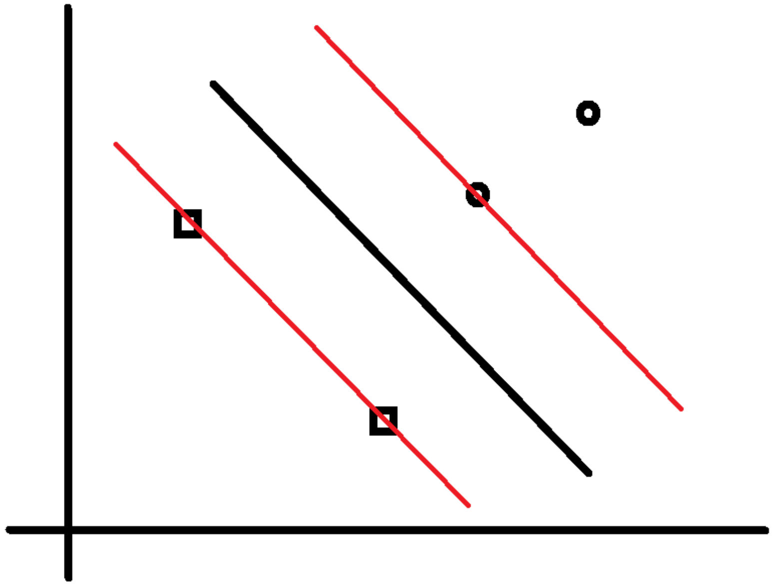
SVM (Support Vector Machine)

"두 집단을 구분하는 가장 넓은 도로"



하드 마진

- 모든 샘플이 도로 바깥쪽에 위치
- 이상치에 민감

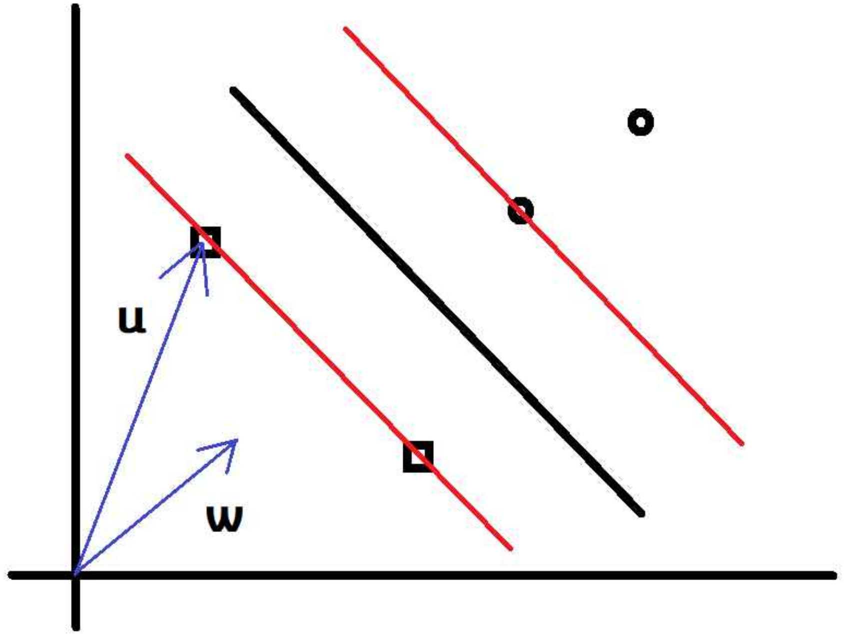
소프트 마진

- 마진오류를 일부 허용
- 과대적합을 방지

하드 마진 Idea

결정 경계와 수직인 어떤 벡터 w

어떤 점에 대한 위치벡터 u



u를 w에 정사영한 벡터가 결정경계를

(넘는다면 / 넘지 않는다면)

