9. Support Vector Machines

서포트 벡터 머신(SVM): 뒤에 나올 9.1의 maximal margin classifier(최대마진분류기)라고 불리는 직관적인 분류기를 일반화 한 것임. 클래스들이 선형 경계에 구별될 때만 사용가능

서포트벡터분류기 : 최대마진분류기를 확장한 것

9.1 최대 마진 분류기

9.1.1 우선 초평면(hyperplane)이란?

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

- p차원 공간에서의 초평면은 p-1차원인 평평한 부분공간. 초평면은 여러개일 수 있으며 초평면의 식은 다음과 같다

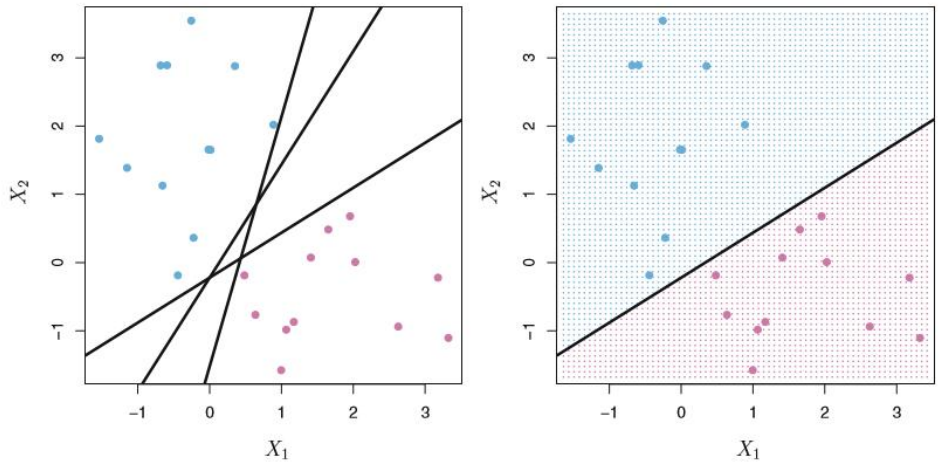
- 위 식을 만족하는 X값들의 벡터가 초평면 상의 점이 되며 좌변은 초평면의 함수로서 그 값이 0보다 큰지, 작은지로 p차원 공간을 분할할 수 있다.

9.1.2 분리 초평면(seperating hyperlane)

-분리 초평면을 사용해서 분류를 할 수도 있다. train데이터를 속하는 클래스에 따라 완벽하게 분리하는 초평면을 구성할 수 있다고 생각해보자. 클래스의 라벨은 -1과 1 두가지만을 가지며 분리초평면의 식과 어떤 데이터를 사용해 예제를 만들었을 때 분리 초평면을 이용한 분리 시각화 결과는 다음과 같다.

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} > 0 \text{ if } y_i = 1,$$

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} < 0 \text{ if } y_i = -1.$$



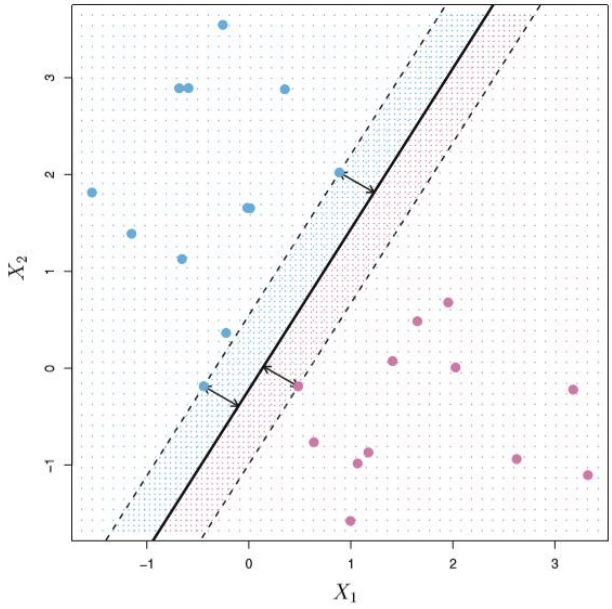
세가지 선 중 데이터를 가장 잘 구분하는 초평면은 오른쪽 그래프에 남은 직선이다. 즉 분리 초평면에 기초한 분류기는 선형결정경계로 이어진다.

