게 됨.

- AC = 0으로 정의하면, 최적화 공식이 아래와 같이 단순해 진다.

$$\min_\theta \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

Θ만 최적화에 영향을 줌

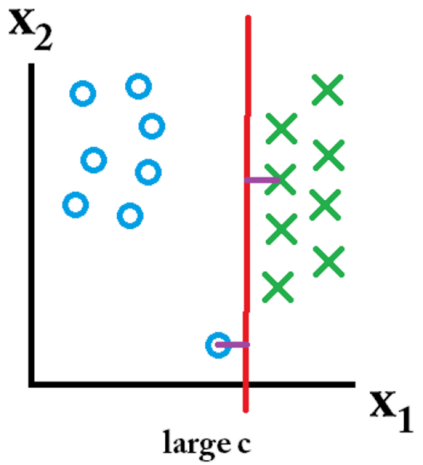
- 이제부터 SVM Decision Boundary를 찾기 위해서는
- Θ가 최소가 되는 선형을 찾으면 되도록 공식을 단순화 함.

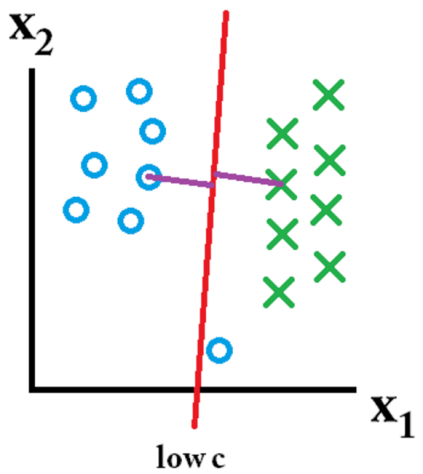
Week7. SVM – [참고] 파라미터 C가 SVM모델에 미치는 영향

C는 regularization 파라미터 λ와 연관되어, 모델학습시 overfitting을 조정함

C값이 decision boundary에 미치는 영향은?

$$\min_{\theta} C\sum_{i=1}^{m}\left[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$



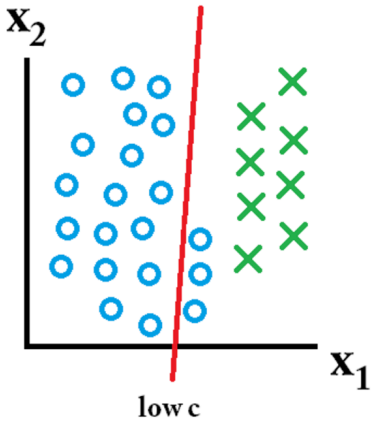


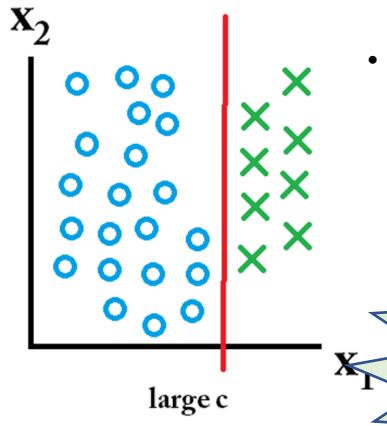
- C가 크면 상대적으로 Cost($\Theta^T x^{(i)}$)의 값이 작아져야 전체 cost가 최소화됨.
- 또한 $C = \frac{1}{\lambda}$ 이 관계에서 λ 가 작아진다는 의미임.

- 즉, λ가 작으면 back propagation 과정에서 가중치 Θ가 미치는 영향이 커지고, 학습데이터에 최적화 → overfitting 확률이 높아짐.
- 결국 모든 데이터가 오류없이 분류되도록 Decision boundary 생성됨

그럼 어떤 C값을 선택해야 할까?

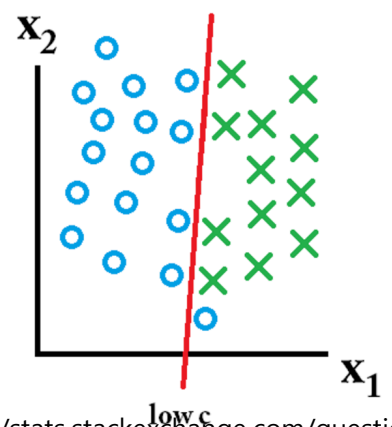
- 결국 실제 데이터 or Test Data를 얼마나 잘 예측하느냐가 중요함.
- 이 관점에서 보면 왼쪽의 decision boundary에서 어떤 데이터가 앞으로 입력되는지에 따라 C값이 평가됨.

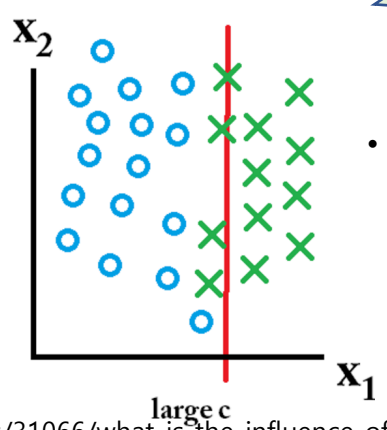




- 이 경우에는 large C로 학습한 모델이 뛰어남.

