워크북

교과목명 : 머신 러닝

차시명: 6차시

◈ 담당교수: **장 필 훈**

⊙ 세부목차

- 커널방법론
- 듀얼표현
- 가우시안 프로세스
- SVM
- 클래스 분포간 중첩

학습에 앞서

■ 학습개요

입력 데이터포인트들을 변환하여 다른 공간에서 해결한다는 아이디어에 관해 배우고, 구체적으로 커널이 어떤 역할을 하는지 수식으로 확인하게 된다. 이 과정에서 듀얼표현과 그 풀이과정을 본다. 희박한 커널머신의 필요성을 알게 되면, 그 일환으로 SVM을 보게 된다. 기본적인 아이디어부터 공식화, 푸는 과정등을 관찰하고 더 나아가 slack variable까지 도입한 수식의 전개를 이해한다. 다음시간에도 SVM을 이어서 배우게 되므로 그 기초과정이 된다.

■ 학습목표

1	커널의 개념을 이해하고 커널을 쓰는 이유, 구체적인 방법에 대해 배운다.
2	듀얼표현을 커널의 개념과 함께 이해하고 푸는 방법을 배운다.

3	SVM 의 기본 아이디어, 공식화를 배우고 slack variable 까지 포함한 analysis 를 익힌다.
4	가우시안 프로세스의 과정을 개념만 이해하고, 실제 프로그램의 결과를 본다.

■ 주요용어

용어	해설
커널	한 공간에서 다른 공간으로 데이터포인트들을 이동시킬 수 있고, 새로운 공간에서의 조작이 원 공간에서 충분히 유의미할 때, 그 이동하는 함수를 커널이라고 부른다. 당연히, 아무런 변환이나 되는 것은 아니고, 유효한 커널의 조건이 존재한다.
듀얼표현	입력 데이터포인트들에 커널을 적용한 뒤 다른 공간에서 성립하는 식을 얻어낼 수 있는데, 이것을 듀얼 표현(듀얼 공식화)라고 한다. 이 두 식은 각 공간에서의 관점을 나타낸 것일 뿐 본질적으로 동일하며, 한쪽에서 푼 결과가 다른쪽에서도 유효하다.
서포트 벡터	SVM 에서 선형분리 가능한 집합의 경계부분 마진을 최대화하려고 하면, 필연적으로 경계부분에 있는 몇 개의 데이터 포인트만이 중요하고 나머지는 다 버려도 되는데, 이 경계부분에서 마진 형성에 영향을 주는 데이터포인트들을 서포트벡터라고 한다. 데이터포인트는 모두 벡터로 볼 수 있기 때문에 이렇게 부른다.
slack variable	완전히 선형분리가 가능하지 않더라도 약간의 오분류 가능성을 주면 SVM 을 적용할 수 있는데, 어느정도 오분류를 허용할 것인가를 결정하는 변수를 slack variable 이라고 한다.

학습하기

<커널방법론>

지금까는 모델이 훈련집합과 새로운 입력을 분리해서 사용했습니다. 이것은 어떻게 보면 당연해 보이기도 합니다. 하지만 만약 우리의 모델이 충분한 메모리를 가지고 있어서 훈련집합 전부를 기억할 수 있다면 어떨까요. 예를 들어 24x24이미지에서 사람의 얼굴을 검출해 내야 하는데, 아예 가능한 모든 이미지 모양을 기억해 놓고 충분이 빠른 시간 안에 검색해서 입력과 대조해볼 수 있다면 복잡한 네트워크가 필요하지 않을 것입니다. 하지만 현실적으로 그럴 수가 없기 때문에(8비트 그레이 이미지로 24*24사이즈에 가능한 모든 경우의 이미지를 저장하는데 얼마만큼의 저장공간이 필요할까요?) 우리의 모델은 추상성을 획득하여 다양한 입력에 대응할 수 있어야 합니다. 하지만 훈련집합 그 자체를 이용하면서도 메모리 방식이 아닌 것이 있는데, 그것이 바로 지금부터 배울 커널 함수를 이용하는 방식입니다. 이것은 개념적인 설명을 위한 것이고, 훈련집합 그 자체를 이용하느냐 하지 않느냐, 한다면 어떤 식이냐는 것은 사실 전혀 중요하지 않습니다. 그래서 크게 신경쓰지 않아도 됩니다. 분류를 위

한 것일 뿐 방법은 언제나 다양한 것이고, 무엇이 어떤식으로 동작하는지만 이해하고 있으면 됩니다.