9.1.3 최대 마진 분류기

- 위의 예시에서 봤듯이 분리 초평면을 그릴 수 있으면 그것을 아주 미세하기 움직이거나 각도를 아주 미세하기 움직임으로써 그러한 분리 초평면을 무수히 많이 존재한다는 것을 알 수 있다. 그렇다면 그 무수히 많은 것들 중 어떤 것을 선택해야 할까?

- 정답은 train데이터들로부터 가장 멀리 떨어진 분리 초평면을 고르는 것이다. 이러한 초평면을 최대 마진 초평면이라 한다. 각각의 train 데이터에서 초평면까지의 거리를 계산할 수 있으며, 데이터들 중 어떠한 분리 초평면에 대해 수직 거리가 가장 짧게 되는 데이터가 존재할 것이고 그 가장 짧은 거리를 margin이라고 한다.

- 최적의 마진 초평면은 분리 초평면 중 마진이 가장 큰 것이다. train데이터에 큰 마진을 가지는 분류기가 test데이터에 대해서도 큰 마진을 가질 것이라고 생각할 수 있다.



최대 마진 초평면은 위 예시에서 점선이며 화살표로 연결된 파란색 2개점과 보라색 1개점을 서포트 벡터라고 한다. 최대 마진 초평면은 두 클래스 사이에 끼울 수 있는 가장 넓은 평판(slab)의 중간선을 의미한다고 할 수 있다. 위 예시에서 세 화살표의 길이는 모두 같다.

- 서포트 벡터로 명명한 이유는 우선 이들 세 점은 p차원(여기서는 2)차원 공간의 좌표를 가지는 벡터이며, 이 점들이 이동하면 최대 마진 초평면도 이동하기 때문에 최대 마진 초평면을 support한다고 생각할 수 있다.
- 최대 마진 초평면은 서포트벡터에는 직접적으로 의존하지만 다른 벡터들에는 (그 데이터가 마진으로 인해 설정된 경계를 넘어가지 않으면) 의존적이지 않다.

9.1.4 최대 마진 분류기의 구성

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,M}{\text{maximize}} \; M \tag{9.9}$$

$$\text{subject to} \; \sum_{j=1}^{p} \beta_j^2 = 1, \tag{9.10}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M \;\; \forall\, i = 1, \ldots, n. \tag{9.11}$$

- n개의 train 데이터들(x들)과 클래스 라벨(y값들 = 1또는 -1)에 대해위의 최적화 문제의 해가 최대 마진 초평면의 해이다.

- (9.11)의 조건은 M이 양수이면 각 관측치가 초평면에 어디에 있게 되는지를 정해준다. 사실 M이 아니라 0이기만 해도 위치에 대한 정의가 가능하지만 양수로 지정해서 완충공간(cushion)까지 최대마진초평면에 반영되게 한다.

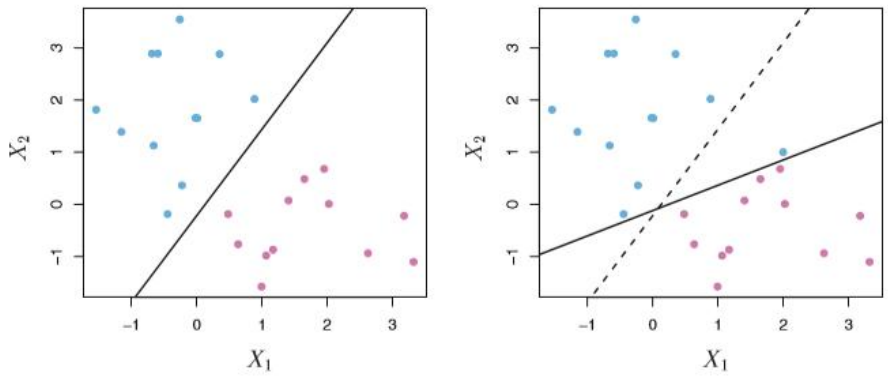
- (9.10)조건을 정의함으로써 (9.11)의 좌변이 i번째 관측치에서 초평면까지의 수직 거리가 될 수 있다.

- (9.9)에서 M은 초평면의 마진을 의미하고 그러한 M을 최대로 하는 파라미터 베타값들을 정해야 한다.