Cross validation을 통해 결정





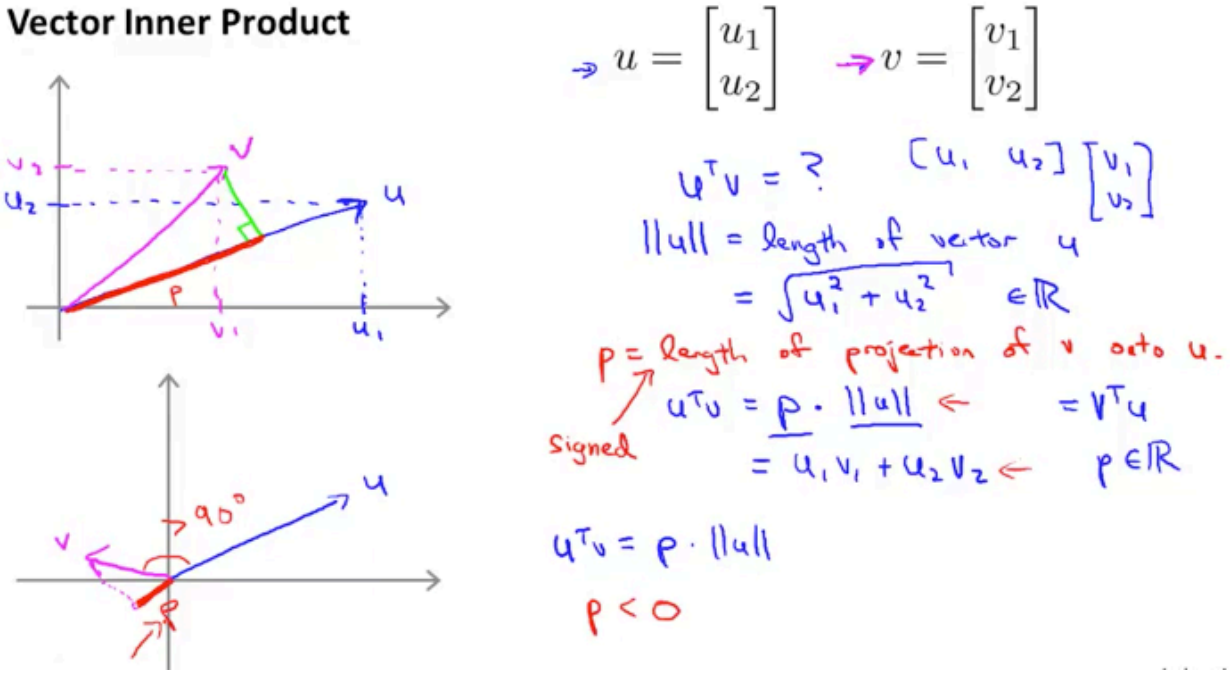
- 이 경우에는 low C로 학습한 모델이 더 좋음.

https://stats.stackexchange.com/questions/31066/what-is-the-influence-of-c-in-svms-with-linear-kernel

Week7. SVM – Margin은 어떻게 최대화 할까? (벡터 내적 이해)

Vector 내적을 이용하여 margin을 최대화(cost 최소화) 하는 ||Θ||를 계산

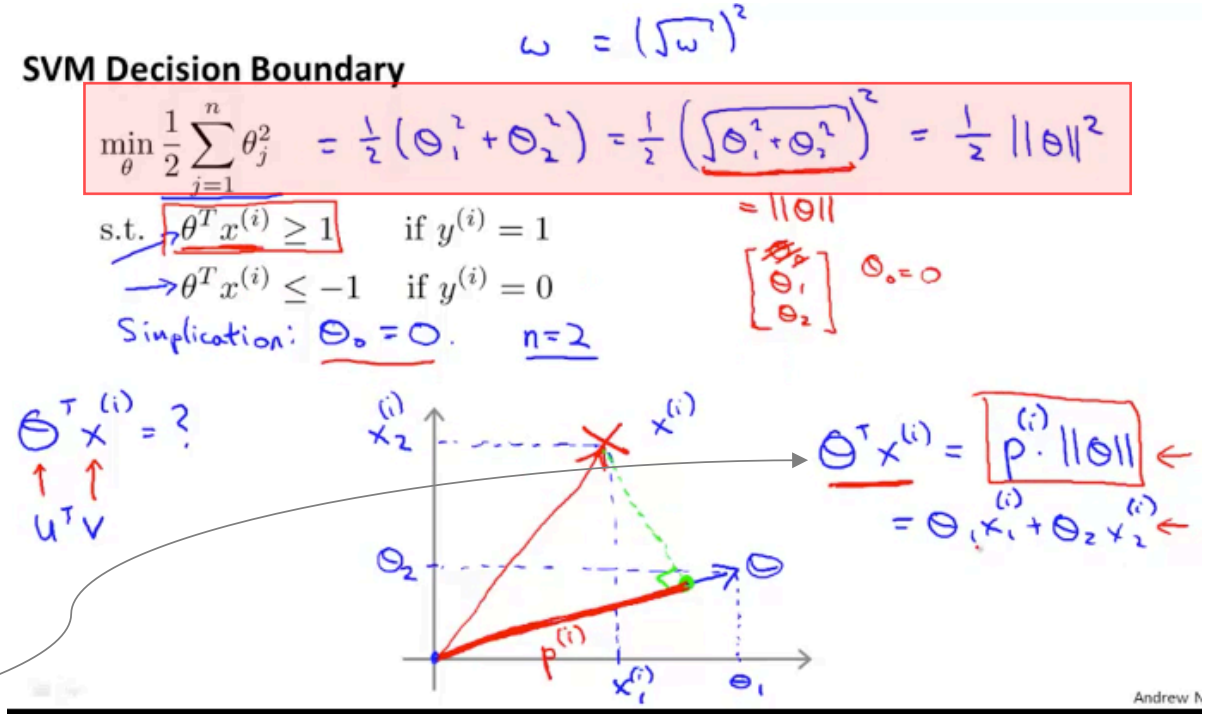
Vector 내적의 이해



- P는 V를 90도로 U에 투영한 선을 나타낸다.
- 이를 공식으로 표현하면, $U^T V = P \cdot ||U||$
- 이때 90도 이상인 P는 음수를 가지게 된다.

- 결국 벡터의 내적은 2개 벡터간의 각도에 따라서 양수/음수가 결정된다.

SVM의 cost 함수에서 Θ 를 계산하는 용도로 활용



- 공식을 변형하여 ||Θ|| (Θ 의 거리, $\sqrt{\Theta 1^2 + \Theta 2^2}$)를 활용

- Θ 값을 U로 정의하면, 위와 같은 그래프가 그려지고
- V를 90도로 투영하면 P가 계산될 수 있다.

- 그리고 $\Theta^T x^{(i)} = P^{(i)} ||\Theta||$ 로 대체가능하게 된다.

Week7. SVM – Decision Boundary를 선택하는 과정은?

