Chapter 12 SVM과 커널법

[학습목표]
이 장에서는 6장에서 알아본 선형 판별함수를 바탕으로, 일반화 오차를 줄이기 위한 최대 마진 분류기의 개념을 알아본다. 또한 선형 최대 마진 분류기에 커널함수를 이용한 고차원 매핑을 적용하여 선형 분류기가 가지는 한계점을 극복한 새로운 형태의 분류기인 서포트벡터머신(SVM)에 대하여 알아본다.

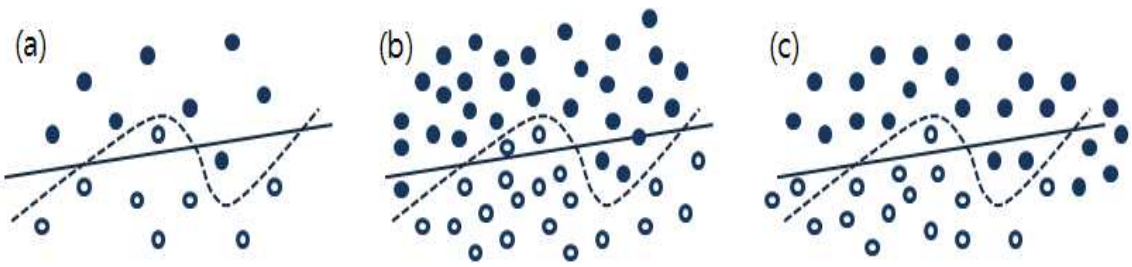
12. SVM과 커널법

12.1. 선형 초평면에 의한 분류

12.1.1 과다적합과 일반화 오차

이 장에서 소개하는 서포트벡터머신(SVM, Support Vector Machine)은 일반화 오차를 최소화할 수 있는 방향으로 학습을 수행하는 선형 분류기이다. 서포트벡터머신에 대하여 알아보기에 앞서, 먼저 1장에서 간단히 소개한 학습 오차와 일반화 오차에 대하여 다시 한 번 살펴보겠다. 1장에서 설명한 바와 같이, 학습 시스템의 성능을 평가하는 기준으로 학습 오차와 일반화 오차가 있다. 학습 오차란 학습 시에 사용된 데이터에 대한 학습 시스템의 출력과 원하는 출력과의 차이를 의미하고, 학습 시스템은 이 오차를 최소화하도록 학습을 진행한다. 그러나 학습 오차는 완성된 시스템의 성능을 평가하기 위한 정당한 기준으로 사용될 수 없다. 왜냐하면, 학습된 시스템이 실제로 사용될 때에는 학습 시에 사용된 데이터가 아닌 새롭게 주어지는 데이터에 대해 원하는 결과를 얻는 것이 목적이기 때문이다. 따라서 학습 시에 사용되지 않은 데이터에 대해 어느 정도 오차를 낼 것인지를 계산할 필요가 있는데, 이를 일반화 오차라고 한다. 일반화 오차는 앞으로 주어질 입력 데이터의 분포를 고려하여 그 확률분포로 평균한 오차로서 정의한다.

SVM은 일반화 오차를 감소시키는 방향으로 학습이 이루어 질 수 있도록 설계된 시스템이다. 구체적인 방법에 대해서는 다음 절에서 자세히 알아볼 것이나, 여기서는 학습 시스템의 복잡도와 학습 오차 및 일반화 오차의 관계에 대해 살펴봄으로서 작은 일반화 오차를 얻기 위한 방법을 생각해 보자. [그림 12-1]의 두 개의 클래스를 분류하는 문제에서, [그림 12-1a]와 같이 학습 데이터가 주어졌다고 하자. 이때 낮은 복잡도의 선형 분류기를 사용한 경우의 분류경계(실선)와 높은 복잡도의 비선형 분류기를 사용한 경우의 분류경계(점선)가 각각 그림과 같이 얻어졌다고 하자. [그림 12-1a]에서 보면, 비선형 분류기의 경우 학습 데이터를 완벽히 분류하여 학습 오차가 없는 반면, 선형 분류기의 경우 학습 데이터를 제대로 분류하지 못하여 학습 오차가 발생한다.



[그림 12-1] 학습 시스템의 복잡도와 일반화 오차의 관계

이와 같이 학습 오차만을 생각한 경우는 복잡도가 높은 비선형 분류기가 더 선호될 수 있으나, 일반화 오차를 생각하면 반드시 그렇지 않을 수도 있다. [그림 12-1]의 (b)와 (c)는 학습되지 않은 데이터에 대한 일반화 오차를 생각해 보기 위해, 앞으로 주어질 것으로 예상되는

데이터의 분포를 나타낸 것이다. 만약 데이터 집합이 [그림 12-1b]와 같은 분포를 가지고 있다면, 선형 분류기에 비해 비선형 분류기가 일반화 오차도 작은 값을 가지게 될 것이다. 그러나 [그림 12-1c]와 같은 분포를 가진 경우는 오히려 복잡도가 높은 비선형 분류기가 더 큰 일반화 오차를 가지게 된다.
이러한 현상이 발생한 이유는 학습에 사용된 데이터가 전체 데이터의 분포를 제대로 표현하지 못함에도 불구하고, 복잡도가 높은 분류기는 학습 데이터만을 잘 분류할 수 있도록 과다하게 학습했기 때문이다. 이와 같이 학습이 과다하게 일어나서 일반화 성능이 저하되는 현상을 과다적합이라고 하였다 (1장 참조). 과다학습을 피하고 일반화 오차를 줄이기 위해서는 학습 시스템의 복잡도를 적절히 조정하는 것이 매우 중요하다. SVM에서는 이러한 관점에서 선형 분류기를 사용하는 것에서 학습 시스템의 설계를 시작한다. 다음 절에서 이에 대해 알아보겠다.

12.1.2 선형 초평면 분류기

SVM은 6장에서 소개한 선형 판별함수에 의한 분류기에 기반을 두고 있다. 선형 분류기는 학습 시스템의 분류 복잡도가 가장 낮은 분류기로, 표현할 수 있는 결정경계에 제약이 많아 분류 성능 측면에서 좋은 결과를 기대하기는 힘드나 과다적합의 문제는 피할 수 있다. SVM은 이러한 선형 분류기의 장점을 취하여, 기본적인 선형 분류기에서 논의를 시작한다. 추후 커널법을 적용하여 선형 분류기가 가지는 표현 능력의 제약을 해결함으로써 복잡한 분류경계를 가진 문제에 대해서도 성공적으로 적용될 수 있는 분류기로 확장될 것이다. 이 절에서는 논의의 기본 단계로서 선형 판별함수로 정의되는 선형 초평면 분류기에 대하여 설명하고, 다음 절에서는 선형 판별함수를 학습하기위한 SVM의 특별한 목적함수와 학습 방법에 대하여 소개할 것이다. 비선형 결정경계를 얻기 위한 커널법에 대해서는 12.4절에서 설명할 것이다.
입력 $\boldsymbol{x}$에 대한 선형 초평면 결정경계 함수는 다음과 같이 정의할 수 있다.

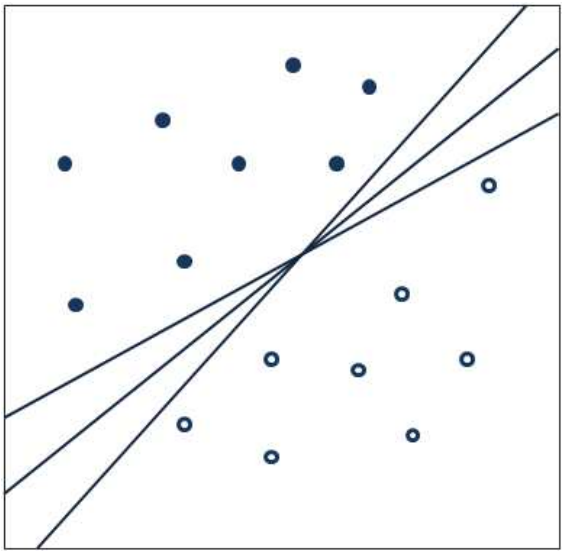
$$g(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + w_0 = \boldsymbol{w}^T\boldsymbol{x} + w_0 = \sum_{i=1}^{n} w_i x_i + w_0 \qquad \text{[식 12-1]}$$

이 결정경계 함수를 이용하여 주어진 입력 데이터 $\boldsymbol{x}$에 대한 분류는 다음과 같은 판별함수에 의해 수행된다.

$$f(\boldsymbol{x}) = \mathrm{sign}(g(\boldsymbol{x})) \qquad \text{[식 12-2]}$$

즉, 판별함수 $f(\boldsymbol{x})$가 1의 값을 가지는 경우 입력 데이터 $\boldsymbol{x}$는 클래스 C_1으로 할당되며, 반대로 -1의 값을 가지는 경우는 클래스 C_2로 할당된다. 우리는 학습을 통하여 주어진 학습 데이터들이 올바른 클래스에 할당될 수 있도록 함수 $g(\boldsymbol{x})$를 찾아주어야 한다.
[그림 12-2]에서 보는 바와 같이 선형 분리가 가능한 학습 데이터 집합이 주어졌다고 하자. 이 경우에는 모든 데이터를 바르게 분류하는 분류경계가 매우 많다. 모든 분류경계가 학습

오차만을 고려한 경우는 같은 성능을 가지지만, 일반화 오차까지 고려하게 되면 그 성능이 달라질 수 있다. 따라서 여러 선형 분류기 중 일반화 오차를 최소로 하는 분류기를 얻기 위한 전략이 필요하다. SVM에서는 이를 위해 마진(Margin)이라는 개념을 도입하여 각 분류기를 평가하는 목적함수를 정의한다. 다음 절에서 이에 대해 자세히 알아보겠다.



