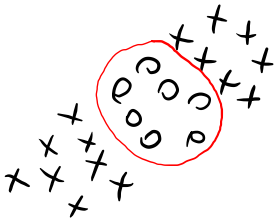
SUPPORT VECTOR MACHINES

27/4/21



Main Idea

1. Can we augment the feature space so that the data become linearly separable in the higher dimensional space?
2. Can we do this without having to work on the higher dimensional space?

MOTIVATION

p features : $X_1, \dots, X_p$

Augmented 2p features : $X_1, X_1^2, X_2, X_2^2, \dots, X_p, X_p^2$
$y_1 \quad y_2 \quad \dots \quad y_{2p}$

$X_1 X_2$
$X_i X_k$
$X_1 X_2 X_3$

SUPPORT VECTOR CLASSIFIER ON AUGMENTED SPACE

$$\min_{\beta_0, \beta, \underline{\varepsilon}} \frac{1}{2} \|\beta\|^2$$

s.t.

$$y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2 \right) \geq 1 - \varepsilon_i \quad \forall i$$

$$\varepsilon_i \geq 0$$

$$\sum_{i=1}^{n} \varepsilon_i \leq C$$

In the augmented space : LINEAR DECISION SURFACE $\alpha_0 + \underline{\alpha}^t \underline{y} = 0$

In the original space : $\beta_0 + \sum_{j=1}^{p} \beta_{j1} x_{ij} + \sum_{j=1}^{p} \beta_{j2} x_{ij}^2 = 0$ QUADRATIC DECISION SURFACE

SVC : Linear decision surface

$$f(\underline{x}) = \beta_0 + \underline{\beta}^t \underline{x} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j = \beta_0 + \underline{\beta} \cdot \underline{x}$$

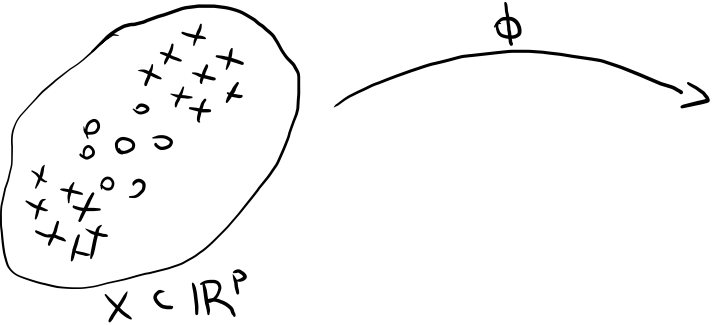
$$= \beta_0 + \sum_{i=1}^{n} \gamma_i y_i (\underline{x}_i \cdot \underline{x})$$

$$\underline{\beta} = \sum_{i=1}^{n} \gamma_i y_i \underline{x}_i$$

Also in the estimation of parameters, only inner products appear:

$$\max_{\underline{\gamma}} \sum_{i=1}^{n} \gamma_i - \frac{1}{2} \sum_{i,j=1}^{n} \gamma_i \gamma_j y_i y_j (\underline{x}_i \cdot \underline{x}_j)$$

$\binom{n}{2}$ inner products



$$f(\underline{x}) = \beta_0 + \underline{\beta}^t \phi(\underline{x})$$

$$\underline{\beta} = \sum_{i=1}^{\hat{n}} \gamma_i y_i \phi(\underline{x}_i)$$

$$\Downarrow$$

$$f(\underline{x}) = \beta_0 + \sum_{i=1}^{\hat{n}} \gamma_i y_i \boxed{\phi(\underline{x}_i) \cdot \phi(\underline{x})}$$

For estimation, we would need

$$\phi(\underline{x}_i) \cdot \phi(\underline{x}_j)$$

KERNEL TRICK

Find a function (Kernel) which is defined in the input space but which acts as a dot product in some feature space $\mathbb{R}^N$ for some map ϕ.

$K(\underline{x}, \underline{z}) = \phi(\underline{x}) \cdot \phi(\underline{z})$ in some $\mathbb{R}^N$ for some function ϕ

Mercer's Theorem states the properties that K needs to satisfy to be a dot product in $\mathbb{R}^N$

Using one such Kernel:

$$f(\underline{x}) = \beta_0 + \sum_{i=1}^{n} \gamma_i y_i K(\underline{x}_i, \underline{x})$$

SUPPORT VECTOR MACHINE (Vapnik, 1990)

And feed $K(\underline{x}_i, \underline{x}_j)$ to the optimisation problem.