Θ가 최소가 되는 선형을 선택

SVM Decision Boundary

$$\min_\theta \frac{1}{2}\sum_{j=1}^{n}\theta_j^2 = \frac{1}{2}\|\theta\|^2$$

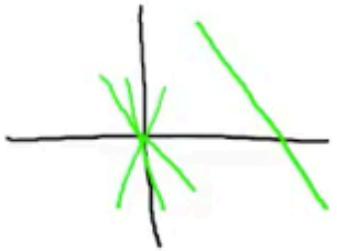
s.t. $p^{(i)} \cdot \|\theta\| \geq 1$ if $y^{(i)} = 1$

$p^{(i)} \cdot \|\theta\| \leq -1$ if $y^{(i)} = 1$

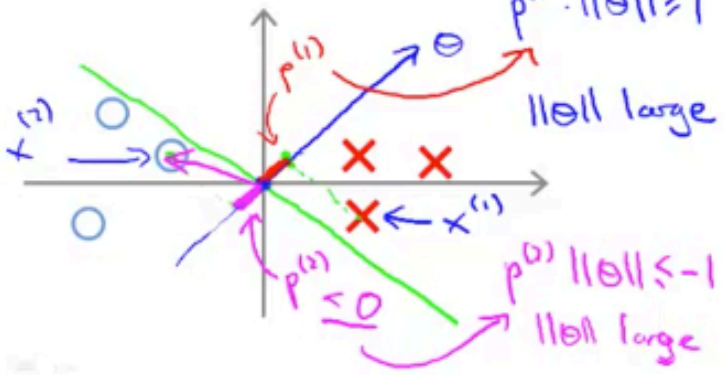
C very large

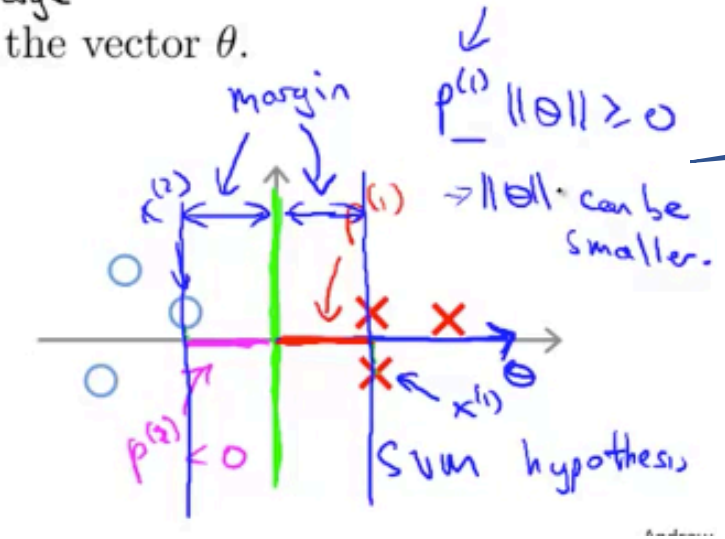
where $p^{(i)}$ is the projection of $x^{(i)}$ onto the vector θ.

Simplification: $\theta_0 = 0$



Θ0가 0이라는 의미는 Θ의 직선이 (0,0)을 지남을 의미





P가

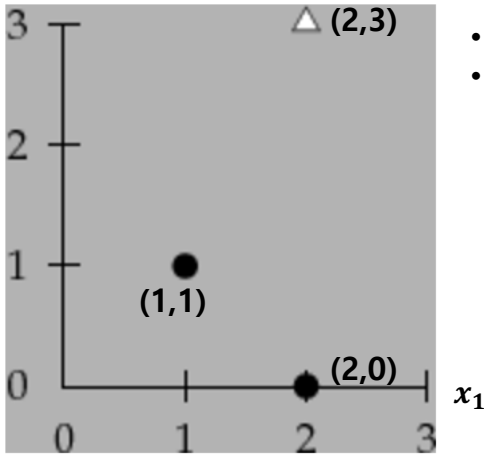
- [왼쪽 그래프]
- 녹색 Decision boundaryrk가 주어졌을때, 직교하는 Θ에, $X^{(1)}$이 투영된 거리인 P가 너무 작다.
- P가 작다는 의미는, ||Θ||가 아주 크다는 의미이므로 Cost Function의 값이 커지는 문제가 발생한다 (cost function = $\frac{1}{2}||\Theta||^2$)

- [오른쪽 그래프]
- Θ의 선이 X축의 수령으로 주어졌을 때, $X^{(1)}$이 투영된 거리인 P가 크가.
- P가 크다는 의미는, ||Θ||가 작아진다는 의미이므로 Cost 가 줄어들게 된다. (최적)

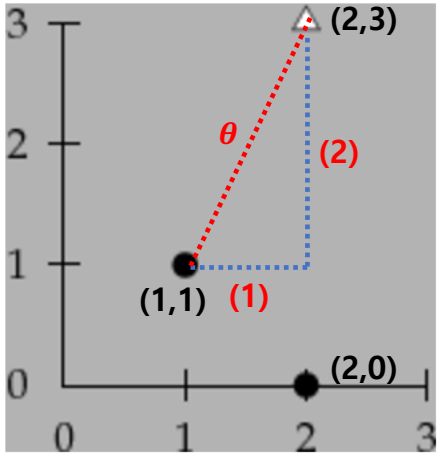
Week7. SVM – 연습문제 풀이 (Decision Boundary를 직관적으로 계산)

3개의 point를 2개의 class로 분류해 보자.

Weight Vector를 찾아보자



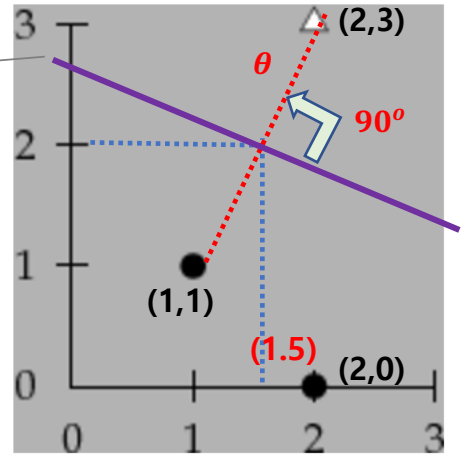
- Class 1 : (2,3) 1개
- Class 2 : (1,1), (2,0) 2개



- Weight vector는 어떻게 찾을 수 있을까?
  - 2 class간에 가까운 좌표 (1,1) 과 (2,3) 사이에 존재 함
  - 즉 (1, 2)값을 가지게 됨 ➔ 즉, 기울기 2/1을 가지는 선형임.
  - $\vec{\theta}$ = (1, 2)

직관적(geometrically)으로 계산한 decision boundary는?

