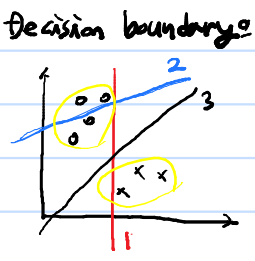
Support Vector Machine, SVM

- 분류 문제를 푸는 모델 중 하나.
- decision boundary의 margin을 크게 해서 일반화 능력을 최대화할 수 있는 방법



1 → model이 명확한 경향을 알지만 decision boundary와 가까워서 새로운 model이 나왔을 때 틀릴 수도 있음

3 → decision boundary와 가까운 점들이 없어 일반화 성능이 우수.

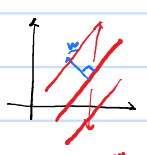
선형 SVM

- 선형 SVM에서 사용하는 model은 일반적인 선형 분류와 같음.
- 최적화 알고리즘이 다르다.

$$d(\underline{x}) = w_0 + w_1x_1 + \cdots + w_dx_d = \underline{w}^T\underline{x} + w_0 = 0$$

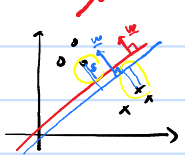
초평면의 방정식

- $\underline{x} = (x_1, \cdots, x_d)^T$, $\underline{w} = (w_1, \cdots, w_d)^T$
- $d(\underline{x})$는 특징공간을 2개의 부분공간으로 분할. 2개의 값을 줌
- 초평면의 법선에 상수 c를 곱해도, 수학적으로 동일한 초평면이다.
- $\underline{w}$는 초평면에 수직인 법선벡터이고, bias w_0는 위치를 결정한다.



⇒ 이때 점 $\underline{x}$에서 초평면까지의 거리 $h = \frac{|d(\underline{x})|}{\|\underline{w}\|_2}$

⇒ 선형 SVM에서는, 초평면의 법선 $\underline{w}$가 정해지면 그에 따라 bias w_0가 정해진다. 이때, margin을 최대화하는 w_0로 정한다.



* 가운데 가까운 Data point와의 거리를 재서 margin을 최대화하는 w_0를 정함

선형 SVM 문제 1.

- Margin이 가운데 넓은 $\underline{w}$를 찾기 (s = 가운데까지의 거리)

$$\Rightarrow \text{Margin} = 2s = \frac{2|d(\underline{x})|}{\|\underline{w}\|_2} = \frac{2}{\|\underline{w}\|_2}$$

초평면의 법선에 상수 c를 곱해도, 수학적으로 동일한 초평면이다.
⇒ $|d(\underline{x})| = 1$인 $\underline{w}$를 찾음
↳ 가운데까지의 거리 1

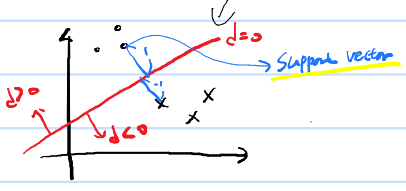
선형 SVM 문제2

- 최적화 문제로 정식화하면 다음과 같음.

(비용함수) Cost function ⇒ maximize $J(\underline{w}) = \frac{2}{\|\underline{w}\|_2}$ ⇒ 잘 분류하게 만들기 위한 목표 함수
"얼마나 분류"

(제약조건) constraints ⇒ $\underline{w}^T\underline{x}_i + w_0 \geq 1$, $\forall y_i = 1$ (label)
"부호가 분류"
$\underline{w}^T\underline{x}_i + w_0 < -1$, $\forall y_i = -1$ (label)
⇒ 모든 샘플을 만족하는 $\underline{w}$를 찾는 중. (만족의 정도에 따라 알고리즘에서 해로운 샘플 존재)

⇒ Cost function과 constraints를 선형계획법에 사용



선형 SVM 문제3.

Cost function ⇒ minimize $J(\underline{w}) = \frac{\|\underline{w}\|_2^2}{2}$

constraints ⇒ $y_i(\underline{w}^T\underline{x}_i + w_0) - 1 \geq 0$, $(i = 1, 2, \cdots, n)$

- 제약조건에서 등호를 만족하는 sample $(\underline{x}_i, y_i)$가 support vector이다.
- Support vector는 decision boundary에서의 거리가 1이다. 나머지 sample들은 거리가 1보다 멀리 떨어져 있다.
- 이와 같은 형태의 최적화 문제를 조건부 최적화, Conditional Optimization 문제라고 부름. (조건을 만족하는 범위에서 최적화)
- 조건부 최적화 문제를 푸는 방법 중 하나로, 라그랑주 승수법을 사용.