(positive / negative)로 분류

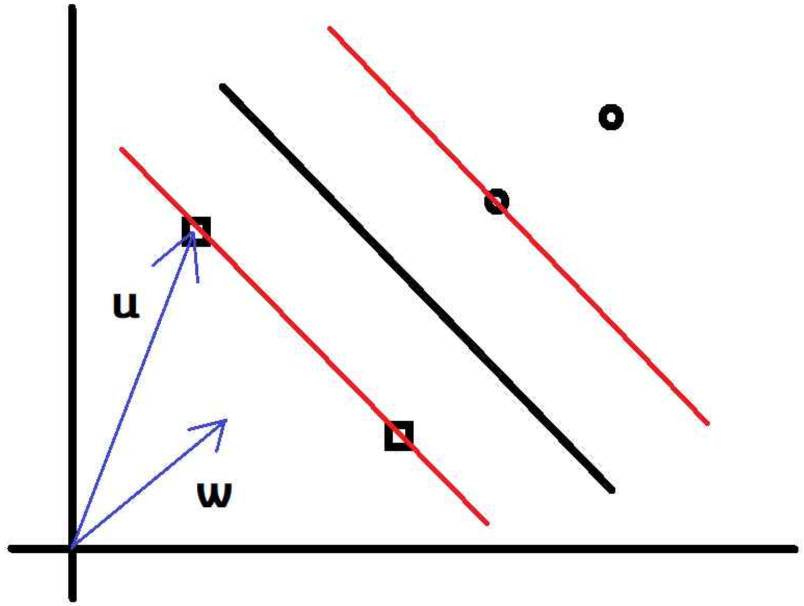
$w \bullet u \geq c$ 이면 positive

$\Rightarrow w \bullet u + b \geq 0$ 이면 positive "Decision Rule"

목표: 마진을 가장 크게 하는 w와 b 찾기

w: 결정경계의 방향 결정, b: 결정경계의 위치 결정

마진 결정 (도로의 양 끝선)



positive: $w \bullet x_+ + b \geq 1$

negative: $w \bullet x_- + b \leq -1$

t_i 정의: I번째 샘플이

positive 일 때 1, negative 일 때 −1

$$t_i(w \bullet x_i + b) - 1 \geq 0$$

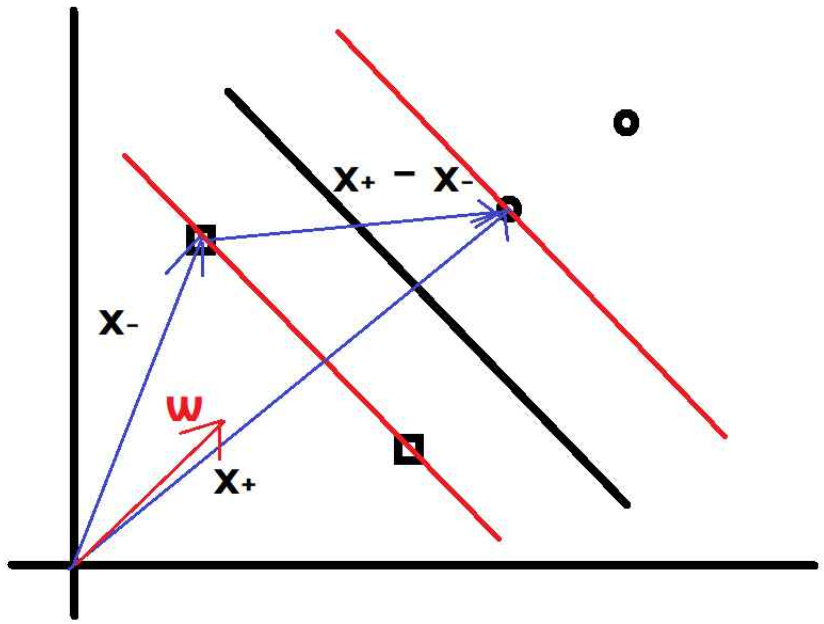
도로의 양 끝에 위치한 <u>서포트 벡터</u>의 경우

$$t_i(w \bullet x_i + b) - 1 = 0$$

마진의 너비

$x_+ - x_-$ 벡터를 w 벡터에 정사영한 벡터의 길이

$$(x_+ - x_-) \bullet \frac{w}{\| w \|}$$



양 서포트 벡터 x_+, x_-에 대해

$t_i(w \bullet x_i + b) - 1 = 0$ 에서

positive: $t_i = 1$이고, $w \bullet x_+ = 1 - b$ / negative: $t_i = -1$이고, $w \bullet x_- = 1 + b$

$$(x_+ - x_-) \bullet \frac{w}{\| w \|} = \frac{1}{\| w \|}(1 - b + 1 + b) = \frac{2}{\| w \|}$$

SVM 목표: 마진을 최대화

$\sim \quad \dfrac{2}{\| w \|}$ 최대화 $\quad \sim \quad \| w \|$ 최소화 $\quad \sim \quad \dfrac{1}{2} \| w \|^2$ 최소화

라그랑주 승수법

$g_i(x_1, \ldots, x_n) = C_i \ \ (i = 1, \ldots, k)$의 제약조건에서 $f(x_1, \ldots, x_n)$의 최소 조건

$$\nabla f = \sum_i \alpha_i \nabla g_i \ \Rightarrow \ \nabla f - \sum_i \alpha_i \nabla g_i = 0$$