- Decision boundary의 조건은 θ 에 직교해야 한다.
- 또한 각 support vector (2,3) (1,1)의 중간에 위치해야 한다.
- 따라서 decision boundary는 2 좌표의 중간지점인 (1.5, 2)를 지나야 함
  - 1 + (2-1)/2 = 1.5
  - 1 + (3-1)/2 = 2
- 이렇게 도출된 W와 좌표(1.5, 2)를 기반으로 계산해 보면
  - $x_1 + 2x_2 + b = 0$ 가 도출되고,
  - 여기서 b는 θ(1,2)와 X(1.5, 2)를 대입하면 계산가능함.
  - $1.5 + 2 * 2 + b = 0$ ➔ $b = -5.5$
- 최종 Decision Boundary
  - $x_1 + 2x_2 - 5.5 = 0$

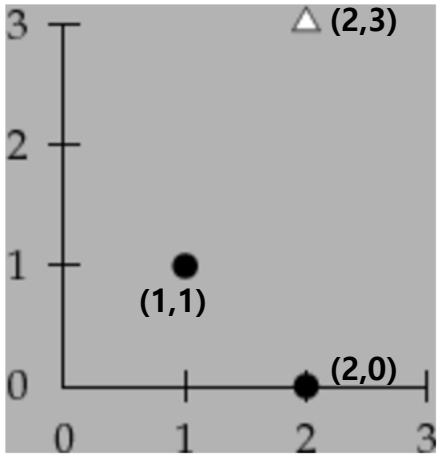


https://nlp.stanford.edu/IR-book/html/htmledition/support-vector-machines-the-linearly-separable-case-1.html

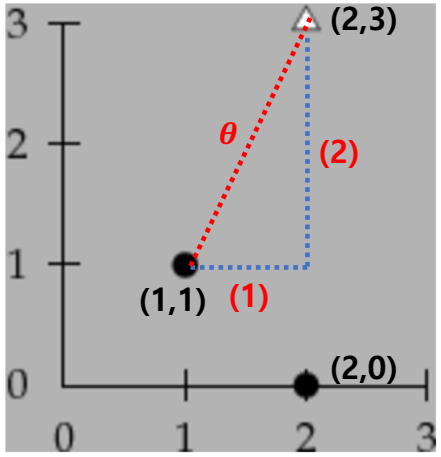
Week7. SVM –연습문제 풀이 (Decision Boundary를 대수학으로 계산)

3개의 point를 2개의 class로 분류해 보자.

Weight Vector를 찾아보자



- Class 1 : (2,3) 1개
- Class 2 : (1,1), (2,0) 2개

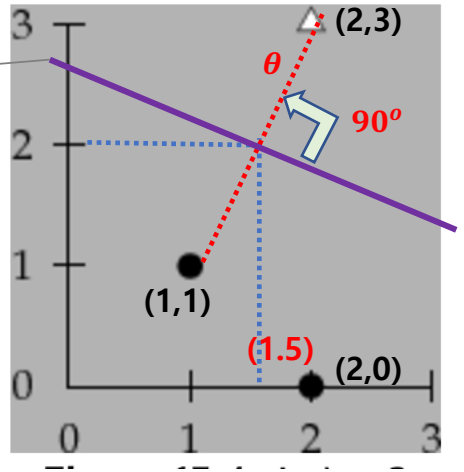


- Weight vector는 어떻게 찾을 수 있을까?
  - 2 class간에 가까운 좌표 (1,1) 과 (2,3) 사이에 존재 함
  - 즉 (1, 2)값을 가지게 됨 ➔ 즉, 기울기 2/1을 가지는 선형임.
  - $\vec{\theta}$ = (1, 2)

대수학(algebraically)으로 계산한 decision boundary는?

$$\min_{\theta} \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$
$$\text{s.t.} \quad \theta^T x^{(i)} \geq 1 \quad \text{if } y^{(i)} = 1$$
$$\theta^T x^{(i)} \leq -1 \quad \text{if } y^{(i)} = 0$$

- Cost 를 최소화 하려면, θ 를 작게 해야 함.
- $(\theta^T x^i) = p^i * ||\theta||$
  $= \theta_1 x^i_1 + \theta_2 x^i_2$
- 이미 $\vec{\theta} = (a, 2a)$ 를 알고 있으므로,
- $x_1 + 2x_2 + b = 0$
- $a + 2a = -1$ ➔ Class 2 (1,1) 대입
- $2a + 6a = 1$ ➔ Class 1 (2,3) 대입
- $a = \frac{2}{5}, \quad b = -\frac{11}{5}$
- 따라서 최적의 decision boundary는
- $\vec{\theta} = \left(\frac{2}{5}, \frac{4}{5}\right)$ and $b = -\frac{11}{5}$ 를 가짐
- 이를 대입해 보면
- $x_1 + 2x_2 - 5.5 = 0$
- Margin 은 $\frac{2}{\|W\|}$ 이므로,
- $\frac{2}{\sqrt{\frac{4}{25}+\frac{16}{25}}} = \frac{2}{2\sqrt{\frac{5}{5}}} = \sqrt{5}$



https://nlp.stanford.edu/IR-book/html/htmledition/support-vector-machines-the-linearly-separable-case-1.html

Week 7-2

Kernel

Week7-2. Kernels

선형분류가 어려운 저차원 데이터를 고차원 공간으로 매핑하는 함수

저차원 → 고차원

원래 공간 — 매핑 함수 — 매핑 결과

$\mathbf{a} = (0,0)^T,\ t_a = 1$

$\mathbf{b} = (1,0)^T,\ t_b = -1$

$\mathbf{c} = (0,1)^T,\ t_c = -1$

$\mathbf{d} = (1,1)^T,\ t_d = 1$

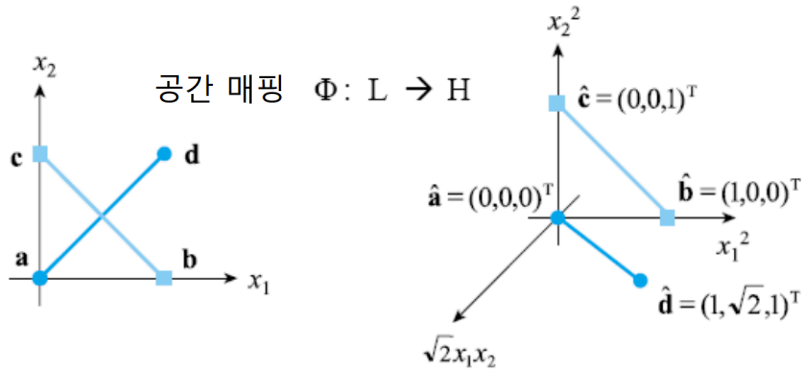
$$\Phi_1(\mathbf{x}) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$