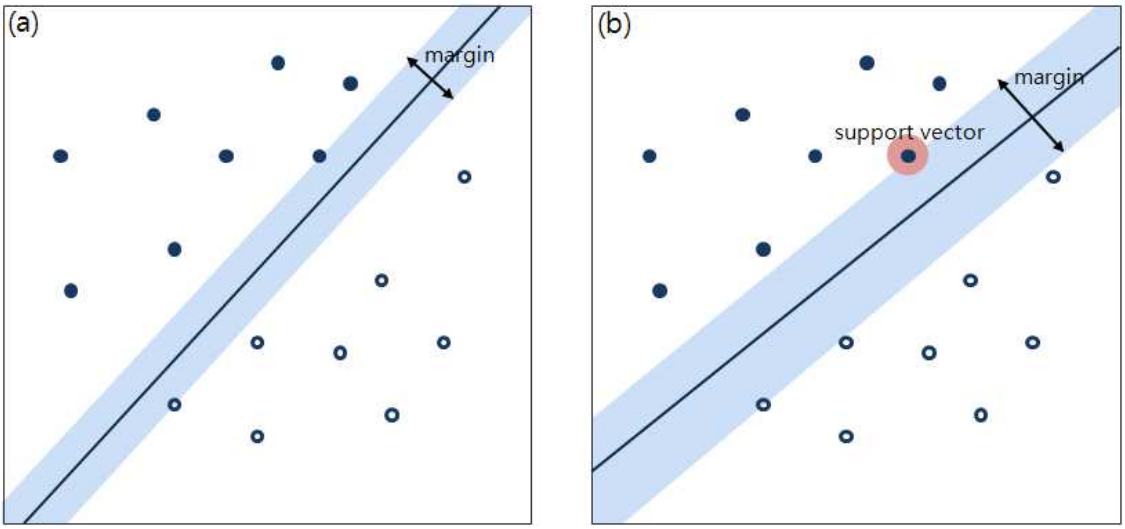
[그림 12-2] 최소 학습 오차를 만족하는 여러 가지 선형 결정경계

12.2 서포트벡터머신 (SVM)

12.2.1 최대 마진 분류기

SVM에서는 여러 선형 결정경계 중 최적의 경계를 찾기 위하여, 마진을 이용하여 학습의 목적함수를 정의하였다. 따라서 SVM을 최대 마진 분류기(maximum margin classifier)라고도 부른다. 먼저 마진의 개념에 대하여 알아보겠다. 두 개의 클래스로 이루어진 학습 데이터에 대해, 이를 완전히 분류하는 분류경계가 주어졌다고 하자. 이때 <마진(margin)>이란, 학습 데이터들 중에서 분류경계에 가장 가까운 데이터로부터 분류경계까지의 거리를 의미한다. 또 이때 분류경계에 가장 가까운 곳에 위치한 데이터를 <서포트벡터(Support Vector)>라고 한다.

학습 데이터가 정해져 있는 경우, 마진과 서포트벡터는 분류경계에 따라 달라진다. 여러 가지 분류경계에 따른 마진을 [그림 12-3]에 나타내고 있다. 일반화 오차가 작아지기 위해서는 두 클래스간의 간격을 최대로 하는 것이 좋으므로, 마진을 최대로 하는 분류경계를 찾는 것이 바람직하다. 이러한 목적에 맞추어 최적화된 선형 분류경계를 찾는 분류기를 <최대 마진 분류기(maximum margin classifier)>라고 하며, 일반적으로 <서포트벡터머신>으로 알려져 있다.



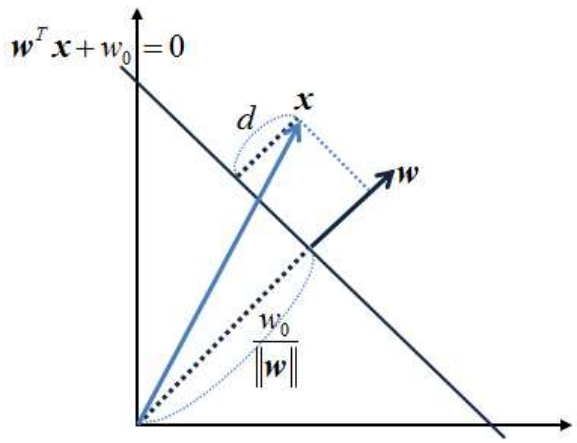
[그림 12-3] 분류경계에 따른 마진의 차이

최대 마진을 가진 선형 분류경계(초평면)를 얻기 위하여 다음과 같은 선형 분류경계를 생각한다.

$$g(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 = \sum_{i=1}^{n} w_i x_i + w_0 = 0 \qquad \text{[식 12-3]}$$

이 식에서 $\boldsymbol{w}$는 초평면의 법선벡터가 되고, w_0는 원점에서 직선까지의 거리를 결정하는 값이 된다.([그림 12-4] 참조) 이때, 한 점 $\boldsymbol{x}$에서 분류경계까지의 거리는 다음과 같이 계산될 수 있다.

$$d = \frac{\boldsymbol{w}^T\boldsymbol{x} + w_0}{\| \boldsymbol{w} \|} \qquad \text{[식 12-4]}$$



[그림 12-4] 한 점에서 분류경계까지의 거리

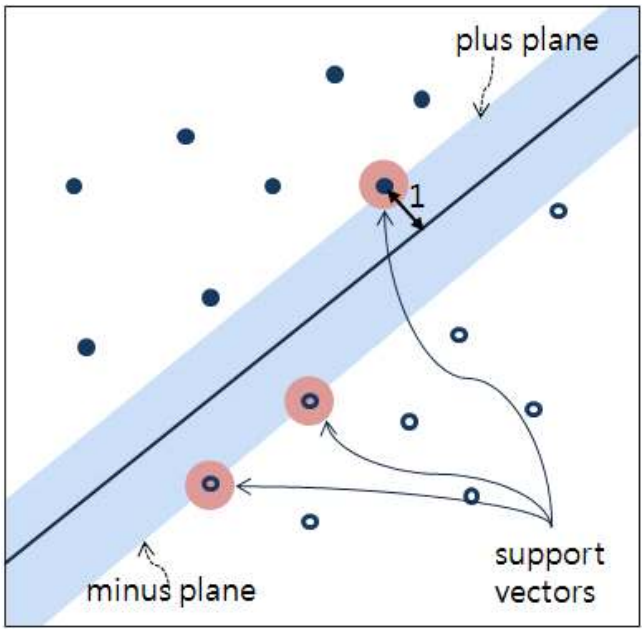
이 분류경계를 중심으로 $\boldsymbol{w}^T\boldsymbol{x}+w_0>0$ 이 되는 클래스 C_1의 영역과 $\boldsymbol{w}^T\boldsymbol{x}+w_0<0$ 이 되는 C_2의 영역으로 나뉘게 되고, 각 영역에서 분류경계에 가장 가까운 데이터, 즉 서포트벡터 $\boldsymbol{\chi}$에 대해서 $\boldsymbol{w}$와 w_0의 비율을 적절히 조절하여 다음과 같은 조건이 성립되도록 정한다.

$$\begin{cases}\boldsymbol{w}^T\boldsymbol{\chi}+w_0=+1 & \text{if } \boldsymbol{\chi}\in C_1\\ \boldsymbol{w}^T\boldsymbol{\chi}+w_0=-1 & \text{if } \boldsymbol{\chi}\in C_2\end{cases} \qquad \text{[식 12-5]}$$

이 때 클래스 C_1의 서포트벡터를 지나는 초평면, 즉, $\boldsymbol{w}^T\boldsymbol{\chi}+w_0=+1$ 을 플러스평면(Plus-Plane)이라 하고, 클래스 C_2의 서포트벡터를 지나는 초평면, 즉, $\boldsymbol{w}^T\boldsymbol{\chi}+w_0=-1$ 을 마이너스평면 (Minus-Plane)이라고 한다.([그림 12-5] 참조) 이 분류경계에 의해 분류를 수행하는 경우, 주어진 데이터 $\boldsymbol{x}$에 대해, $\boldsymbol{w}^T\boldsymbol{x}+w_0$가 1보다 크거나 같은 경우는 클래스 C_1으로, -1보다 작거나 같은 경우는 클래스 C_2로 할당하게 된다.
이때 마진은 플러스평면까지의 거리와 마이너스평면까지의 거리의 합으로 정의 계산될 수 있으므로 [식 12-4]와 [식 12-5]를 이용하면 [식 12-6]과 같은 관계식을 얻을 수 있다.

$$\mathrm{M}=|\chi^+-\chi^-|=\frac{1}{\|\boldsymbol{w}\|}(\boldsymbol{w}^T\chi^+-\boldsymbol{w}^T\chi^-)=\frac{2}{\|\boldsymbol{w}\|} \qquad \text{[식 12-6]}$$

따라서 마진을 최대화하기 위해서는 $\|\boldsymbol{w}\|$ 의 값을 최소화 하여야 함을 알 수 있다. 다음 절에서는 $\|\boldsymbol{w}\|$ 의 값을 최소화하는 파라미터를 학습을 통해 추정하는 방법에 대하여 알아보겠다.