MOST COMMON KERNELS

LINEAR KERNEL

$$K(\underline{x}, \underline{z}) = \sum_{j=1}^{p} x_j z_j$$

(standard inner product, SVC, linear decision surface)

POLYNOMIAL KERNEL

$$K(\underline{x}, \underline{z}) = \left(1 + \sum_{j=1}^{p} x_j z_j\right)^m$$

degree m

RADIAL (RBF) KERNEL

$$K(\underline{x}, \underline{z}) = \exp\left(-\gamma \sum_{j=1}^{p} (x_j - z_j)^2\right) \qquad \gamma > 0$$

$\left(\gamma = \frac{1}{2\sigma^2}\right.$ GAUSSIAN KERNEL$)$

$\underline{z}$ far from $\underline{x}$ $\rightarrow K(\underline{x}, \underline{z})$ small

Local

Tuning parameters typically selected by cross validation.

EXAMPLE : QUADRATIC KERNEL ($m=2$)

$$K(\underline{x}, \underline{z}) = \left(1 + \sum_{j=1}^{p} x_j z_j\right)^2 = \sum_{j,k=1}^{p} x_j x_k z_j z_k + 2\sum_{j=1}^{p} x_j z_j + 1$$

$$= \sum_{j,k=1}^{p} (x_j z_j)(x_k z_k) + \sum_{j=1}^{p} (\sqrt{2} x_j)(\sqrt{2} z_j) + 1$$

So

$$\phi(\underbrace{\underline{x}}_{(x_1,\dots,x_p)}) = [\underbrace{x_1^2, x_1x_2, \dots, x_p^2}_{p^2}, \underbrace{\sqrt{2}x_1, \sqrt{2}x_2, \dots, \sqrt{2}x_p}_{p}, 1]$$

$$K(\underline{x}, \underline{z}) = \phi(\underline{x}) \cdot \phi(\underline{z}) \quad \text{with} \quad \phi : \mathbb{R}^p \to \mathbb{R}^{p^2+p+1}$$

This kernel leads to a quadratic decision surface in the input space.

MORE THAN 2 CLASSES

Two possible approaches:

1. ONE VERSUS THE REST
2. ONE VERSUS ONE

---

(1) K classes → Train K binary classifiers where the labels are "class c" versus "not class c" for $c = 1, \dots, K$

$\underline{x}$ test observation → $f_1(\underline{x}), \dots, f_K(\underline{x})$ → classify $\underline{x}$ to the class with largest $|f_c(\underline{x})|$

(2) Fit $\binom{K}{2} = \frac{K(K-1)}{2}$ classifiers, one for each pair of classes

⇒ classify $\underline{x}$ to class with the majority vote among the $\frac{K(K-1)}{2}$ predictions

SVM vs LOGISTIC REGRESSION

SVC

$$\min_{\beta_0, \beta, \xi} \ \frac{1}{2}\|\beta\|^2$$

s.t. ① $y_i(\beta_0 + \beta^t x_i) \geq 1 - \xi_i$ $\rightarrow$ $\xi_i \geq 1 - y_i(\beta_0 + \beta^t x_i)$

② $\xi_i \geq 0$ $\rightarrow$ $\xi_i \geq 0$

③ $\sum_{i=1}^{n} \xi_i \leq C$

①② $\Rightarrow$ $\xi_i \geq \max\{0, 1 - y_i(\beta_0 + \beta^t x_i)\} = [1 - y_i(\beta_0 + \beta^t x_i)]_+$ $\quad [a]_+ = \max\{0, a\}$

Using ③:

$$\sum_{i=1}^{n} [1 - y_i(\beta_0 + \beta^t x_i)]_+ \leq \sum_{i=1}^{n} \xi_i \leq C \Rightarrow \sum_{i=1}^{n} [1 - y_i(\beta_0 + \beta^t x_i)]_+ \leq C$$

So the original problem can be reformulated as:

$$\min_{\beta_0, \beta} \ \frac{1}{2}\|\beta\|^2$$

$$\text{s.t.} \quad \sum_{i=1}^{n} [1 - y_i(\beta_0 + \beta^t x_i)]_+ \leq C$$

Using Lagrange multipliers this becomes

$$\min_{\beta_0,\beta} \frac{1}{2}\|\beta\|^2 + \lambda\left[\sum_{i=1}^{n}\left[1-y_i(\beta_0+\beta^t x_i)-C\right]_+\right.$$

$$= \min_{\beta_0,\beta} \frac{1}{2}\|\beta\|^2 + \lambda \sum_{i=1}^{n}\left[1-y_i(\beta_0+\beta^t x_i)\right]_+$$

$$= \min_{\beta_0,\beta} \underbrace{\sum_{i=1}^{n}\left[1-y_i(\beta_0+\beta^t x_i)\right]_+}_{\text{Loss}} + \frac{\gamma}{2}\|\beta\|^2$$