$\mathbf{a} = (0,0)^T \rightarrow \hat{\mathbf{a}} = (0,0,0)^T$

$\mathbf{b} = (1,0)^T \rightarrow \hat{\mathbf{b}} = (1,0,0)^T$

$\mathbf{c} = (0,1)^T \rightarrow \hat{\mathbf{c}} = (0,0,1)^T$

$\mathbf{d} = (1,1)^T \rightarrow \hat{\mathbf{d}} = (1,\sqrt{2},1)^T$



(a) 원래 공간 L (b) 매핑된 공간 H

커널 함수 $K(\mathbf{x},\mathbf{y}) = (\mathbf{x}\cdot\mathbf{y})^2 = x_1^2y_1^2 + 2x_1y_1x_2y_2 + x_2^2y_2^2$

Kernel function for SVMs & Data binarization pdf

http://gentlej90.tistory.com/43

고차원 변환을 위한 계산비용 증가

- 왼쪽의 예시와 같이 2차원(L) → 3차원(H)으로 변환한 후에 SVM과 같은 분류 모델로 학습(계산)하는 것은 성능상의 문제가 없다.
- 하지만, 만약 차원이 아주 큰 고차원으로 변환한다면, 고차원 데이터(벡터)를 SVM으로 연산(내적)하는 것은 너무 많은 연산비용이 소모되어 사용이 어렵다.

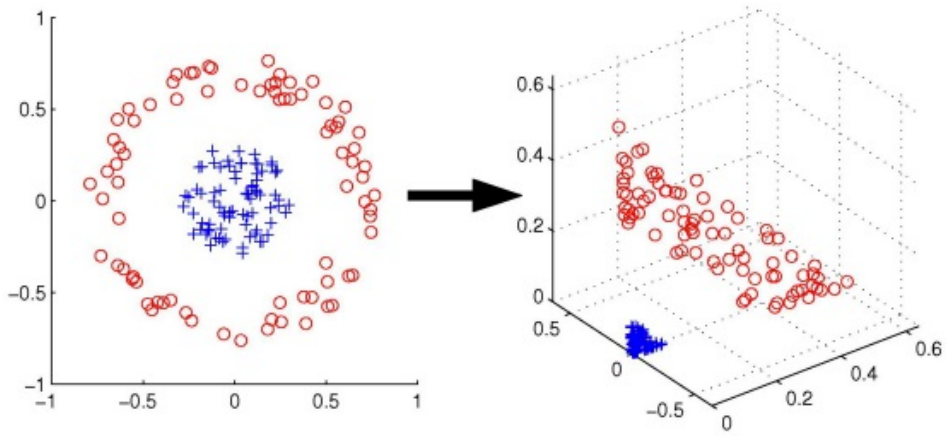
커널 트릭 (Kernel Trick) 활용

- 수학적으로 고차원 데이터인 $a^{\wedge}, b^{\wedge}$를 내적하는 것과,
- 내적한 결과를 고차원으로 보내는 것은 동일하다.
- 결국 $\mathrm{K}(x_{i,}, y_{i,}) = \phi(x_{i,}) \cdot \phi(y_{i,})$ 와 동일하다.
- 따라서 알고리즘의 수식 중 $\phi(x_{i,}) \cdot \phi(y_{i,})$ 이 있는 것은 모두 $\mathrm{K}(x_{i,}, y_{i,})$ 로 대체 가능

Week7-2. Kernels

선형분류가 어려운 저차원 데이터를 고차원 공간으로 매핑하는 함수

Kernel함수의 예시



- **선형으로 풀수 없는 문제를 커널함수를 이용해서 고차원의 데이터로 변환하여 계산함.**

Kernel함수의 종류

Polynomial kernel:

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

Gaussian(Radial-Basis Function (RBF)) kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2}$$

Sigmoid:

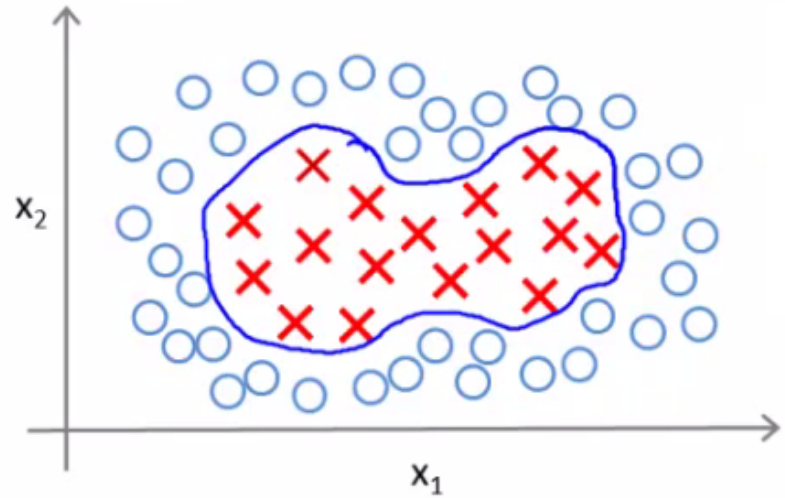
$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\alpha \mathbf{x}_i \cdot \mathbf{x}_j + \beta)$$

문제에 따라 최적의 커널함수를 직접 학습& 테스트하여 찾아야함.

Week7-2. Kernels – 좀 더 자세히 이해해 보자.

비선형 decision boundary를 찾기 위해서 커널을 어떻게 이용할까?

비선형 Decision boundary를 찾자



- 위 학습 데이터를 잘 분류할 복잡한 다항 feature를 생각해 보자.

왜 이런 전제를 가질까?