일단 선형회귀모델의 제곱합 오류함수부터 시작해보겠습니다.

$$J(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) - t_n\right\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \qquad \boldsymbol{\lambda \geq 0}$$

여기서 phi는 기저함수, lambda는 정규화계수, w는 weight vector, t_n은 target value를 합니다.

오류함수를 최소화하는 w를 구하기 위해 dJ(w)/dw=0으로 두고 w에 대해 풀면 다음을 얻을 수 있습니다.

$$\mathbf{w} = -\frac{1}{\lambda}\sum_{n=1}^{N}\left\{\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) - t_n\right\}\boldsymbol{\phi}(\mathbf{x}_n) = \sum_{n=1}^{N} a_n\boldsymbol{\phi}(\mathbf{x}_n) = \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{a}$$

$$\text{where} \qquad a_n = -\frac{1}{\lambda}\left\{\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) - t_n\right\}.$$

이제 여기서 얻은 a를 사용해서 제곱합 오류함수를 다시 적을 수 있습니다.

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{a}$$

이것을 듀얼표현이라고 합니다. 여기서 $\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}$를 그램행렬이라고 합니다.

원소 $K_{nm} = \boldsymbol{\Phi}(x_n)^T\boldsymbol{\Phi}(x_m)$인 행렬 K를 정의하고 다시 J를 적으면 아래와 같고,

$$J(\mathbf{a}) = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{K}\mathbf{a} - \mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{t} + \frac{1}{2}\mathbf{t}^{\mathrm{T}}\mathbf{t} + \frac{\lambda}{2}\mathbf{a}^{\mathrm{T}}\mathbf{K}\mathbf{a}.$$

J를 미분하여 0으로 두고 얻은 해는 다음과 같습니다.

$$\mathbf{a} = (\mathbf{K} + \lambda\mathbf{I}_N)^{-1}\mathbf{t}.$$

(계산을 모두 생략했으므로 이해가 안가더라도 일단 계속 수업진도를 따라가면 됩니다)

이것을 새로운 회귀모델에 대입해서 새로운 예측치를 얻을 수 있습니다.

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) = \mathbf{a}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\phi}(\mathbf{x}) = \mathbf{k}(\mathbf{x})^{\mathrm{T}}\left(\mathbf{K} + \lambda\mathbf{I}_N\right)^{-1}\mathbf{t}$$

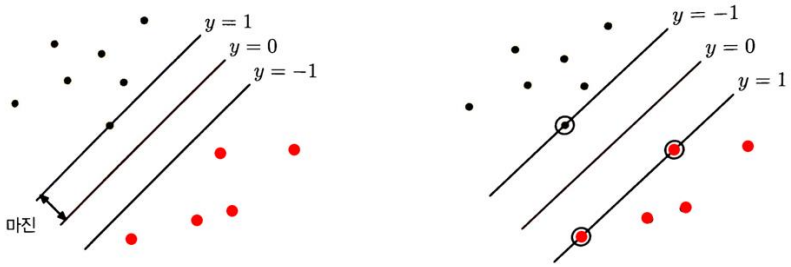
여기서 $\mathbf{k}(\mathbf{x}) = k_n(\mathbf{x}) = k(\mathbf{x}_n, \mathbf{x}) = \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})$입니다.

계산과정보다 훨씬 더 주목해야 할 점은, 해를 온전히 커널함수에 관해 표현할수 있다는 것입니다. 이것을 듀얼공식화라고 합니다. 이렇게 하면, 커널을 이용해 데이터포인트를 변환한 공간을 직접적으로 다루지 않아도 해를 얻어낼 수 있습니다. 다시말하면, 현재공간에서 해가 존재하지 않더라도 적절한 커널을 선택하여 다른 공간으로 옮기고 나면 해결 가능한 경우가 있을 수 있는데, 그 다른차원의 공간에서 모든 계산을 수행할 필요가 없기 때문에 고차원공간을 다룰 필요가 없다는 뜻입니다. 그저 식에 따라 계산만 하면 됩니다.