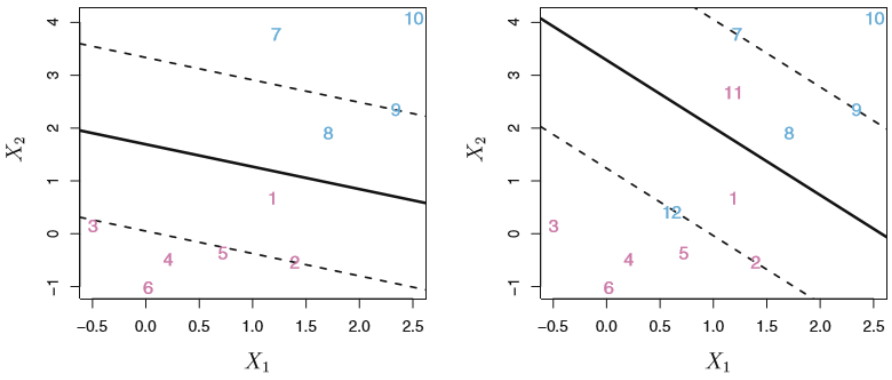
※ 하지만 분리 초평면을 정의할 수 없으면 최대 마진 분류기를 사용할 수 없다. 분리 초평면의 개념을 확장하고 보완한 것이 서포트 벡터 분류기이다.

9.2 서포트 벡터 분류기

- 분리 초평면을 이용하여 분류를 하면 여러 문제가 있을 수 있다. 분리 초평면 자체가 정의가 되지 않거나 정의가 되더라도 아웃라이어가 소량 유입되었을 때 최대 마진 분류기가 급격하게 변하는 (오버피팅에 취약함) 등의 문제가 발생할 수 있다.



- 각각의 데이터에 대해 더 robust하고 몇몇 train 데이터는 잘못 분류하더라도 나머지의 대부분을 잘 분류할 수 있는 분류기를 서포트 벡터 분류기라 하며 소프트 마진 분류기라고 하기도 한다.
- 서포트 벡터 분류기에서는 일부 데이터가 분류 될 때 마진을 침범할 수도 있고 초평면을 기준으로 옳지 않은 방향에 있을 수도 있다.



서포트벡터를 학습한 결과는 위와 같을 수 있다. 최대 마진 분류기처럼 엄격하지 않음을 보자. 오른쪽 그래프는 왼쪽 데이터에 11,12번데이터 두 개를 추가한 것이다.

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n,M}{\text{maximize}} M \tag{9.12}$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1, \tag{9.13}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1 - \epsilon_i), \tag{9.14}$$

$$\epsilon_i \geq 0, \ \sum_{i=1}^{n} \epsilon_i \leq C, \tag{9.15}$$

- 위 식의 M(마진의 폭)을 최대화하는 파라미터들을 찾는 것이 서포트 벡터 분류기의 목표이다. 엡실론값들은 개별 데이터들이 마진을 침범하거나 초평면의 옳지 않은 쪽에 있게 허용하는 슬랙 변수(slack variable)이다. 이전 페이지의 식과 마찬가지로 (9.14)의 좌변의 부호를 판별해 test데이터가 어느 위치에 놓이게 되는지 분류할 수 있다.

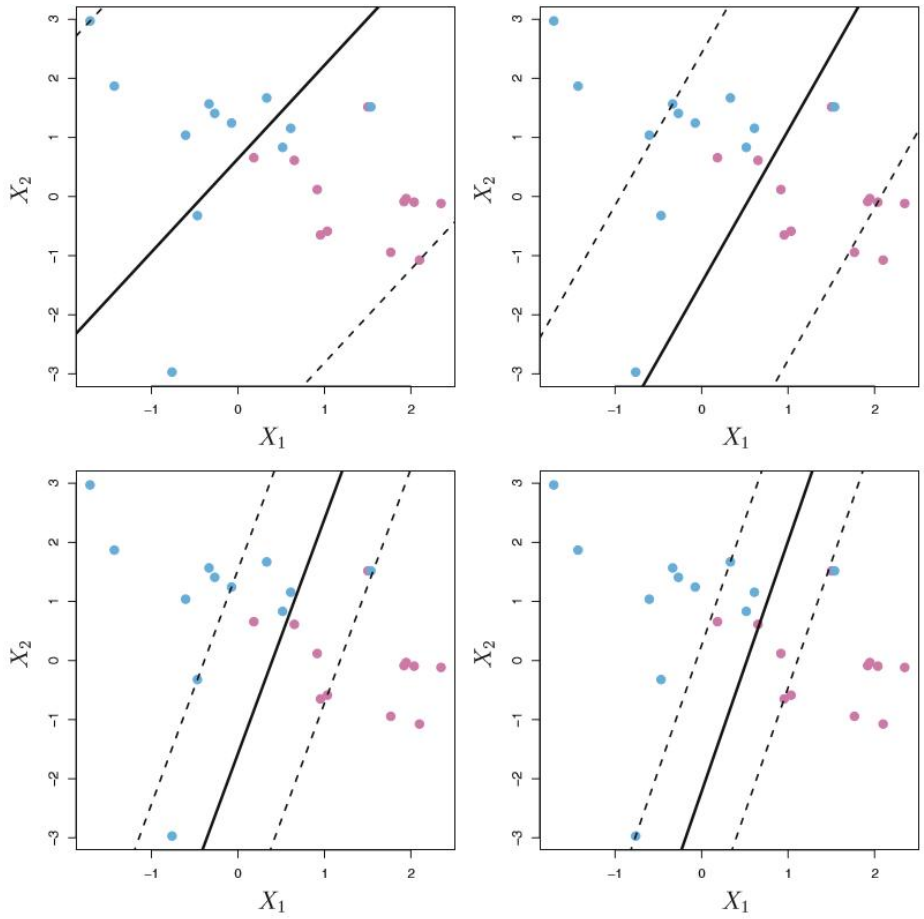
- 엡실론 I는 I번쨔 관측치가 초평면과 마진에 관해 어디에 위치하는지를 알려주는데. 만약 0이면 I가 0이면 올바르게 위치한다. 하지만 1보다 크면 M에 음수가 곱해져 좌표평면상에서 잘못된 부분에 위치하게 된다.