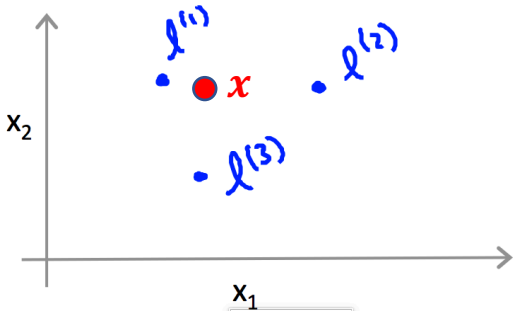
- $h_\theta(x)$ (아래의 전제를 가짐)
  - Return 1 : 전체 벡터(θ 에 의해 weighted된)의 합이 >= 0
  - Return 0 : else

f_0는 항상 1로 설정, θ_0 값을 그대로 반영

- 위 공식을 다르게 표현하면
- 기존 X를 다양한 고차원 X항으로 표현한 새로운 feature f에 의해 곱해진 θ의 합을 취하는 decision boundary를 계산한다.
  - $h_\theta(x) = \theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3$
  - $f_1 = x_1$
  - $f_2 = x_1 x_2$, $f_3 = x^2$ …

고차원 다항식보다 좋은 feature f 가 있을까?
→ 고차원 다항식은 계산시간이 오래걸림

Kernel 함수를 정의하자



- 3개의 feature를 정의하고($x_0 = 0$는 무시)이를 2개 차원으로 표시하자. (x_1, x_2)
- 그리고 2차원 공간에 3개의 임의 점을 선택한다. (l^1, l^2, l^3) → landmark

- 새로운 X가 주어졌을 때, X와 l^1 간의 유사도(simility)를 f_1으로 정의
  - $f_1 = \exp\left(-\frac{\|x-l^{(1)}\|^2}{2\sigma^2}\right)$
  - $\|x - l^1\|$ = euclidean 거리 $\|x-l^{(1)}\|^2 = \sum_{j=1}^{n}(x_j - l_j^{(1)})^2$
  - σ = 표준편차
  - σ^2 = 분산

- 나머지 f 도 아래와 같이 정의해 보자
  - $f_2 = simility(x, l^2) = \exp(-\frac{\|x-l^2\|^2}{2\sigma^2})$
  - $f_3 = simility(x, l^3) = \exp(-\frac{\|x-l^3\|^2}{2\sigma^2})$

Kernel

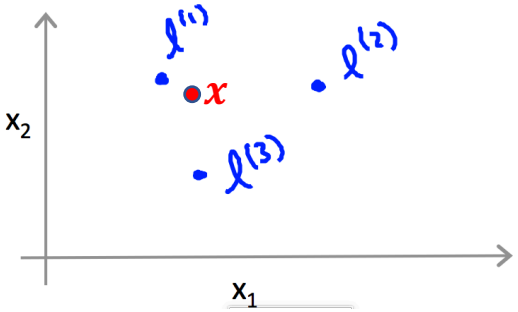
Gausian Kernel

$f_1 = K(x, l^3)$

Week7-2. Kernels – 좀 더 자세히 이해해 보자.

정의한 커널이 무슨 일을 할까?

X가 landmark(l)의 거리에 따른 결과값 계산



거리가 아주 가까운 경우

$x \approx l^1$

$$f_1 \approx \exp\left(-\frac{\|x - l^1\|^2}{2\sigma^2}\right)$$
$$\approx \exp\left(-\frac{0}{2\sigma^2}\right)$$
$$\approx 1$$

거리 0

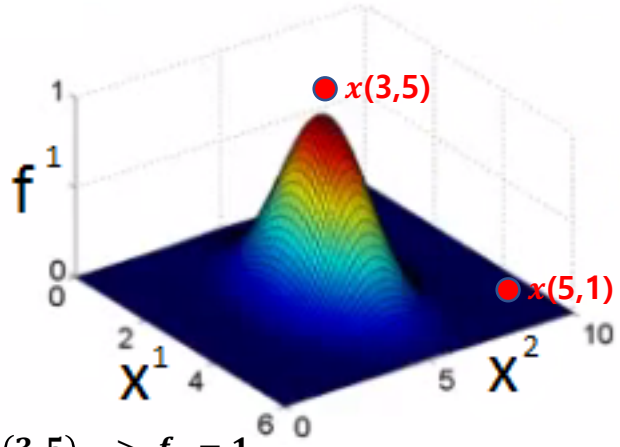
거리가 아주 먼 경우

$x \approx l^1$

$$f_1 \approx \exp\left(-\frac{\|x - l^1\|^2}{2\sigma^2}\right)$$
$$\approx \exp\left(-\frac{Big\ Number}{2\sigma^2}\right)$$
$$\approx 0$$

커널함수와 f_1의 관계를 그래프로 시각화 해보자

- 아래와 같은 값을 가질 경우
- $X = (3,5)\ and\ f_1 = 1$
  - 만약 X가 (3,5) 포인트로 이동하면 → $f_1 = 1$이 된다.
  - 만약 X가 (3,5) 포인트에서 멀어지면 → $f_1 = 0$에 가깝게 된다
- 따라서 X가 l (landmark)과 얼마나 근접한지 계산할 수 있게된다.



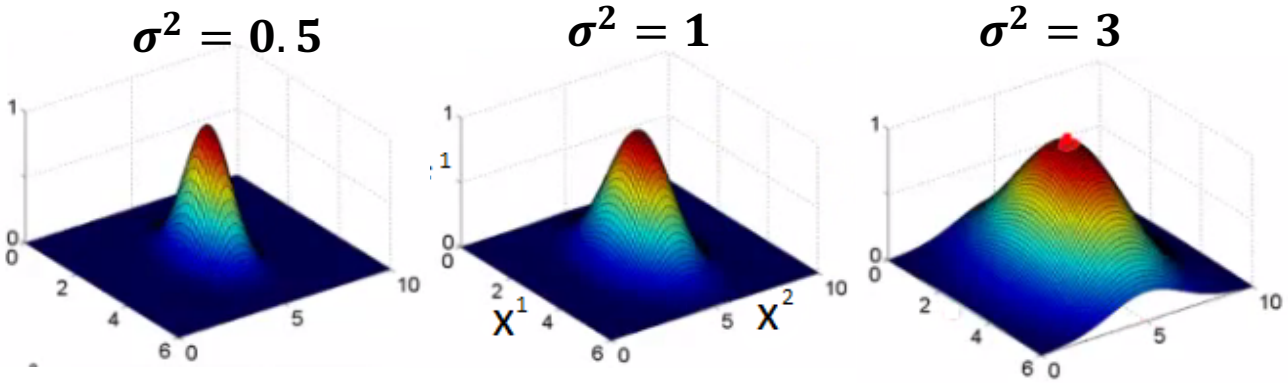
- $X = (3,5) \rightarrow f_1 = 1$
- $X = (5,1) \rightarrow f_1 = 0.1..$
- 그런데 여기서 σ^2은 어떤 역할을 하게 되는걸까?