커널의 조건도 상당히 느슨한 편이어서, 커널을 찾기 위해 특별히 많은 노력을 기울이지 않아도 됩니다. 커널이 만족해야 할 필요충분 조건도 알려져 있고, 알려진 커널들을 조합해서 새로운 커널을 만들 수도 있습니다.

<SVM>

커널에 기반한 모든 알고리즘은 커널함수의 값을 훈련집합의 모든 데이터포인트 쌍에 대해 계산해야 하는 단점이 있습니다($k(x_m, x_n)$ for all m, n). 따라서 더 희박한 해를 가지는 방법에 대한 연구가 이루어졌고 그 대표적인 결과가 SVM(최대마진 분류기)입니다.



Bishop, Christopher M. Pattern Recognition and Machine Learning. New York :Springer, 2006. fig7.1

위 그림을 먼저 살펴보겠습니다. 왼쪽과 오른쪽 두 분류기는 모두 검은점과 빨간점을 성공적으로 분류해냅니다. 하지만 자세히 살펴 보면 오른쪽의 경계(y=0)가 훈련셋의 가운데에 가깝습니다. 다시말해, 선형분리 가능한 두 class 의 가운데 중립(?)적인 공간을 정확히 반으로 가릅니다. 왼쪽 그림에서 마진을 최대화 한다는 뜻입니다.

선형분리를 가정했을 때, y(x)=0에 해당하는 초 평면으로부터 데이터포인트 x까지 수직거리는 $\frac{|y(x)|}{\|w\|}$ 이고, 모두 올바르게 분류된 경우 $t_n y(x_n) > 0$ 입니다. ($t_n = -1 \; or \; 1$) 따라서 x_n으로부터 결정표면까지 거리는 다음과 같이 나타낼 수 있습니다.

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b)}{\|\mathbf{w}\|}.$$

따라서 최대마진해는 다음과 같이 적을 수 있습니다.

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[t_n \left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b \right) \right] \right\}$$

마진은 오직 가장 가까운 데이터포인트에만 영향을 받기 때문에 식 안에 min이 있음을 유의하세요.

위 식에서 w와 b에 상수배를 해도 데이터포인트로부터 결정경계까지 거리가 변하지 않음을 볼 수 있습니다. 따라서 표면에 가장 가까운 데이터포인트 x에 대해 다음과 같이 둘 수 있습니다.

$$t_n \left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b \right) = 1$$

따라서 모든 점이 다음을 만족합니다. (위의 1이 최솟값이기 때문입니다)

$$t_n \left(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b \right) \geqslant 1, \qquad n = 1, \ldots, N.$$

이 조건을 만족시키면서 $\|w\|^2$를 최소화하면, 최대마진해를 구하게 됩니다. 따라서 다음과 같습니다.

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$ ……(식4)

제약조건 있는 최적화문제를 보면 이제 자연스럽게 라그랑주 승수법이 떠오르실 것입니다. 라그랑지안은 다음과 같습니다.

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{ t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) - 1 \right\}$$

제약조건마다 하나의 승수(a_n)를 걸고 식을 만든 것입니다. 이제 라그랑지안의 w에 대한 미분을 0으로 두고 하나, b에 대한 미분을 0으로 두고 하나의 식을 얻습니다. 여기서부터는 라그랑지안을 푸는 방법일 뿐이므로 이해가 잘 가지 않으면 라그랑주 승수법 예제문제를 찾아보시면 됩니다.

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n) \quad , \quad 0 = \sum_{n=1}^{N} a_n t_n$$ ……(식1)

위 두 식을 이용해서 w를 구하고 L에 대입하면 최대마진문제의 듀얼표현을 얻습니다.

$$\widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

이 식을 a에 대해 최대화합니다.

최대마진해를 구한 뒤에 새 데이터포인트를 분류할때는 아래 식에 대입하면 됩니다.