- C는 조율 파라미터로서 엡실론들의 합을 한정하며 마진의 침범과 초평면에 대해 잘못 위치하게 되는 것을 얼마나 허용할지 정해주는 파라미터이다. C=0이면 서포트 벡터 분류기는 그냥 최대 마진 분류기로 생각할 수 있다.

- C는 cross-validation을 통해 조율하고 정하며 bias-variance trade-off를 고려해야 한다. C가 작으면 허용마진이 작아지고 데이터에 오버피팅하는 문제가 발생할 수 있다. 즉 편향은 낮지만 분산이 높아지게 된다. 반대로 C가 크면 언더피팅의 문제가 발생해 마진의 위반이 더 많이 허용되고 편향은 높지만 분산은 낮아지게 된다.

-서포트 벡터 머신에서 마진에 대해 올바르게 놓은 데이터들은 모델에 영향을 주지 않는다. 서포트 벡터(마진상에 높이거나 마진에 대해 잘못 침범한 데이터들을 의미)들만이 서포트 벡터 분류기에 영향을 준다.

- C의 크기와 서포트 벡터의 수는 정비례관계에 있다.



- 서포트 벡터 분류기는 어느 정도 마진에 대한 침범과 잘못 분류를 허용한다 하더라도 여전히 선형성 경계로 데이터들을 분류한다는 한계를 가지고 있다. 이러한 문제를 해결하려면 비선형 결정경계를 사용해야 한다.

9.3 서포트 벡터 머신

- 선형 분류기를 비선형 결정경계를 제공하는 것으로 변환하는 일반적 메커니즘을 다루고
- 이 과정을 자동으로 하는 서포트 벡터 머신에 대해 알아보자

9.3.1 비선형 결정경계

- 서포트 벡터 분류기의 클래스들 사이에 경계가 비선형이면 변수들의 2차원 이상의 더 높은 차수의 다항식 함수를 사용하여 변수공간을 확장해 문제를 다루게 된다.

$$X_1, X_2, \ldots, X_p,$$

$$X_1, X_1^2, X_2, X_2^2, \ldots, X_p, X_p^2.$$

위의 식처럼 p개의 변수가 2p개가 된다.

$$\underset{\beta_0,\beta_{11},\beta_{12}\ldots,\beta_{p1},\beta_{p2},\epsilon_1,\ldots,\epsilon_n,M}{\text{maximize}} M \tag{9.16}$$

$$\text{subject to } y_i\left(\beta_0 + \sum_{j=1}^{p}\beta_{j1}x_{ij} + \sum_{j=1}^{p}\beta_{j2}x_{ij}^2\right) \geq M(1-\epsilon_i),$$

$$\sum_{i=1}^{n}\epsilon_i \leq C,\ \ \epsilon_i \geq 0,\ \ \sum_{j=1}^{p}\sum_{k=1}^{2}\beta_{jk}^2 = 1.$$

- (9.12)~(9.15) 의 maximum 마진에 대한 식은 위 경우 (9.16)처럼 변하게 된다. 위 식을 확장하면 더 고차원 비선형 결정경계에 대해 다룰 수 있다. 선형 결정경계에서 위의 2번째 줄의 좌변을 q(x)라 하면 q(x) = 0 일 때 선형 결정경계가 정의되는데 위 식의 경우 역시 q(x) = 0일 때 결정경계가 정의되기는 하지만 q(x)이 x에 대한 n차 다항식이기 때문에 비선형이 된다.