[그림 12-5]. 3개의 서포트벡터와 초평면

12.2.2 SVM의 학습

최대 마진 분류기 SVM을 얻기 위하여, 학습 데이터를 이용하여 선형 분류기의 파라미터 값을 최적화 하는 방법에 대해 알아본다. 마진을 최대화하면서 원하는 분류를 수행하는 분류 경계를 찾기 위해 파라미터 $\boldsymbol{w}$와 w_0를 추정하여야 한다. 학습 데이터 집합 $\{(\boldsymbol{x_i}, y_i)\}_{i=1\ldots N}$ 이 주어졌다고 가정한다. 이때 출력값 y_i는 다음과 같이 정한다.

$$\begin{cases} y_i = +1 & \text{if } \boldsymbol{x} \in C_1 \\ y_i = -1 & \text{if } \boldsymbol{x} \in C_2 \end{cases} \qquad \text{[식 12-7]}$$

이 학습 데이터들에 대해 파라미터가 만족해야하는 조건을 정리하면 다음과 같다.

$$y_i(\boldsymbol{w^T x_i} + w_0) - 1 \geq 0 \qquad (i = 1, \ldots, N) \qquad \text{[식 12-8]}$$

위의 [식 12-8]의 조건은 모든 데이터에 대해 제대로 분류를 수행하기 위한 조건으로, 아래의 두 조건을 하나의 식으로 표현한 것이다.

$$\begin{cases} (\boldsymbol{w^T x_i} + w_0) \geq +1 & \text{for } y_i = +1 \\ (\boldsymbol{w^T x_i} + w_0) \leq -1 & \text{for } y_i = -1 \end{cases} \qquad \text{[식 12-9]}$$

또한 SVM에서는 마진을 최대화하는 조건이 추가되므로, 다음 [식12-10]에서 정의되는 목적함수를 최소화해야 한다.

$$J(\boldsymbol{w}) = \frac{\| \boldsymbol{w} \|^2}{2} \qquad \text{[식 12-10]}$$

이와 같이 하나의 함수식 $J(\boldsymbol{w})$를 최소화하는 동시에 또 다른 조건 $y_i(\boldsymbol{w^T x_i} + w_0) - 1 \geq 0$ 을 만족시키는 파라미터를 찾는 문제를 해결하기 위해서는 라그랑제 승수를 이용한 최적화 방법을 적용하여야 한다. 라그랑제 승수 $(\alpha_i \geq 0, i = 1,2,\ldots,N)$ 를 도입하여 두 조건을 하나의 함수식으로 표현하면 다음과 같다.

$$J(\boldsymbol{w}, w_0, \alpha) = \frac{1}{2} \| \boldsymbol{w} \|^2 - \sum_{i=1}^{N} \alpha_i \{ y_i(\boldsymbol{w^T x_i} + w_0) - 1 \} \qquad \text{[식 12-11]}$$

이 식을 라그랑지안 함수라 하고, 이를 원래 주어진 파라미터 $\boldsymbol{w}$와 w_0에 대해 극소화하고 라그랑제 승수 α_i에 대해 극대화 하면 우리가 원하는 조건을 만족하는 파라미터 값을 찾을 수 있다.

라그랑지안 함수 $J(\boldsymbol{w}, w_0, \alpha)$ 를 이용하여 파라미터를 추정하기 위해, 먼저 파라미터 $\boldsymbol{w}$와

w_0에 대해 극소화한다. 즉 $J(\boldsymbol{w}, w_0, \boldsymbol{\alpha})$를 $\boldsymbol{w}$와 w_0에 대해 미분하여 다음 관계식을 얻을 수 있다.

$$\frac{\partial J(\boldsymbol{w}, w_0, \boldsymbol{\alpha})}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x_i} = 0 \qquad \text{[식 12-12]}$$

$$\frac{\partial J(\boldsymbol{w}, w_0, \boldsymbol{\alpha})}{\partial w_0} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \qquad \text{[식 12-13]}$$

[식 12-12]로부터 이렇게 $\boldsymbol{w}$에 대한 관계식 $\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x_i}$ 와 [식 12-13]을 이용하여 [식 12-11]을 다시 표현하면 다음 식과 같이 α_i에 대하여 표현된 새로운 형태의 목적함수를 얻을 수 있다.

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i^T x_j} \qquad \text{[식 12-14]}$$

이와 함께 α_i는 다음 조건을 만족하여야 한다.

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \ \alpha_i \geq 0 (i = 1, ..., N) \qquad \text{[식 12-15]}$$

이렇게 $Q(\boldsymbol{\alpha})$로 표현된 최적화 문제를 원래의 라그랑지안 함수 $J(\boldsymbol{w}, w_0, \boldsymbol{\alpha})$로 표현된 최적화 문제에 대한 이원적 문제(Dual problem)라고 하며, $J(\boldsymbol{w}, w_0, \boldsymbol{\alpha})$를 최적화하는 대신 $Q(\boldsymbol{\alpha})$를 최적화하여 원하는 파라미터를 얻을 수 있다. $Q(\boldsymbol{\alpha})$를 이용하여 최적화 문제를 접근할 때 얻을 수 있는 장점은, 목적 함수 $Q(\boldsymbol{\alpha})$는 α_i에 대한 이차함수로 이차계획법(Quadratic programming)을 이용하면 간단히 해를 구할 수 있을 뿐 아니라, 유일한 최대치를 갖게 된다는 것이다. 이는 신경망에서 기울기 강하 학습을 통해 해를 찾는 경우 지역극소에 빠지게 되는 문제점을 갖고 있음을 생각하면, 매우 큰 장점이라고 볼 수 있다.
이차계획법을 이용하여 $Q(\boldsymbol{\alpha})$를 최대화하는 추정치 $\hat{\alpha}_i$를 찾으면, 이를 이용하여 [식 12-12]와 [식 12-13]으로부터 다음과 같이 $\boldsymbol{w}$와 w_0도 찾을 수 있다.

$$\hat{\boldsymbol{w}} = \sum_{i=1}^{N} \hat{\alpha}_i y_i \boldsymbol{x}_i \qquad \text{[식 12-16]}$$

$$\hat{w_0} = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \hat{\boldsymbol{w}}^{\boldsymbol{T}} \boldsymbol{x_i} \right) = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{N} \alpha_j y_j \boldsymbol{x_j^T x_i} \right) \qquad \text{[식 12-17]}$$

여기서 추정치 $\hat{\alpha}_i$는 한 가지 중요한 성질을 가진다. [식 12-11]에서 $\hat{\alpha}_i$와 함께 묶여 있는 조건식 $y_i(\boldsymbol{w^T x_i} + w_0) - 1 \geq 0 \ (i = 1, ..., N)$ 을 생각해 보자. [그림 12-5]에서 본 바와 같이 대부분의 데이터는 결정경계와는 떨어져 있어서 $y_i(\boldsymbol{w^T x_i} + w_0) - 1 \geq 0$ 의 조건식을 만족하게 된

다. 이 경우에 [식 12-11]을 최대화 하는 음이 아닌 $\hat{\alpha}_i$의 값은 0이 됨을 쉽게 알 수 있다. 따라서 대부분의 학습 데이터에 대응되는 라그랑제 승수 $\hat{\alpha}_i$는 0이 되며, 오직 $y_i(\boldsymbol{w}^T\boldsymbol{x_i}+w_0)-1=0$ 인 경우만 $\hat{\alpha}_i$가 0이 아닌 값을 가지게 된다.
그런데 $y_i(\boldsymbol{w}^T\boldsymbol{x_i}+w_0)-1=0$ 을 만족하는 데이터 $(\boldsymbol{x_i},y_i)$는 결정경계에 가장 가까이 있어 마진을 결정하는 데이터로, 서포트벡터가 된다. 즉, 오로지 서포트벡터 데이터에 대해서만 $\hat{\alpha}_i$는 0이 아닌 값을 가지게 됨을 알 수 있다. [그림 12-5]에서는 오직 세 개의 $\hat{\alpha}_i$만이 의미 있는 값을 가진다.