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b.$$ ……(식2)

처음에 가정한 선형분류식($\mathbf{y(x) = w^T\phi(x) + b}$)에 식1의 w를 대입한것입니다. 이때 KKT조건에 의해 $a_n\{t_n y(x_n) - 1\} = 0$이어야 합니다. a_n=0이라면 y(x)에 반영되지 않으므로 $t_n y(x_n) = 1$(식(3)) 입니다. 이 조건을 만족하는 데이터포인트를 support vector라고 합니다. SVM은 한번 모델이 훈련되면 이 서포트벡터만 중요하고 나머지 데이터포인트는 예측이 쓰이지 않습니다. 최대마진을 구하는 공간은 커널을 한번 통과한 특징공간이라는 점에 유의하세요. b는 식2를 식3에 대입하여 구합니다.

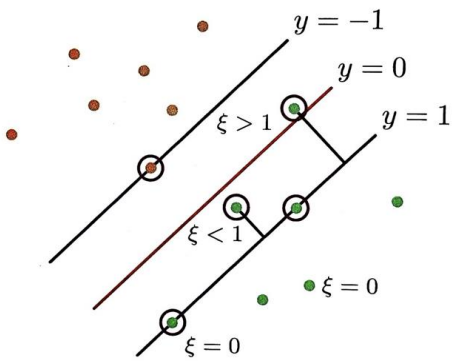
지금까지는 특징공간상에서 선형분리 가능하다는 것을 전제로 식을 전개했습니다. 그런데 만약 선형분리 가능하지 않다면 어떻게 할까요. 선형분리 가능한 다른 특징공간을 찾는 방법도 있겠지만, 클래스간 중첩을 허용하는 방법도 있습니다. 몇몇 포인트의 오분류를 허용하는 것입니다

slack variable(ξ)을 도입해서 이 문제를 해결합니다. 오분류에 대한 불이익을 경계면으로부터 거리에 대한 선형함수로 만드는 것입니다. 만일 마진경계에 포인트가 존재한다면 $\xi = 1$입니다. 그 외의 경우는 $\xi_n = |t_n - y(x_n)|$이 됩니다. 만일 오분류된다면 $\xi > 1$이 됩니다. 마진경계와 결정경계의 거리가 1

이기 때문입니다. 이 경우 분류제약조건은 다음과 같습니다.

$$t_n y(\mathbf{x}_n) \geq 1 - \xi_n, \qquad \xi_n \geq 0$$

정리하면, (1) $\xi = 0$: 올바르게 분류됨 (2) $0 < \xi \leq 1$: 마진 내부에 존재하지만 올바르게 분류됨 (3) $\xi > 1$: 오분류, 가 됩니다. 이를 소프트마진 제약조건이라고 부르기도 합니다. 그림으로 보면 다음과 같습니다



Bishop, Christopher M. Pattern Recognition and Machine Learning. New York :Springer, 2006. fig7.3

오분류에러가 크시(ξ)에 대해 선형이라서 아웃라이어에 민감합니다. 우리의 목표는 식4에 slack variable을 추가한 형태가 됩니다.

$$C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|\mathbf{w}\|^2 \qquad \cdots\cdots\cdots\cdots\cdots\cdots\text{(식5)}$$

여기서 C는 상수로서 둘 사이의 균형을 조절합니다. 정규화계수가 하는 역할을 하는 것입니다. C가 무한대가 된다면(크시가 0이 되어야 하므로) 원래 SVM을 다시 얻습니다.

식5도 위에서 했던것과 똑같이 라그랑주 승수법으로 풉니다.

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n - \sum_{n=1}^{N}a_n\{t_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^{N}\mu_n\xi_n$$

KKT조건들$\left(\frac{\partial L}{\partial \mathbf{w}} = \frac{\partial L}{\partial b} = \frac{\partial L}{\partial \xi_n} = 0\right)$을 이용해서 다음 듀얼 라그랑지안을 얻고 그것을 풀면 됩니다.

$$\widetilde{L}(\mathbf{a}) = \sum_{n=1}^{N}a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

다음시간에는 이것을 푸는 지점부터 이어서 하겠습니다.