9.3.2 서포트 벡터 머신

- 서포트 벡터 머신은 앞에서 다룬 서포트 벡터 분류기의 확장으로 커널을 사용하여 변수공간을 확장한 결과이다. 변수공간의 확장은 어떻게 이루어지는지 간략하게 알아보자.

- 서포트 벡터 분류기에서의 해는 벡터들의 내적만이 관련된다. 그리고 선형 서포트 벡터 분류기에 대해 다음과 같이 나타낼 수 있다. 우선 내적을 구하는 식은

선형 서포트 벡터 분류기는 다음과 같이 나타낼 수 있다.

$$f(x) = \beta_0 + \sum_{i=1}^{n}\alpha_i\langle x, x_i\rangle, \tag{9.18}$$

- 비선형 결정경계, 즉 서포트 벡터 머신의 경우에 대해 알아보자. 우선 커널(K)에 대해 알아야 하는데 커널이란 두 관측치들의 유사성을 수량화하는 함수이다. 선형 커널의 식은 다음과 같고

$$K(x_i, x_{i'}) = \sum_{j=1}^{p} x_{ij} x_{i'j}, \qquad (9.21)$$

- 선형 커널은 피어슨 상관을 사용하여 데이터 쌍의 유사성을 수량화한다. 커널 함수의 형태를 변형할 수 있는데

$$K(x_i, x_{i'}) = (1 + \sum_{j=1}^{p} x_{ij} x_{i'j})^d. \qquad (9.22)$$

- 위 식은 d차 다항식 커널로 알려져 있다. 이를 서포트 벡터 분류기 알고리즘에 사용하면 훨씬 더 유연한 결정경계가 만들어진다. 그리고 그때의 분류기를 서포트 벡터 머신이라고 한다. 이 경우 서포트 벡터 머신의 함수는 다음 형태를 가진다.

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i). \qquad (9.23)$$

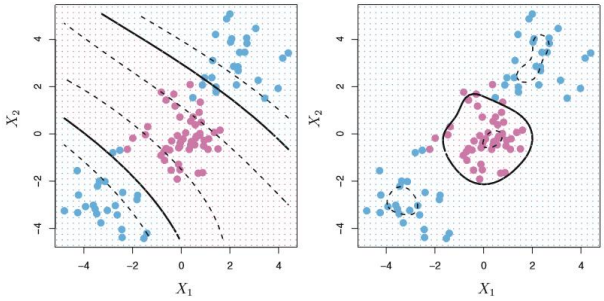
- S는 서포트인 점들의 인덱스 모임이다.

지금까지 다룬 다항식 커널은 비선형 커널의 한 예시이며 여러 다른 커널들이 있다. 그중 가장 많이 쓰이는 것이 방사커널(radial)이다.

$$K(x_i, x_{i'}) = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2). \qquad (9.24)$$

- 식 (9.16)에서처럼 원래 변수들을 이용하여 변수공간을 확장하는 대신 커널을 사용하는 것의 장점으로는 계산의 편리성인데 확장된 변수공간이 너무 커서 계산하기 힘들 기 때문이다. 커널을 이용하면 nC2의 서로 다른 모든 쌍 i와 i프라임에 대해서만 커널함수를 계산하면 되기 때문이다. 변수공간은 명시적이지 않으며 차원이 무한이기 때문에 계산이 아예 불가능하기도 한다.(예시 : 9.24의 식)

$$\langle x_i, x_{i'} \rangle = \sum_{j=1}^{p} x_{ij} x_{i'j}. \qquad (9.17)$$



왼쪽은 3차원 커널, 오른쪽은 방사커널

9.4 Multiclass SVM

Multiclass SVM에는 가장 널리 사용되는 두가지 방법이 있는데 OvO(일대일) 기법과 OvA(일대 전부)기법이다.