Week7-2. Kernels – σ^2 은 어떤 역할을 하게 되나?

$Gausian\ Kernel$의 파라미터로 landmark 주변의 기울기를 정의한다.

σ^2 의 변화에 따른 기울기 시각화

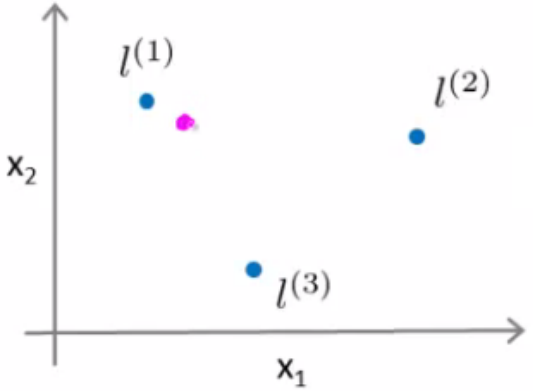
- σ^2 이 0과 가까워 지면, 기울기(경사)가 커지면서 폭이 좁아진다.



- 이렇게 정의된 조건을 이용하여, 어떤 가설을 배울 수 있을까?
- 만약 학습 샘플 X가 아래의 조건을 만족하면, 정답 "1"을 예측한다.
- → $Return\ 1\ when\ \ \theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$

예를 들어서 어떻게 정답을 예측하는지 보자

- 예시를 위해, 이미 알고리즘을 실행한 결과로 아래의 값을 가진다고 가정
  - $\theta_0 = -0.5$
  - $\theta_1 = 1$
  - $\theta_2 = 1$
  - $\theta_3 = 0$
- 아래와 같이 3개의 샘플이 존재 할때, 새로운 X는 어떤 값을 예측할까?



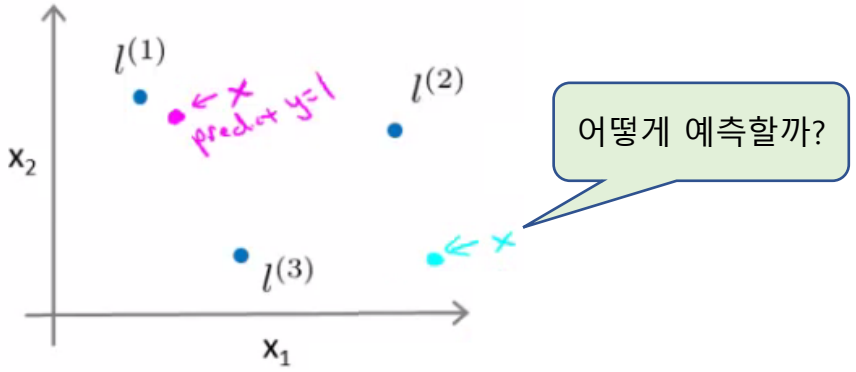
- $\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$
- $= -0.5 + 1 * 1 + 1 * 0 + 0 * 0 = 0.5 \geq 0$ ➔ 1을 예측
- 왜 $f_1 = 0, f_2 \& f_3 = 0$ 일까?
  - → X와 가장 가까운 f_1은 왼쪽 그림과 같이 1
  - →$f_2 \& f_3$는 거리가 멀기 때문에 0

Week7-2. Kernels – σ^2 은 어떤 역할을 하게 되나?

최종적으로 σ^2를 사용한 Kernel함수를 통해 비선형 boundary를 분류 가능

만약 3개의 샘플에서 멀리 떨어진 X가 있다면?

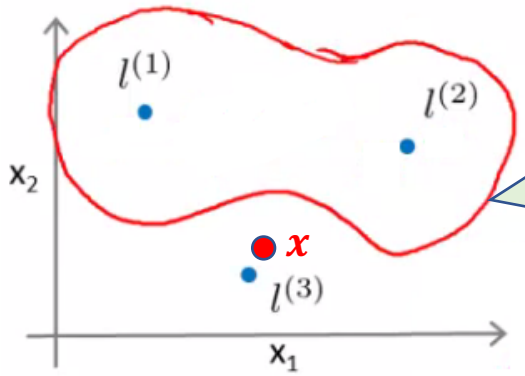
- $landmark$와의 거리가 멀면 어떻게 예측할까?
- 예시를 위해, 이미 알고리즘을 실행한 결과로 아래의 값을 가진다고 가정
  - $\theta_0 = -0.5$
  - $\theta_1 = 1$
  - $\theta_2 = 1$
  - $\theta_3 = 0$



- $\theta_0+\theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$
- $= -0.5 + 1 * 0 + 1 * 0 + 0 * 0 = -0.5 < 0$ ➔ 0을 예측
- 왜 $f_1 = 0, f_2 \& f_3 = 0$ 일까?
  - ➔ X와 가장 가까운 f_1은 왼쪽 그림과 같이 1
  - ➔$f_2 \& f_3$는 거리가 멀기 때문에 0

그럼 l^3 근처에 X가 있다면?

- 예시를 위해, 이미 알고리즘을 실행한 결과로 아래의 값을 가진다고 가정
  - $\theta_0 = -0.5$
  - $\theta_1 = 1$
  - $\theta_2 = 1$
  - $\theta_3 = 0$
- 파라미터 값($\theta_1, \theta_2, \theta_3$)이 서로 다르게 정의되어 있으므로, 이 값에 따라 예측이 다르게 됨.
  - $\theta_1, \theta_2 = 1$ ➔ f_1, f_2 값이 분류에 영향을 미침
  - $\theta_3 = 0$ ➔ f_3 값이 영향을 주지 못함.



- 빨강 선 안에 X가 있으면 1 예측
- 빨강 선 밖에 X가 있으면 0 예측
- ➔ 비선형 Boundary 분류 가능!!