12.2.3 SVM에 의한 분류

추정된 파라미터를 이용하여 새로운 데이터에 대한 분류를 수행하는 방법에 대해 알아보겠다. 새로운 데이터에 대해 클래스를 할당하기 위해 사용하는 결정함수는 [식 12-18]에 나타난 것과 같이 주어진다.

$$g(\boldsymbol{x})=\boldsymbol{w}^T\boldsymbol{x}+w_0=\sum_{i=1}^{n}w_ix_i+w_0=0 \qquad \text{[식 12-18]}$$

이 함수식에 학습 데이터를 이용하여 추정된 파라미터 [식 12-16]과 [식 12-17]을 대입하면 다음과 같이 $\hat{\alpha}_i$와 학습 데이터를 이용한 식을 얻을 수 있다.

$$\begin{aligned} f(\boldsymbol{x}) &= \text{sign}\,(g(\boldsymbol{x})) = \text{sign}(\hat{\boldsymbol{w}}^T\boldsymbol{x}+\hat{w}_0) \\ &= \text{sign}\left(\sum_{i=1}^{N}\hat{\alpha}_i y_i \boldsymbol{x_i}^T\boldsymbol{x_i}+\hat{w_0}\right) \end{aligned} \qquad \text{[식 12-19]}$$

이 때 $\text{sign}(x)$는 부호 함수로 x의 값이 양수이면 +1, 음수이면 -1의 출력값을 가진다. 따라서 새로운 데이터 $\boldsymbol{x}$가 주어지면, 학습 데이터 집합 $\{(\boldsymbol{x_i},y_i)\}_{i=1\cdots N}$ 와 $\hat{\alpha}_i$의 값들을 이용하여 함수 $f(\boldsymbol{x})$를 계산한 후, 1인 경우 클래스 C_1로, -1인 경우 클래스 C_2로 분류한다.
여기서 한 가지 고려해야할 점은, 학습 데이터의 수가 많은 경우 분류함수 $f(\boldsymbol{x})$의 정의에 의하면 함수값을 계산하기 위해 저장해 두어야 하는 데이터의 수도 많고 계산량도 많아질 가능성이 있다는 점이다. 그러나 앞 절에서 잠깐 언급한 바와 같이, 대부분의 $\hat{\alpha}_i$들은 0이 되어 사라지고, 서포트벡터가 되는 일부의 데이터에 대응되는 $\hat{\alpha}_i$들만 의미 있는 값을 가지므로, 실제로 $f(\boldsymbol{x})$의 값을 얻기 위해서 필요한 데이터 수와 계산량은 현격히 줄어든다는 것을 알 수 있다. [그림 12-5]의 경우에는 서포트벡터가 되는 데이터는 모두 3개 이므로, 이들 데이터만 분류함수 $f(\boldsymbol{x})$를 계산하기 위해서 저장해 두면 된다.

다음에 SVM의 학습과 인식을 위한 과정을 단계별로 정리하였다.

[선형 SVM 분류기의 학습과 인식 단계]

① N개의 입출력 쌍으로 이루어진 학습 데이터 집합 $X=\{(\boldsymbol{x}_i, y_i)\}_{i=1\cdots N}$ 을 준비한다. 이때 목표 출력값은 $y_i \in \{-1,1\}$ $(i=1,...,N)$을 만족한다.

② 다음과 같은 과정을 통해 SVM을 학습한다.

②-1. 학습 데이터를 이용하여 파라미터 추정을 위한 목적함수 $Q(\boldsymbol{\alpha})$를 정의한다.

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{x}_i^T\boldsymbol{x}_j \quad , \ \sum_{i=1}^{N}\alpha_i y_i = 0,\ \alpha_i \geq 0 (i=1,...,N)$$

②-2. 주어진 조건을 만족하면서 $Q(\boldsymbol{\alpha})$를 최소화하는 추정치 $\hat{\alpha}_i$를 이차계획법에 의해 찾는다.

②-3. $\hat{\alpha}_i \neq 0$이 되는 서포트벡터를 찾아 집합 $X_S = \{\boldsymbol{x}_i \in X | \hat{\alpha}_i \neq 0\}$ 를 생성한다.

②-4. $\hat{\alpha}_i$와 서포트벡터를 이용하여 $\hat{w}_0$를 계산한다.

$$\hat{w_0} = \frac{1}{N_S}\sum_{\boldsymbol{x}_i \in X_S}\left(y_i - \sum_{\boldsymbol{x}_j \in X_S}\alpha_j y_j \boldsymbol{x}_j^T\boldsymbol{x}_i\right)$$

이때 N_S는 집합 X_S의 원소의 수이다.

②-5. 서포트벡터 집합 $X_S = \{\boldsymbol{x}_i \in X | \hat{\alpha}_i \neq 0\}$ 와 파라미터 벡터 $\hat{\boldsymbol{\alpha}}$, 그리고 $\hat{w}_0$를 저장해 둔다.

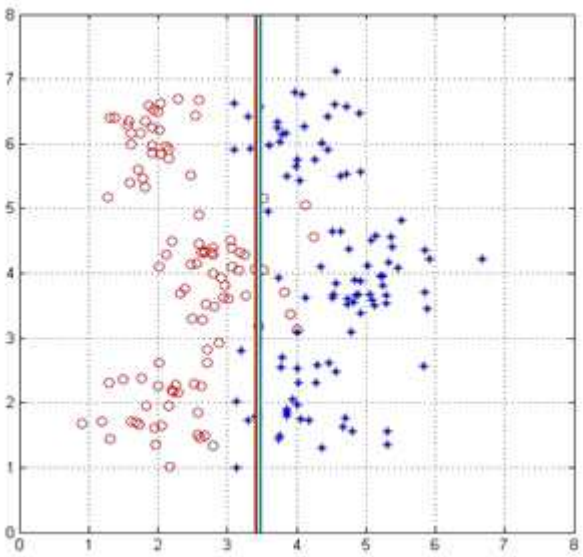
③ 새로운 데이터 $\boldsymbol{x}$가 주어지면, 저장해둔 서포트벡터와 파라미터를 이용하여 다음 판별함수로 분류를 수행한다.

$$f(\boldsymbol{x}) = \text{sign}\left(\sum_{\boldsymbol{x}_i \in X_S}\hat{\alpha}_i y_i \boldsymbol{x}_i^T\boldsymbol{x} + \hat{w_0}\right)$$

12.3 슬랙변수를 가진 SVM

12.3.1 슬랙변수의 도입

앞 절에서는 주어진 학습 데이터가 선형 분리 가능한 경우에 대해 알아보았다. 그러나 일반적으로 주어지는 학습 데이터가 선형 분리 가능하다고 가정하는 것은 무리가 있다. 따라서 [그림 12-6]과 같이 선형 분류기를 가지고 분류를 수행하는 경우 잘못 분류되는 데이터가 존재하는 경우에 대한 처리방법을 생각해야 한다.



[그림 12-6] 선형 분리 불가능한 데이터 집합의 예

이를 처리하기 위해 먼저 잘못 분류된 데이터로부터 해당 클래스의 분류경계선까지의 거리를 나타내는 슬랙변수 $\xi_i \ (i=1,...,N)$를 도입한다.([그림 12-7] 참조) 슬랙변수를 포함하여 학습 데이터에 대한 분류조건을 다음과 같이 정의한다.

$$\begin{cases} \left(\boldsymbol{w}^T\boldsymbol{x_i}+w_0\right) \geq +1-\xi_i & \text{for } y_i=+1 \\ \left(\boldsymbol{w}^T\boldsymbol{x_i}+w_0\right) \leq -1+\xi_i & \text{for } y_i=-1 \end{cases} \qquad \text{[식 12-20]}$$

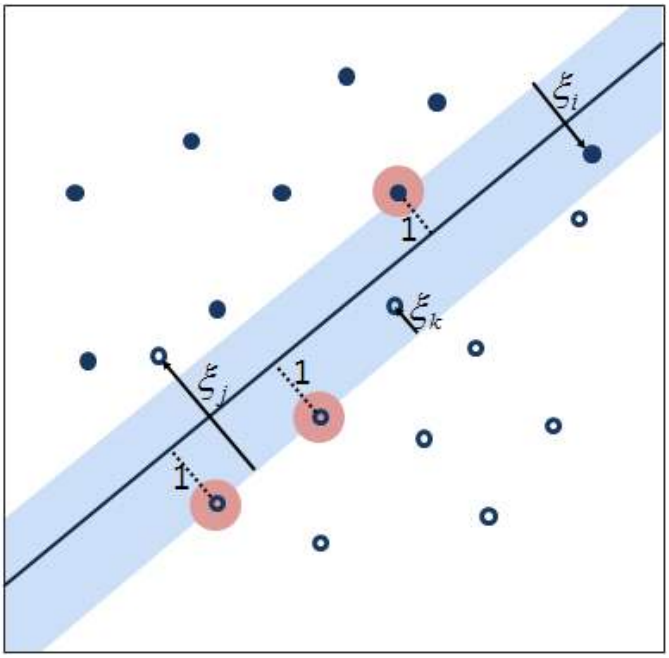
새롭게 정의된 분류 조건과 기존의 조건을 비교해보면 슬랙변수의 의미를 명확히 이해할 수 있다. 클래스 C_1에 속하는 데이터들에 대한 기존의 조건은 다음과 같았다.

$$\left(\boldsymbol{w}^T\boldsymbol{x_i}+w_0\right) \geq +1 \quad \text{for } y_i=+1 \qquad \text{[식 12-21]}$$

이 조건이 의미하는 바는 클래스 C_1에 속하는 모든 데이터들은 엄격하게 플러스평면보다

윗부분에 존재해야 한다는 것임에 반해, 슬랙변수 ξ_i를 추가함으로써 클래스 C_1에 속하는 데이터가 플러스평면보다 ξ_i 만큼 아래 부분에 존재할 수 있도록 허용하게 된다. 따라서 ξ_i가 클수록 더 심한 오분류를 허용함을 의미한다. 이러한 의미를 가진 슬랙변수를 추가한 분류 조건을 하나의 식으로 통합하여 표현하면 다음과 같다.

$$y_i(\boldsymbol{w}^T\boldsymbol{x_i}+w_0)\geq 1-\xi_i \qquad (i=1,...,N) \qquad \text{[식 12-22]}$$



[그림 12-7] 슬랙변수의 도입

12.3.2 파라미터의 추정

슬랙변수를 추가한 분류 조건을 만족하고 마진을 최대화하는 파라미터를 찾기 위한 목적함수는 다음과 같이 정의할 수 있다.