Ridge regression

$$\sum_{i=1}^{n}(y_i-\beta^t x_i)^2 + \lambda\|\beta\|^2$$

HINGE LOSS

$$L(y,f(x)) = \begin{cases} 0 & 1-yf(x)<0 \iff yf(x)>1 \\ 1-yf(x) & 1-yf(x)>0 \iff yf(x)<1 \end{cases}$$

No loss for points on correct side of margin

If on wrong side of margin, loss is proportional to distance from margin

Let us now consider logistic regression

OBJECTIVE FUNCTION $-\sum_{i=1}^{n} \log(p(c_i|x_i))$

LOSS $L(y_i, \underbrace{f(x_i)}_{\beta_0 + \beta^t x_i}) = \begin{cases} -\log p(1|x_i) & y_i = 1 \\ -\log(1-p(1|x_i)) & y_i = -1 \end{cases}$

$y_i = 1$ $\quad -\log p(1|x_i) = \log\left(\frac{1}{p(1|x_i)}\right) =$

$= \log\left(\frac{1+e^{\beta^t x_i}}{e^{\beta^t x_i}}\right) = \log\left(e^{-\beta^t x_i} + 1\right)$

$p(1|x_i) = \frac{e^{\beta^t x_i}}{1+e^{\beta^t x_i}}$

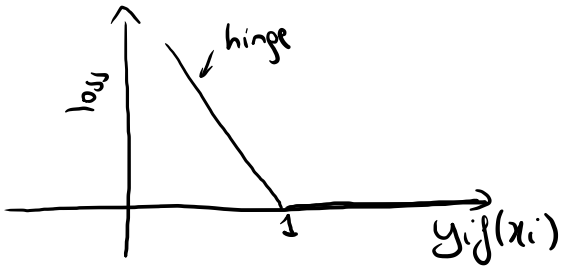
$y_i = -1$ $\quad -\log(1-p(1|x_i)) = \log\left(\frac{1}{1-\frac{e^{\beta^t x_i}}{1+e^{\beta^t x_i}}}\right) = \log\left(1+e^{\beta^t x_i}\right)$

$L(y_i, f(x_i)) = \log\left(1+e^{-y_i f(x_i)}\right)$

So logistic regression in an SVM framework would be:

$$\min_{\beta_0, \beta} \sum_{i=1}^{n} \log\left(1 + e^{-y_i f(x_i)}\right) + \frac{\gamma}{2}\|\beta\|^2$$

LOGISTIC REGRESSION WITH RIDGE PENALTY



In general, logistic regression and SVC perform similarly when the decision surface is linear.

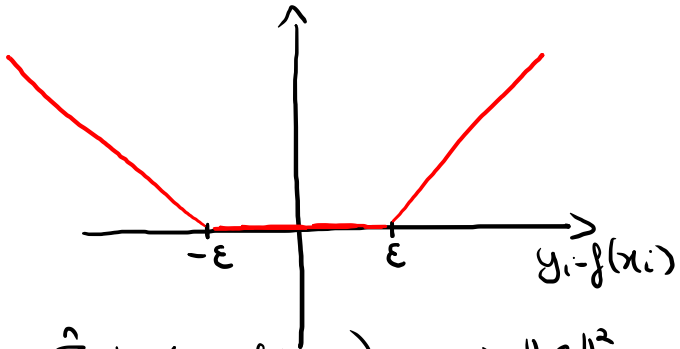
Support vector machines perform better on non-linear problems (unless you make logistic regression non-linear by adding features)

SVM in regression

The hinge loss is replaced by a **ε-INSENSITIVE LOSS**

ε-INSENSITIVE LOSS

$$L_\varepsilon(y_i, f(x_i)) = \begin{cases} 0 & \text{if } |y_i - f(x_i)| \le \varepsilon \\ |y_i - f(x_i)| - \varepsilon & \text{if } |y_i - f(x_i)| > \varepsilon \end{cases}$$



SVM REGRESSION

$$\min_{\beta_0, \beta} \sum_{i=1}^{n} L_\varepsilon(y_i, f(x_i)) + \frac{\gamma}{2}\|\beta\|^2$$