Lagrange Multiplier (제약조건을 비용함수에 녹여내기)

- 라그랑주 함수

$$\Rightarrow L(\underline{w}, w_0, \underline{\alpha}) = \underbrace{\frac{\|\underline{w}\|_2^2}{2}}_{\text{기존 비용함수}} - \sum_{i=1}^{n} \alpha_i \underbrace{(y_i(\underline{w}^T\underline{x}_i + w_0) - 1)}_{\text{제약조건 항}}$$

$J(\underline{w})$ / 라그랑주 승수 (제약조건 보이지 않는 변수 생성) / constraints

⇒ 구속조건의 최적화를 푸는 것과 동일

- 제약조건을 만족하는 sample에 대해, $\alpha_i = 0$이면 기존 비용함수 항만 남음.
- 제약조건을 위배하는 sample에 대해 $\alpha_i > 0$이라면 $-\alpha_i(y_i(\underline{w}^T\underline{x}_i + w_0) - 1)$이므로, 제약조건항은 양수이고, 라그랑주 함수의 값은 커짐.

라그랑주 승수: 비용함수 (라그랑주 승수를 추가로 도입)
제약조건 항: 위배하는 sample에 대해 양수 ($\alpha_i > 0$)
비선형조건을 위배하는 sample들을 준다

⇒ 제약조건을 만족하는 점 = $\alpha_i = 0$

* 제약조건 만족 ⇒ $\alpha_i = 0$
제약조건 위배 ⇒ $\alpha_i > 0$
⇒ 제약조건은 비용함수에 넣을 수 있음.

⇒ 제약조건을 무조건 만족하려고 하지만 비용함수에서는 비용함수 값이 커지더라도 최적화하고 내버릴 수 있음

↳ 수학적으로 제약조건을 위배하는 점을 최대한 배제하면서 기존 비용함수를 최적화

라그랑주 승수법 기본개념

- 라그랑주 함수 L을 새로운 비용함수로 생각할 수 있음
- 라그랑주 함수에 대한 Karush-Kuhn-Tucker (KKT) 조건이 추가됨

KKT conditions

$$\Rightarrow \frac{\partial L}{\partial \underline{\theta}} = \underline{0}, \quad \alpha_i \geq 0, \quad \alpha_i f_i(\underline{\theta}) = 0, \quad (i = 1, 2, \cdots, n)$$

($\underline{\theta}$: $\underline{w}, w_0$ / 0벡터)

- 부등식 제약조건이 $f_i(\underline{\theta}) \geq 0$의 형태로 주어지는 경우 → Data sample의 제약조건

SVM의 라그랑주 승수법 적용

* $\frac{\partial L(\underline{w}, w_0, \underline{\alpha})}{\partial \underline{w}} = \underline{0} \longrightarrow \underline{w} = \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \underline{x}_i$ ⇒ $L(\underline{w}, w_0, \underline{\alpha}) = \frac{\|\underline{w}\|_2^2}{2} - \sum_{i=1}^{n} \alpha_i (y_i(\underline{w}^T\underline{x}_i + w_0) - 1)$

⇒ $\frac{\partial L}{\partial \underline{w}} = \frac{1}{2} \cdot 2 \cdot \underline{w} - \sum_{i=1}^{n} \alpha_i y_i \underline{x}_i = 0$ ⇒ $\underline{w} = \sum_{i=1}^{n} \alpha_i \cdot y_i \cdot \underline{x}_i$ — (1)

⇒ $\frac{\partial L}{\partial w_0} = \sum_{i=1}^{n} \alpha_i y_i = 0$ — (2)

* $\alpha_i \geq 0$ — (3)

* $\alpha_i(y_i(\underline{w}^T\underline{x} + w_0) - 1) = 0$ — (4)

- Cost function (Lagrange function)

* maximize w.r.t α_i : $L(\underline{w}, w_0, \underline{\alpha})$

constraints

$\underline{w} = \sum_{i=1}^{n} \alpha_i y_i \underline{x}_i$, $\sum_{i=1}^{n} \alpha_i y_i = 0$, $\alpha_i \geq 0$ — 선형계획 제약조건

(1), (4)

- Cost function

* maximize w.r.t $\underline{\alpha}$: $\tilde{L}(\underline{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\underline{x}_i \cdot \underline{x}_j)$

kernel trick (내용 공부해)

constraints

$\sum_{i=1}^{n} \alpha_i y_i = 0$, $\alpha_i \geq 0$, $(i = 1, 2, \cdots, n)$

* 볼록한 함수 : convex → 수학적으로 최적해를 찾기 쉬워짐.

- 최적화 문제 분석
  - 비용함수는 $\underline{\alpha}$에 대한 2차함수이므로, 볼록함수 (convex)이고, 전역 최적해를 찾을 수 있음.

SVM 예측.

- 선형 SVM의 예측모델

$$d(\underline{x}) = \underline{w}^T\underline{x} + w_0, \quad y_{pred} = \begin{cases} 1 & d(\underline{x}) > 0 \\ -1 & \text{otherwise.} \end{cases}$$