$$J(\boldsymbol{w},\boldsymbol{\xi})=\frac{1}{2}\|\boldsymbol{w}\|^2+c\sum_{i=1}^{N}\xi_i \qquad \text{[식 12-23]}$$

이 함수를 최소화하는 동시에 다음 조건을 만족해야 한다.

$$y_i(\boldsymbol{w}^T\boldsymbol{x_i}+w_0)\geq 1-\xi_i\,, \quad \xi_i\geq 0, \qquad (i=1,...,N) \qquad \text{[식 12-24]}$$

목적함수에서 슬랙변수에 의존하는 항 $c\sum_{i=1}^{N}\xi_i$는 ξ_i의 값을 되도록 최소화하여 오분류의 허용도를 낮추기 위해 추가되었다. c는 이 최소화 조건을 반영하는 정도를 결정하는 값으로 사용자가 적절히 정해주어야 한다. 만약 c의 값이 크면, ξ_i의 값이 커지는 것을 강하게 저지하므로 오분류 오차가 적어지며, 반대로 c의 값이 적어지면 오분류 허용도가 높아진다. [식 12-23]의 목적함수와 [식 12-24]의 조건을 함께 결합하여 라그랑제 함수식으로 표현하면 다음과 같다.

$$J(\boldsymbol{w}, w_0, \boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + c\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i\{y_i(\boldsymbol{w}^T\boldsymbol{x_i} + w_0) - 1 + \xi_i\} - \sum_{i=1}^{N}\beta_i\xi_i \qquad \text{[식 12-25]}$$

이는 [식 12-11]에 슬랙변수에 의존하는 항 $c\sum_{i=1}^{N}\xi_i$와 슬랙변수를 양의 값으로 유지하기 위한 새로운 라그랑제 상수 β_i를 포함하는 항 $\sum_{i=1}^{N}\beta_i\xi_i$이 마지막에 추가된 형태가 되었다. 앞 절에서와 유사한 계산을 통해 이원적 문제를 찾으면 다음과 같은 함수 $Q(\boldsymbol{\alpha})$를 얻을 수 있다.

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \boldsymbol{x_i}^T\boldsymbol{x_j} \qquad \text{[식 12-26]}$$

이와 함께 다음과 같은 조건을 만족해야 한다.

$$\sum_{i=1}^{N}\alpha_i y_i = 0,\ 0 \le \alpha_i < c \quad (i = 1, ..., N) \qquad \text{[식 12-27]}$$

앞 절의 경우와 비교해 보면, 최대화할 목적함수 $Q(\boldsymbol{\alpha})$는 앞 절과 완전히 일치하고 단지 라그랑제 승수 α_i가 c보다 작거나 같다는 조건만이 추가되었다. 따라서 이를 만족하는 파라미터 α_i값은 이차계획법으로 쉽게 찾을 수 있으며, 이로부터 나머지 파라미터도 다음과 같이 추정할 수 있다.

$$\widehat{\boldsymbol{w}} = \sum_{i=1}^{N}\alpha_i y_i \boldsymbol{x_i} \qquad \text{[식 12-28]}$$

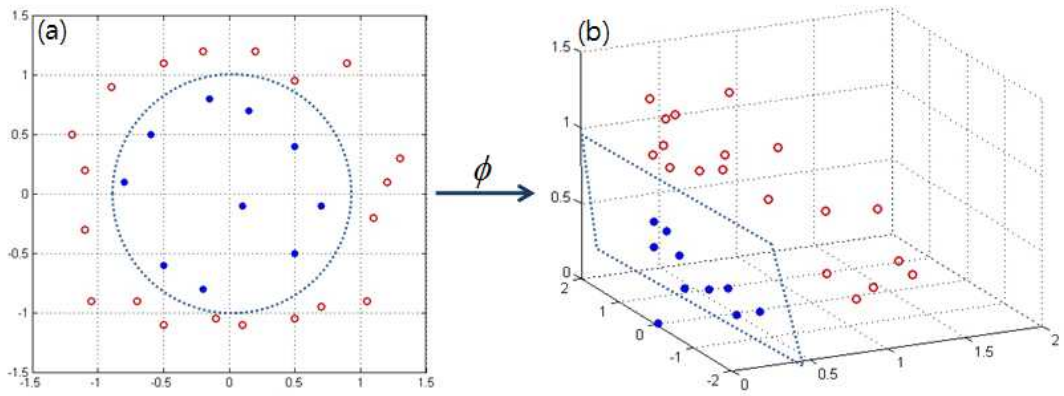
$$\widehat{w_0} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{\boldsymbol{w}^T}\boldsymbol{x_i}) = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \sum_{j=1}^{N}\alpha_j y_j \boldsymbol{x_j}^T\boldsymbol{x_i}\right) \qquad \text{[식 12-29]}$$

이 식에서 알 수 있듯이 α_i값만 정해지면 $\boldsymbol{w}$와 w_0의 값은 슬랙변수가 없는 경우와 완전히 일치하므로, 분류 시에도 마찬가지로 동일한 분류함수에 의해 분류가 가능하다.

12.4 커널법

12.4.1 커널의 필요성

선형 분리가 불가능한 문제의 경우, 앞 절에서와 같이 슬랙변수를 도입하여 어느 정도 해결할 수 있으나 결국 선형 초평면을 분류경계로 사용하는 것이므로 제약이 따른다. 따라서 보다 적극적인 해결책이 필요하다. 비선형 분류 문제를 해결하기 위해 저차원의 입력 $\boldsymbol{x}$를 보다 고차원의 공간의 값 $\phi(\boldsymbol{x})$로 매핑시키는 함수 ϕ를 생각한다.



[그림 12-8] 고차원 공간 매핑

[그림 12-8]에 간단한 예를 보이고 있다. 이 함수는 2차원 공간상의 한 점을 3차원 공간상의 한 점으로 매핑시키는 함수로, 다음과 같이 정의된다.

$$\Phi : R^2 \rightarrow R^3 \qquad \text{[식 12-30]}$$
$$(x_1, x_2) \mapsto (x_1^2,\ \sqrt{2}\, x_1 x_2\ ,\ x_2^2)$$

이렇게 정의된 함수를 이용하여 주어진 입력 데이터를 고차원 공간으로 보냄으로써 얻는 효과는 [그림 12-8]에 잘 나타나 있다. 2차원 공간상에서는 두 클래스는 원형의 결정경계를 가지는 비선형 문제이다. 그러나 이것을 3차원 공간으로 보냄으로써 선형 평면으로 분류 가능한 선형 문제로 변화되었다.

이렇게 입력 데이터의 차원을 높임으로써 문제를 선형화하면 간단한 선형 분류기를 사용하여 분류를 수행할 수 있는 반면, 차원을 높임으로 인해 발생하는 계산량의 증가와 같은 부작용도 고려해야만 한다. 이러한 부작용을 해결하기 위해 제안된 방법이 커널법이다.

12.4.2 커널법과 SVM

n차원의 입력 데이터 $\boldsymbol{x}$를 m차원의 특징 데이터 $\phi(\boldsymbol{x})$로 매핑시킨 후 이를 SVM을 이용하여 분류한다고 가정하자. 만약 차원 m이 아주 크다면 고차원의 특징벡터 $\phi(\boldsymbol{x})$를 사용하여 계산하는 것은 현실적이지 못하다. 그런데, SVM에서 수행하고 있는 연산을 살펴보면, 개개의 값 $\phi(\boldsymbol{x})$이 아니라 두 벡터의 내적, 즉, $\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y})$를 사용하고 있음을 알 수 있다. 따라서 고차원 매핑 $\phi(\boldsymbol{x})$를 정의하는 대신에 $\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y})$를 하나의 함수 $k(\boldsymbol{x}, \boldsymbol{y})$로 정의하여 사용한다. 이렇게 정의되는 함수를 커널 함수라고 한다.
예를 들어 앞에서 살펴본 2차원에서 3차원으로의 매핑 함수의 경우 커널 함수는 다음과 같이 정의될 수 있다.

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{y}) &= \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y}) \\ &= (x_1^2,\ \sqrt{2}\, x_1 x_2,\ x_2^2) \cdot (y_1^2,\ \sqrt{2}\, y_1 y_2,\ y_2^2) \\ &= (\boldsymbol{x} \cdot \boldsymbol{y})^2 \end{aligned} \qquad \text{[식 12-31]}$$

이 커널 함수를 사용하면 3차원 벡터를 이용한 연산 없이 원래 차원인 2차원에서의 계산만으로 값이 얻어지므로 계산비용의 문제를 해결할 수 있다.
SVM에 고차원 매핑을 적용하는 경우, 입력 $\boldsymbol{x}$ 대신 $\phi(\boldsymbol{x})$를 입력으로 사용하여 계산하면 되므로, 파라미터 추정을 위한 라그랑지안 함수는 다음과 같이 주어진다.

$$L(\boldsymbol{w}, w_0, \boldsymbol{\alpha}) = \frac{1}{2} \| \boldsymbol{w} \|^2 - \sum_{i=1}^{N} \alpha_i \left\{ y_i \left(\boldsymbol{w}^T \phi(\boldsymbol{x_i}) + w_0 \right) - 1 \right\} \qquad \text{[식 12-32]}$$

이 라그랑지안 함수에는 고차원 벡터 $\phi(\boldsymbol{x})$가 주어지지만, 실제로 계산에 사용되는 이원적 문제의 함수 $Q(\boldsymbol{\alpha})$는 다음과 같이 커널 함수로만 표현 가능하다.

$$\begin{aligned} Q(\boldsymbol{\alpha}) &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \phi(\boldsymbol{x_i}) \cdot \phi(\boldsymbol{x_j}) \\ &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x_i}, \boldsymbol{x_j}) \end{aligned} \qquad \text{[식 12-33]}$$

마지막으로, 추정된 파라미터를 이용하여 분류 함수를 표현하면, 역시 커널 함수만으로 표현 가능함을 다음에서 확인할 수 있다.

$$\begin{aligned} f(\boldsymbol{x}) &= \operatorname{sign}\left(\sum_{i=1}^{N} \alpha_i y_i \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{x_i}) + w_0 \right) \\ &= \operatorname{sign}\left(\sum_{i=1}^{N} \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x_i}) + w_0 \right) \end{aligned} \qquad \text{[식 12-34]}$$

이상과 같이 고차원 매핑을 통해 비선형 문제를 선형화하여 해결하면서 커널 함수를 통해

계산량 증가의 문제를 해결하는 방법을 커널법이라고 하며, SVM을 비롯한 선형성을 가정하는 방법론에서 최근 활발히 사용되고 있다. SVM을 비롯한 여러 응용에서 주로 사용되는 몇 가지 커널을 정리하면 다음 표와 같다.