9.4.1 OvO 분류

- K개 클래스에 대하여 2개의 클래스 조합을 선택하여 분류하는 방식으로 모든 조합에 대해 수행하며 각 분류를 통해 판별된 결과 중 가장 많은 결과를 획득한 클래스를 최종 결과로 선택한다.

ex) 클래스가 4개 일 때(A,B,C,D) : AB, AC, AD, BC, BD, CD

- 별도의 설정은 필요 없으며, 데이터가 다중 클래스로 구성되어 있으면 자동적으로 OvO 방식으로 분류가 수행된다.

9.4.2 OvA 분류

- K개 클래스에 대하여 각 클래스 별로 소속 여부를 판별하는 분류를 수행하며 각 분류를 통해 판별된 결과 중 가장 많은 결과를 획득한 클래스를 최종 결과로 선택한다.

ex) 클래스가 4개 일 때(A,B,C,D) : A인지 아닌지 분류, B인지 아닌지 분류, C인지 아닌지 분류, D인지 아닌지 분류

9.5 로지스틱 회귀와의 상관관계

- SVM은 1990년대 중반 처음 소개되었는데 초평면을 이용해 분류를 하는 개념은 고전적인 로지스틱 회귀나 LDA같은 고전적 분류기법에 비해 큰 차이를 가지는 것처럼 보였다. 하지만 그 후 SVM과 고전적 통계방법들 사이에 깊은 연관이 있음이 드러났다.

- 먼저 서포트 벡터 분류기의 식인 f(X)=β0 + β1X1 + ...+ βpXp를 fitting하는 기준이었던 식 9.12 ~ 9.15을 다음과 같이 바꿔 쓸 수 있다는 것이 판명되었다.

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{minimize}} \left\{ \sum_{i=1}^{n} \max\left[0, 1 - y_i f(x_i)\right] + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}, \qquad (9.25)$$

- 람다는 양의 값을 가지는 튜닝 파라미터이고 그 값이 크면 베타값들은 작고 마진에 대한 더 많은 위반이 용인된다.

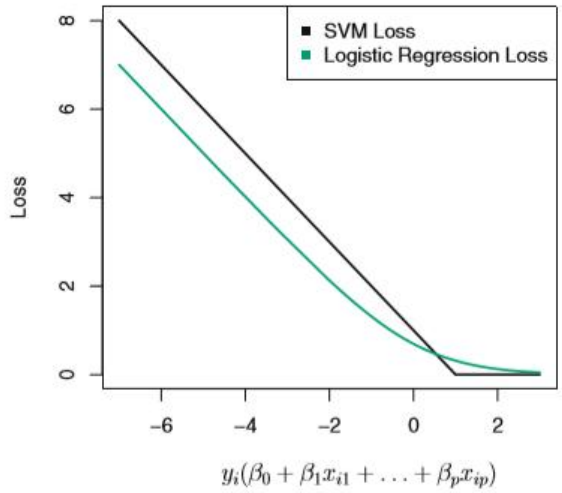
- 람다항은 릿지회귀에서 봤던 형태와 모습이 같으며 서포트 벡터 분류기에 대한 vias-variance tradeoff를 조정하는데 비슷한 역할을 한다.
위 식은 다음과 같이 쓸 수 있으며 loss + penalty 형태를 취한다.

$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\text{minimize}} \left\{ L(\mathbf{X}, \mathbf{y}, \beta) + \lambda P(\beta) \right\}. \qquad (9.26)$$

이 경우 loss function은 다음의 형태를 가지고 이를 hinge loss라고 부른다.

$$L(\mathbf{X}, \mathbf{y}, \beta) = \sum_{i=1}^{n} \max\left[0, 1 - y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip})\right].$$

힌지 로스와 로지스틱 회귀의 loss function을 비교해봤을때 다음과 같은데 둘이 상당히 유사하게 작동한다는 것을 알 수 있다.



- 가로축이 1일때를 기준으로 보자. 서포트 벡터 분류기는 서포트 벡터들만이 분류기에서 역할을 한다 했다. 위의 식에서 loss function이 1보다 큰 구간에서는 loss값이 0이기 때문이다.

- 반면 로지스틱 회귀의 loss function은 어느곳에서도 정확하게 0이 되지 않는다. loss function이 유사하기 때문에 로지스틱 회귀와 서포트 벡터 분류기는 보통 흡사한 분류 결과를 보이곤 한다.

- 또한 커널의 개념 역시 로지스틱 회귀 및 다른 분류모델에서 비선형 커널을 이용할 수 있다.

- SVM을 확장해서 서포트 벡터 회귀라는 회귀모델을 이용하기도 한다.