$L = f - \sum_i \alpha_i g_i$ 일 때, $\nabla L = 0$ 과 동일한 식

$t_i(\boldsymbol{w} \bullet \boldsymbol{x}_i + b) - 1 = 0$ 의 제약조건에서 $\frac{1}{2} \| \boldsymbol{w} \|^2$의 최소조건

$$L = \frac{1}{2} \| \boldsymbol{w} \|^2 - \sum_i \alpha_i (t_i(\boldsymbol{w} \bullet \boldsymbol{x}_i + b) - 1)$$ 에서

$$\nabla_w L = \boldsymbol{w} - \sum_i \alpha_i t_i \boldsymbol{x_i} = 0 \quad \Rightarrow \quad \boldsymbol{w} = \sum_i \alpha_i t_i \boldsymbol{x_i}$$

$$\nabla_b L = -\sum_i \alpha_i t_i = 0 \quad \Rightarrow \quad \sum_i \alpha_i t_i = 0$$

두 결과를 L에 대입

$$L = \frac{1}{2}(\sum_i \alpha_i t_i \boldsymbol{x_i}) \bullet (\sum_i \alpha_i t_i \boldsymbol{x_i}) - (\sum_i \alpha_i t_i \boldsymbol{x_i}) \bullet (\sum_i \alpha_i t_i \boldsymbol{x_i}) - \sum_i \alpha_i t_i b + \sum_i \alpha_i$$

$$L = \sum_i \alpha_i - \frac{1}{2}(\sum_i \sum_j \alpha_i \alpha_j t_i t_j \boldsymbol{x_i} \bullet \boldsymbol{x_j})$$

$$L = \sum_i \alpha_i - \frac{1}{2}\left(\sum_i \sum_j \alpha_i \alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j\right)$$

α, t는 모두 스칼라, L은 내적 $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$와 연관이 있다.

t_i, $\boldsymbol{x}_i$는 주어진 값, L은 α_i에 대한 이차식

목적함수 L: QP (Quadratic Programming)

$L = \sum_i \alpha_i - \frac{1}{2}\left(\sum_i \sum_j \alpha_i \alpha_j t_i t_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j\right)$ 를 최대화하고,

모든 샘플에 대해 $\widehat{\alpha_i} \geq 0$ 이고, $\sum_i \widehat{\alpha_i} t_i = 0$ 를 만족하는 $\hat{\alpha}$ 찾기

$\hat{\alpha}$를 찾았다면

$$\hat{\boldsymbol{w}} = \sum_i \hat{\alpha}_i t_i \boldsymbol{x_i}$$

$$\hat{b} = \frac{1}{n_s} \sum_{\hat{\alpha}_i > 0} (t_i - \hat{\boldsymbol{w}} \bullet \boldsymbol{x_i})$$

"목표: 마진을 가장 크게 하는 w와 b 찾기"

비선형 SVM

도로가 직선으로 결정되지 못하는 경우 샘플들의 공간을 변환

$\boldsymbol{x} \rightarrow \phi(\boldsymbol{x}),\ \boldsymbol{x}_i \bullet \boldsymbol{x}_j \rightarrow \phi(\boldsymbol{x}_i) \bullet \phi(\boldsymbol{x}_j)$

커널 SVM

변환 ϕ의 계산 없이 바로 $\phi(\boldsymbol{x}_i) \bullet \phi(\boldsymbol{x}_j)$를 구하는 커널, 계산복잡도의 이득

$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \phi(\boldsymbol{x}_i) \bullet \phi(\boldsymbol{x}_j)$

이 경우 $L = \sum_i \alpha_i - \frac{1}{2}(\sum_i \sum_j \alpha_i \alpha_j t_i t_j \phi(\boldsymbol{x}_i) \bullet \phi(\boldsymbol{x}_j))$

$$= \sum_i \alpha_i - \frac{1}{2}(\sum_i \sum_j \alpha_i \alpha_j t_i t_j K(\boldsymbol{x}_i, \boldsymbol{x}_j))$$

파이썬에서

- sklearn.svm의 LinearSVC
- sklearn.svm의 SVC
- sklearn.linear_model의 SGDClassifier

라이브러리	알고리즘	커널 트릭	계산 복잡도
LinearSVC	목적 함수	불가능	$O(m \times n)$
SGDClassifier	확률적 경사하강법	불가능	$O(m \times n)$
SVC	목적 함수	가능	$O(m^2 \times n)$ ~ $O(m^3 \times n)$

참고 자료

- 오헬리앙 제롱, *핸즈온 머신러닝 2판* (한빛미디어, 2020), 205-228, 887-889

- Patrick Winston, “16. Learning: Support Vector Machines” *MIT 6.034 Artificial Intelligence, Fall 2010*. last modified JAN 11, 2014, https://youtu.be/_PwhiWxHK8o