[표 12-1] 대표적인 커널 함수

선형 커널 (Linear Kernel)	$k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y})$
다항식 커널 (Polynomial Kernel)	$k(\boldsymbol{x}, \boldsymbol{y}) = (\boldsymbol{x} \cdot \boldsymbol{y} + c)^d$
시그모이드 커널 (Sigmoid Kernel)	$k(\boldsymbol{x}, \boldsymbol{y}) = \tanh(\theta_1 \boldsymbol{x} \cdot \boldsymbol{y} + \theta_2)$
가우시안 커널 (Gaussian Kernel)	$k(\boldsymbol{x}, \boldsymbol{y}) = \exp\left\{-\frac{\| \boldsymbol{x} - \boldsymbol{y} \|^2}{2\sigma^2}\right\}$

각 커널 함수는 고유의 파라미터(다항식 커널의 c와 d, 시그모이드 커널의 θ, 가우시안 커널의 σ)를 가지고 있으며, 이것은 문제의 성격에 맞추어 적절히 조정해 주어야 하는 하이퍼 파라미터이다.

슬랙변수와 커널을 가진 SVM에 대한 학습과 분류 단계를 정리하면 다음과 같다.

[SVM 분류기의 학습과 인식 단계]

① N개의 입출력 쌍으로 이루어진 학습 데이터 집합 $X=\{(\boldsymbol{x}_i, y_i)\}_{i=1\cdots N}$ 을 준비하고, 하이퍼 파라미터 c와 커널 함수 $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$를 정의한다. 이때 목표 출력값은 $y_i \in \{-1, 1\}$ $(i = 1, ..., N)$을 만족한다.

② 다음과 같은 과정을 통해 SVM을 학습한다.

②-1. 학습 데이터를 이용하여 파라미터 추정을 위한 목적함수 $Q(\boldsymbol{\alpha})$를 정의한다.

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad , \quad \sum_{i=1}^{N} \alpha_i y_i = 0,\ 0 \le \alpha_i < c\ (i = 1, ..., N)$$

②-2. 주어진 조건을 만족하면서 $Q(\boldsymbol{\alpha})$를 최소화하는 추정치 $\hat{\alpha}_i$를 이차계획법에 의해 찾는다.

②-3. $\hat{\alpha}_i \neq 0$이 되는 서포트벡터를 찾아 집합 $X_S = \{\boldsymbol{x}_i \in X \mid \hat{\alpha}_i \neq 0\}$ 를 생성한다.

②-4. $\hat{\alpha}_i$와 서포트벡터를 이용하여 $\hat{w}_0$를 계산한다.

$$\hat{w}_0 = \frac{1}{N_S} \sum_{\boldsymbol{x}_i \in X_S} \left(y_i - \sum_{\boldsymbol{x}_j \in X_S} \alpha_j y_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) \right)$$

이때 N_S는 집합 X_S의 원소의 수이다.

②-5. 서포트벡터 집합 $X_S = \{\boldsymbol{x}_i \in X \mid \hat{\alpha}_i \neq 0\}$ 와 파라미터 벡터 $\hat{\boldsymbol{\alpha}}$, 그리고 $\hat{w}_0$를 저장해 둔다.

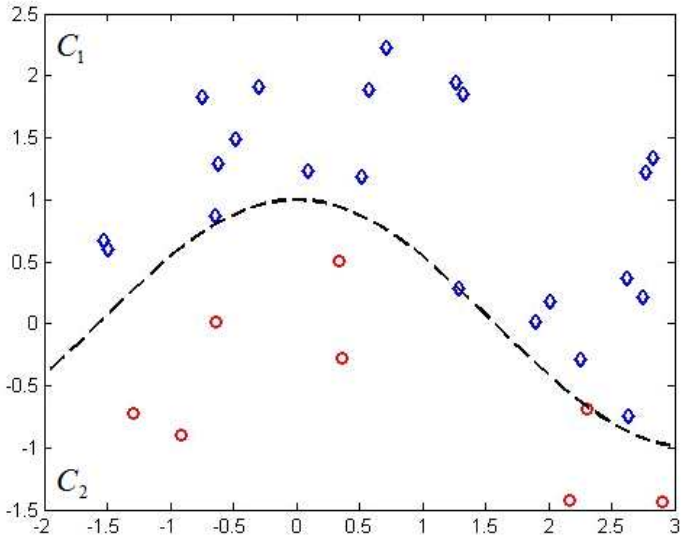
③ 새로운 데이터 $\boldsymbol{x}$가 주어지면, 저장해둔 서포트벡터와 파라미터를 이용하여 다음 판별함수로 분류를 수행한다.

$$f(\boldsymbol{x}) = \text{sign}\left(\sum_{\boldsymbol{x}_i \in X_S} \hat{\alpha}_i y_i k(\boldsymbol{x}_i, \boldsymbol{x}) + \hat{w}_0 \right)$$

12.5 매트랩을 이용한 실험

SVM을 학습하고 분류하는 프로그램을 매트랩으로 구현해 보겠다. SVM 학습에 사용될 데이터를 [그림 12-9]에 나타내었다. 2차원 입력공간에서 랜덤하게 선택된 데이터에 대하여, 그림에 점선으로 나타난 비선형 결정경계를 바탕으로 두 클래스로 나누었다. 결정경계의 위쪽

에 있는 데이터들을 클래스 C_1으로 할당하고, 목표 출력값을 1로 두었다. 반대로 아래쪽에 있는 데이터들은 클래스 C_2로 할당하고, 목표 출력값을 -1로 두었다. 데이터는 모두 30개이다.



[그림 12-9] SVM의 학습 데이터

주어진 학습 데이터를 분류하기 위한 SVM 분류기를 [프로그램 12-1]에 나타내었다. 전체적인 코드에 대하여 설명하기에 앞서, 프로그램에서 사용한 함수 quadprog()에 대하여 설명하겠다. 함수 quadprog은 매트랩에서 제공하는 최적화 함수로, 이차계획법에 의해 목적함수를 최소화하면서 특정 조건을 만족하는 파라미터를 찾는다. 최소화하는 목적함수는 [식 12-35]와 같이 정의하고, 만족해야 하는 조건은 [식 12-36], [식 12-37], [식 12-38]과 같이 정의한다.

$$J(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^T \mathrm{H} \boldsymbol{\alpha} + \boldsymbol{f}^T \boldsymbol{\alpha} \quad \text{[식 12-35]}$$

$$\mathrm{A}\boldsymbol{\alpha} < \boldsymbol{b} \quad \text{[식 12-36]}$$

$$\mathrm{A}_{eq}\boldsymbol{\alpha} = \boldsymbol{b}_{eq} \quad \text{[식 12-37]}$$

$$\boldsymbol{lb} \le \boldsymbol{\alpha} \le \boldsymbol{ub} \quad \text{[식 12-38]}$$

함수 quadprog는 이 목적함수와 조건의 구체적인 형태를 결정하는 행렬 $\mathrm{H}, \mathrm{A}, \mathrm{A}_{eq}$와 벡터 $\boldsymbol{f}$, $\boldsymbol{b}, \boldsymbol{b}_{eq}, \boldsymbol{lb}, \boldsymbol{ub}$를 입력으로 받아 최적화된 파라미터 $\boldsymbol{\alpha}$를 찾아 반환한다. 이 함수를 활용하기 위하여 [식 12-33]의 목적함수에 맞는 행렬 H와 $\boldsymbol{f}$를 찾으면 다음과 같다.

$$\mathrm{H} = \{y_i y_j k(\boldsymbol{x_i}, \boldsymbol{x_j})\}_{i=1\ldots N, j=1\ldots N} \quad \text{[식 12-39]}$$

$$\boldsymbol{f} = [-1, -1, \ldots, -1] \quad \text{[식 12-40]}$$

또한 조건식 [식 12-27]에 맞추어 입력 벡터를 설정하면, [식 12-36]에 해당하는 조건은 없

으므로 [식 12-37]과 [식 12-38]의 입력값이 다음과 같이 찾아질 수 있다.

$$\mathbf{A}_{eq}=[y_1,y_2,\ldots,y_N],\quad b_{eq}=0,\quad lb=\mathbf{0},\ ub=[c,\ldots,c] \qquad \text{[식 12-41]}$$

이상과 같이 결정된 입력값들을 함수 quadprog에 주어 최적화된 파라미터 α를 얻으면, 그 값이 0보다 큰 것을 선택함으로써 대응되는 서포트벡터를 찾을 수 있다.