- $\theta_0+\theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$
- $= -0.5 + 1 * 0 + 1 * 0 + 0 * 1 = -0.5 < 0$ ➔ 0을 예측

Week7-2. Kernels – 그런데 landmark는 어떻게 선택하지?

복잡한 문제의 경우, 많은 landmark가 필요할 수 있다.

Landmark는 어떻게 선택하고, f를 계산할까?

1. 학습 데이터를 읽어온다. (100건)
2. 학습 데이터 별로 landmark를 생성한다. (학습데이터와 동일한 위치)
   - 학습데이터 1건 별로 1개의 landmark가 생성됨
   - 최종 100개의 landmark 생성
3. 새로운 샘플이 주어지면, 모든 landmark와의 거리(f)를 계산한다.
   - $f_0 \sim f_{99}$, 총 100개의 f 결과
   - $f_0 = 1$ (θ_0는 bias 값이므로, 값을 그대로 유지)
   - 자세히 계산과정을 보면
     - $f_1^i, = K(x^i, l^1)$
     - $f_2^i, = K(x^i, l^2)$
     - ....
     - $f_{99}^i, = K(x^i, l^2)$
   - 위 과정을 반복하면, X 자신과 동일한 landmark와 비교하는 구간이 있다. → 이 경우 Gaussian Kernel에서는 1로 평가 (동일한 위치)

X값이 vector(배열)로 구성된 경우

- 이렇게 계산된 f 값을 [m(99)+1 x 1] 차원의 vector로 저장한다.
- f^i → f 벡터의 i 번째 데이터를 의미

결정된 f값을 SVM에서 어떻게 활용하는가?

1. 이전에 정의했던 수식을 확인해보자
   - $Predict\ y = 1$
   - $when\ \ \theta^T f \geq 0$
   - $And\ f = [m+1 * 1]$
2. 그럼 θ는 어떻게 계산할 수 있을까?
   - SVM Optimization 알고리즘 이용

$$\min_\theta C\sum_{i=1}^{m}\left[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})\right] + \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$

   - 위 최적화 결과를 최소화하기 위해 f 벡터를 이용한다.
   - 그리고 이 최적화 알고리즘을 계산하면, θ를 찾을 수 있게 된다.
3. 계산 성능 향상을 위한 Tip
   - 위 예시에서 m = n으로 가정 (학습 데이터와 f 벡터의 수가 같기 때문) $\sum_{j=1}^{n}\theta_j^2 = \theta^T\theta$ ⇒ $\theta^T M \theta$
   - 좌측방식 구현보다는 우측 방식으로 구현하는 것이 계산성능 향상
   - 데이터가 많을 경우 수많은 for loop를 하지 않고, 매트릭 계산

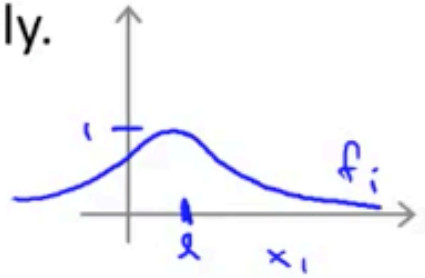
Week7-2. SVM Parameter

SVM 파라미터가 미치는 영향

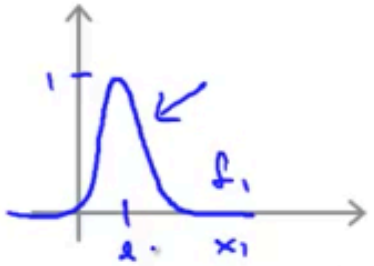
C ($= \frac{1}{\lambda}$). → Large C: Lower bias, high variance. (small λ)
→ Small C: Higher bias, low variance. (large λ)

σ^2 Large σ^2: Features f_i vary more smoothly.
→ Higher bias, lower variance.

$\exp\left(-\frac{\|x-l^{(i)}\|^2}{2\sigma^2}\right)$



Small σ^2: Features f_i vary less smoothly.
Lower bias, higher variance.



C

- **C ↑, λ ↓ : Lower Bias, High Variance ➔ Overfitting**
- **C ↓, λ ↑ : High Bias, Lower Variance ➔ Underfitting**

σ^2

- **σ^2 ↑ → f_i부드러운 곡선 → High Bias, Lower Variance ➔ Underfitting**
- **σ^2 ↓ → f_i부드러운 곡선 → Lower Bias, High Variance ➔ Overfitting**

Week7-2. Using an SVM – 커널 선택

SVM의 커널을 선택할때...

Kernel을 선택하지 않으면? → Linear Kernel

1. 기본 linear classifier로 동작한다.
   - $Predict\ y = 1\ when\ \ \theta^T f \geq 0$
2. 어떤 경우에 사용할까?
   - Feature(n) → 크고
   - Training Data (m) → 작은 경우
   - 고차원 feature인 경우 Overfitting의 위험이 있다.

Gausian Kernel

1. 커널에 필요한 작업들
   - σ^2 값을 정해야 한다.
   - Kernel 함수 정의 : $f = K(x_1, x_2)$
2. 어떤 경우에 사용할까?
   - Feature(n) → 작고
   - Training Data (m) → 큰 경우 (예를 들면 2차원 데이터 학습)
3. 대부분의 경우 좋은 결과를 도출함 (가장 많이 활용하는 kernel)

다른 커널들...

1. Polynomial Kernel
   - 입력 데이터 x와 landmark l간의 similarity를 아래 함수로 계산
     - (xT l)2
     - (xT l)3
     - (xT l+1)3
   - 공식은 : $(x^T l+Con)^D$
   - 파라미터 : Degree of Polynomial(D), L에 더할 상수값(Con)
   - 음수를 가지는 데이터가 없는 경우에 사용
2. String Kernel, Chi-square Kernel ...
   - 소수의 문제해결에 사용하며, 많이 활용되지는 않는다.

Week7-2. SVM 참고자료

- A User's Guide to Support Vector Machines : http://www.cs.colostate.edu/~asa/pdfs/howto.pdf
- https://www.cise.ufl.edu/class/cis4930sp11dtm/notes/intro_svm_new.pdf
- https://nlp.stanford.edu/IR-book/html/htmledition/support-vector-machines-the-linearly-separable-case-1.html