우리가 일일이 식으로 다 전개해보았지만 여러분이 관련분야의 학자가 되지 않는 한 실제로 이 식을 그대로 이용할 일은 아마 앞으로 거의 없을 것입니다. 그래도 라이브러리의 내부가 어떤식으로 동작하는 것인지, 어떤 원리에 의한것인지를 알아두면 현업에도 언젠가 도움이 될 날이 있을것이라고 생각합니다.

연습문제

1. K-nearest neighbor방식은, 훈련데이터의 일부를 예측단계에도 사용한다.

   O : 본질적으로 훈련데이터의 일부를 사용하느냐 하지 않느냐는 의미없는 논쟁일 수 있다. training은 기본적으로 데이터포인트들로부터 시작하는 것이고 학습은 그 추상화의 결과물이므로 데이터의 일부가 예측단계에서 '어떻게'사용되는 것이 '사용하는'것이냐 정의하기 나름이기 때문이다. 그러나 메모리베이스 방식은 대개 훈련데이터의 일부를 사용하는 부류로 분류한다.

2. SVM도 훈련데이터의 일부를 예측단계에 사용한다

   O : SVM의 경우 서포트벡터를 찾아내서 모델에 그대로 반영하므로, 더 이견이 없이 '훈련데이터의 일부를 사용하는' 방식에 속한다.

3. 커널함수를 이용한 듀얼표현은 본질적으로 같은 식을 다른 방식으로 표현한 것에 불과하다.

   O : 맞는 설명.

4. 듀얼문제를 푸는 것이 계산상의 이득이 있다.

   X : 보통 고차원인 특징공간을 직접 다루지 않아도 되는 장점이 있어서 사용하고, 계산상으로는 보통 더 불리하다.

5. 어떤 커널이 유효한 커널이면, 그 커널의 상수배나 exp함수를 적용한 뒤에도 유효하다.

   O : 맞는 설명

6. 가우시안 프로세스는 함수의 예측값의 분포가 아니라 함수의 분포 자체를 추정하는 것이 기본 아이디어다.

   O : 맞는 설명. 다만, 본질적으로 그 결과가 다르다고 보기는 힘들 수도 있다. 예측값의 분포들도 결국 함수의 분포라고 볼 수도 있기 때문이다.

7. SVM의 경우 해석적 해를 얻을 수 없다.

   X : 라그랑주 승수법이나 KKT조건들을 이용해서 해석적인 해를 구할 수 있고 실제 라이브러리도 그런식으로 동작한다.

정리하기

1. 훈련데이터의 일부 혹은 전부를 예측단계에도 사용할 수 있다. 메모리베이스 방식이 모두 여기에 해당하고, 커널함수를 이용하는 방식도 여기에 속한다
2. 데이터들을 다른차원으로 비선형변환 후 우리가 원하는 분류를 수행하기 위해 사용하는 것이 커널함수.
3. 원래의 오류함수를 커널함수로 나타낼 수 있다. 이때 커널함수로 나타낸 데이터포인트들은 변형된 다른 차원에서의 관점이고, 원래 식은 원래 데이터포인트들이 속했던 차원의 관점이다. 둘은

동일하며 표현만 다른 것이다.

4. 계산상의 이점이 없더라도 높은차원의 특징공간을 직접 다루지 않아도 되는 장점이 있다.
5. 가우시안 프로세스는 함수들의 분포를 추측해내는 과정이다. 매개변수모델로부터 시작하는 것이 아니다. 가우시안 프로세스의 과정에서도 자연스럽게 커널함수를 관찰할 수 있다.
6. SVM은 경계부분의 마진을 최대화하는 것이 기본 아이디어.
7. 식을 세우고 풀면, 서포트벡터만이 중요하다는 결론을 얻는다.(나머지는 버려짐)
8. 선형분리가 완전히 가능하지 않더라도 slack variable을 도입해서 해결할 수 있다.

참고하기

Bishop, C. M. "Bishop-Pattern Recognition and Machine Learning-Springer 2006." Antimicrob. Agents Chemother (2014): 03728-14.

다음 차시 예고

- soft margin SVM
- multiclass SVM
- SVM을 이용한 회귀
- 상관벡터머신
- 그래프모델
- 베이지안 네트워크
- 조건부독립