프로그램 12-1-1 커널함수 SVM_kernel

입력행렬과 커널함수 선택 번호, 그리고 관련 파라미터를 받아서 계산된 커널 행렬 반환

```
001 function [out] = SVM_kernel(X1,X2,fmode,hyp)
002 % fmode  1: Linear Kernel
003 %        2: Polynomial Kernel
004 %        3: Gaussian Kernel
005 if (fmode==1)              % 선형 커널 (Linear Kernel)
006     out=X1*X2';
007 end
008 if (fmode==2)              % 다항식 커널 (Polynomial Kernel)
009     out = (X1*X2'+hyp(1)).^hyp(2);
010 end
011 if (fmode==3)              % 가우시안 커널 (Gaussian Kernel)
012     for i=1:size(X1,1)
013         for j=1:size(X2,1);
014             x=X1(i,:); y=X2(j,:);
015             out(i,j) = exp(-(x-y)*(x-y)'/(2*hyp(1)*hyp(1)));
016         end
017     end
018 end
```

또 한 가지 필요한 함수로, 커널 행렬을 계산하는 커널 함수가 있다. [프로그램 12-1-1]에 이를 위해 정의된 함수 SVM_kernel을 나타내었다. 함수 SVM_kernel은 주로 사용되는 세 가지 커널에 대하여 입력으로 주어진 데이터 행렬에 대한 커널함수의 값을 계산하여 행렬 형태로 반환한다. 이때 세 가지 커널 중 어떤 것을 사용할지 선택하기 위한 입력값(fmode)과 각 커널 함수에 관련된 파라미터도 함께 받는다.

프로그램 12-1 SVM 분류기

2차원 데이터 분류를 위해 커널을 사용한 SVM 분류기 생성 및 분류

```
001 %%% SVM for classification
002 load data12_8                          % 데이터 불러옴
003 N=size(Y,1);                           % N: 데이터 크기
004 kmode=2; hyp(1)=1.0; hyp(2)=2.0;       % 커널 함수와 파라미터 설정
005
006 %%%%% 이차계획법 목적함수 정의
007 %% 최소화 함수:   0.5*a'*H*a - f'*a
008 %% 만족 조건 1:   Aeq'*a = beq
009 %% 만족 범위:     lb <= a <= ub
010 KXX=SVM_kernel(X,X,kmode,hyp);          % 커널 행렬 계산
011 H=diag(Y)*KXX*diag(Y)+1e-10*eye(N);     % 최소화 함수의 행렬
012 f=(-1)*ones(N,1);                       % 최소화 함수의 벡터
013 Aeq=Y'; beq=0;                          % 만족 조건 1
014 lb=zeros(N,1); ub=zeros(N,1)+10^3;      % 만족 범위
015
016 %%%%% 이차계획법에 의한 최적화 함수 호출 --> 알파값, w0 추정
017 [alpha, fval, exitflag, output]=quadprog(H,f,[],[],Aeq,beq,lb,ub)
018 svi=find(abs(alpha)>10^(-3));            % alpha>0 인 서포트벡터 찾기
019 svx = X(svi,:);                         % 서포트벡터 저장
020 ksvx = SVM_kernel(svx,svx,kmode,hyp);    % 서포트벡터 커널 계산
021 for i=1:size(svi,1)                      % 바이어스(w0) 계산
022     svw(i)=Y(svi(i))-sum(alpha(svi).*Y(svi).*ksvx(:,i));
023 end
024 w0=mean(svw);
025
026 %%%%% 분류 결과 계산
027 for i=1:N
028     xt=X(i,:);
029     ksvxt = SVM_kernel(svx,xt,kmode,hyp);
030     fx(i,1)=sign(sum(alpha(svi).*Y(svi).*ksvxt)+w0);
031 end
032 Cerr= sum(abs(Y-fx))/(2*N);
033 %%%%% 결과 출력 및 저장
034 fprintf(1,'Number of Support Vector: %d₩n',size(svi,1));
035 fprintf(1,'Classificiation error rate: %.3f₩n',Cerr);
036 save SVMres12_8 X Y posId negId svi svx alpha w0 kmode
```

이제 전체적인 프로그램을 살펴보면, 먼저 데이터를 불러들이고, 커널 함수를 선택하고 관련 파라미터도 결정한다. 이어서 이차계획법에 의해 파라미터를 추정하기 위해 함수 quadprog

에 입력으로 줄 행렬들을 계산한다. 이때 함수 SVM_kernel을 호출하여 데이터의 커널 행렬도 찾게 된다. 함수 quadprog에 의해 α_i값이 추정되면, 그 값이 0보다 큰 것들을 찾아 서포트벡터를 찾게 된다. 찾아진 서포트벡터와 α_i을 이용하여 파라미터 w_0도 계산하면 판별함수를 위한 모든 파라미터 값들이 추정된 것이다. 추정된 파라미터를 이용하여 학습 데이터에 대한 분류를 수행하여 분류 오차를 계산한 후, 마지막으로 찾아진 서포트벡터의 개수와 오분류율을 출력하고 필요한 파라미터들을 저장해 둔다.

프로그램 12-2 결정경계의 그리기

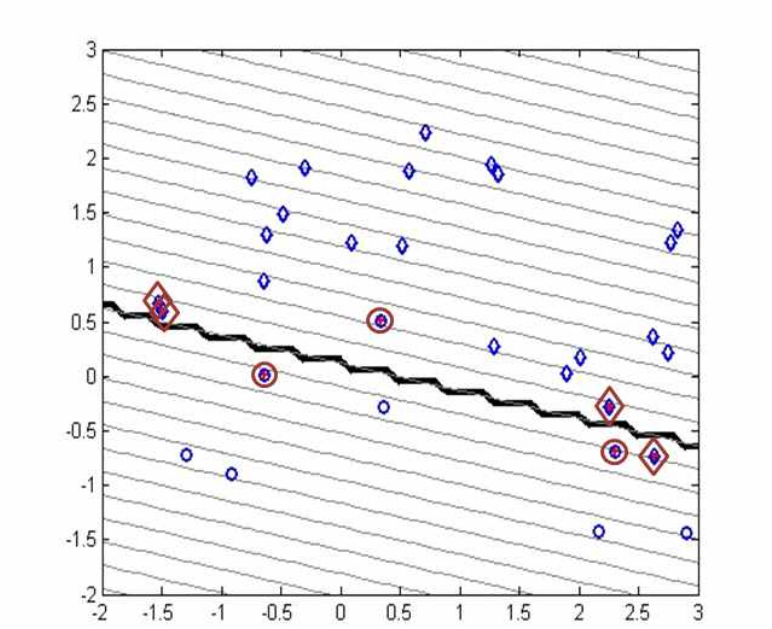
저장된 데이터와 파라미터를 불러와 결정경계를 그림

```
001 load SVMres12_8                          % 데이터와 파라미터 불러오기
002 xM = max(X); xm= min(X);                 % 결정경계 그릴 영역 설정
003 S1=[floor(xm(1)):0.1:ceil(xM(1))];
004 S2=[floor(xm(2)):0.1:ceil(xM(2))];
005 for i=1:size(S1,2)                       % 영역내의 각 입력에 대한 출력 계산
006     for j=1:size(S2,2)
007         xt=[S1(i),S2(j)];
008           ksvxt = SVM_kernel(svx,xt,kmode,hyp);
009           G(i,j)=(sum(alpha(svi).*Y(svi).*ksvxt)+w0);
010           F(i,j)=sign(G(i,j));
011     end
012 end
013 [X1, X2]=meshgrid(S1, S2);
014 figure(kmode); contour(X1',X2',F);          % 결정 경계 그리기
015 hold oncontour(X1',X2',G);                  % 판별함수의 등고선 그리기
016 posId=find(Y>0); negId=find(Y<0);
017 plot(X(posId,1), X(posId,2),'d'); hold on  % 데이터 그리기
018 plot(X(negId,1), X(negId,2),'o');
019 plot(X(svi,1), X(svi,2),'g+');              % 서포트벡터 표시
```

[프로그램 12-2]는 [프로그램 12-1]에서 저장해둔 정보를 불러와 입력 공간에 대한 결정경계와 서포트벡터를 그래프로 나타내는 프로그램이다. 입력 공간의 가능한 입력들에 대하여 판별함수를 계산하여 그 값을 바탕으로 결정경계를 그리고, α_i값들에 의해 찾아진 서포트벡터들도 함께 표시한다.

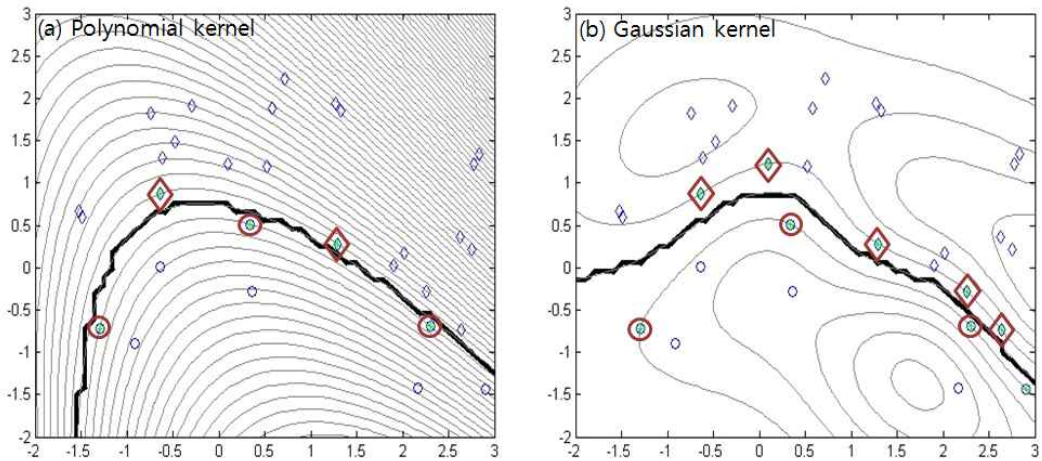
[그림 12-10]에는 선형 커널 함수, 즉 별도의 커널 함수를 사용하지 않는 경우에 얻어진 결정경계와 서포트벡터를 나타내었다. 굵은 선으로 표시된 결정경계가 함수 $g(\boldsymbol{x})=0$가 되는 곳이며, 그 경계선의 상하로 나타난 첫 번째 얇은 실선이 각각 $g(\boldsymbol{x})=1$과 $g(\boldsymbol{x})=-1$이 되는 선으로, 그 선상에 있는 것이 주된 서포트벡터가 된다. 또한 이 문제는 선형 불가능한 문제이므로, 찾아진 결정경계에 의해 제대로 분리되지 못한 데이터들, 즉 슬랙변수 ξ_i가 0이 아닌 데이터들로, 모두 서포트벡터에 해당한다. 찾아진 서포트벡터들도 그림에 표시하였다.

그림에서 모두 7개의 서포트벡터를 가짐을 알 수 있다.



[그림 12-10] 선형 커널 함수를 사용한 경우의 결정경계와 서포트벡터

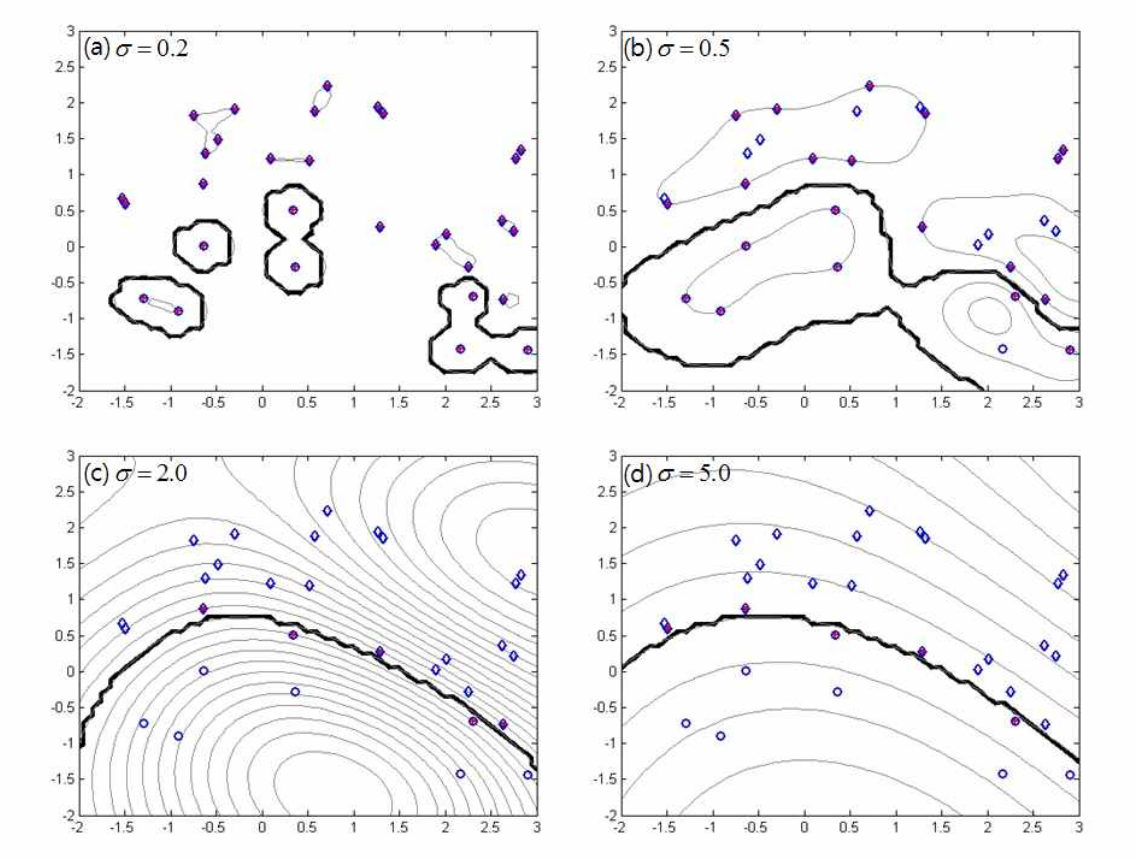
[그림 12-11]에는 다항식 커널과 가우시안 커널을 사용하여 얻어진 결정경계를 보여주고 있다. 다항식 커널의 경우 파라미터는 $b=1, d=2$를 사용하였으며, 가우시안 커널의 경우 $\sigma=1$을 사용하였다. 두 경우 모두 비선형 결정경계를 가지게 되어, 데이터를 모두 정확히 분류하였으며, 다항식 커널의 경우 5개의 서포트벡터를 가지고, 가우시안 커널의 경우 9개의 서포트벡터를 가짐을 볼 수 있다.



[그림 12-11] 다항식 커널과 가우시안 커널을 사용한 경우의 결정경계와 서포트벡터

이어서 [그림 12-12]는 가우시안 커널을 사용하되 파라미터 σ의 값을 달리하여 찾아지는 결

정경계의 변화를 살펴본 것이다. σ의 값, 즉 분산이 작은 가우시안 커널을 사용한 경우 복잡한 결정경계를 찾으며, σ의 값을 증가시키면 완만한 곡선 형태의 결정경계가 찾아짐을 확인할 수 있다. 특히 [그림 12-12a]와 같이 σ가 지나치게 작아지면 데이터에 과다적합된 결정경계를 찾고, 그 결과 모든 데이터가 서포트벡터의 역할을 하게 된다. 따라서 커널 함수의 파라미터를 적절히 조정해 주는 것이 필요할 것이다.



[그림 12-12] 가우시안 커널의 파라미터 값에 따른 결정경계의 변화

연습문제

1. [식 12-3]에 의해 정의되는 결정경계 $g(\boldsymbol{x}) = 0$에서 원점에 이르는 거리가 $w_0 / \| \boldsymbol{w} \|$ 가 되며, 입력공간 내의 임의의 한 점 $\boldsymbol{x}$에서 결정경계에 이르는 거리가 [식 12-4]와 같이 계산될 수 있음을 유도해 보시오.

2. 선형 분류기의 장단점과 커널법의 필요성에 대해 설명하시오.

3. 11장의 연습문제 1번과 같은 XOR문제를 위한 SVM을 설계해 보시오.

4. SVM은 기본적으로 두 개의 클래스에 대한 분류를 수행하는 이진 분류기이다. SVM을 이용하여 3개 이상의 클래스를 분류하는 다중 분류 문제를 풀고자 할 때, 어떻게 하면 좋을지 그 방법을 생각해 보시오.

5. 다층퍼셉트론에 의한 분류기와 SVM에 의한 분류기에 대하여 다음 사항들을 중심으로 그 특성을 비교해 보시오.
   (1) 결정경계의 형태
   (2) 학습의 목적
   (3) 학습과 인식에 필요한 계산량
   (4) 학습을 통해 찾아지는 최적해의 특성

6. 슬랙변수가 필요한 이유에 대해 설명하시오.

참고자료

최근 10여 년간 가장 관심을 모으고 있는 분류기인 SVM에 관한 연구는 [Vapnik 95]에서 시작되었다고 볼 수 있다. 이후 SVM과 커널법에 대한 매우 활발한 연구가 수행되어왔고, 관련 연구들을 소개하고 있는 교과서로 [Scholkopf & Smola 02]와 Shawe-Taylor & Cristianini 04]를 들 수 있겠다. 이밖에 간단한 튜토리얼 자료로 [Gunn 98]과 [Burges 98] 등이 있다. SVM과 커널법에 대한 다양한 소스 프로그램 및 전자 자료를 제공하고 있는 홈페이지들도 다수 존재하는데, 아래에 대표적인 몇 가지를 제시하였다.

[Vapnik 95] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, 1995.

[Scholkopf & Smola 02] B. Scholkopf and A. J. Smola, Learning with Kernels, MIT Press. 2002.

[Shawe-Taylor & Cristianini 04] J. Shawe-Taylor and N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004

[Gunn 98] Support Vector Machines for Classification and Regression, Technical Report, Faculty of Engineering, Science and Mathematics, School of Electronics and Computer Science, University of Southampton, 1998.

[Burges 98] C. J. C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery 2:121–167, 1998

www.kernel-machines.org

www.support-vector-machines.org

SVM matlab toolbox (http://asi.insa-rouen.fr/enseignants/~arakotom/toolbox/index.html)