Learning From Data
Lecture 6: Support Vector Machines (Part Two)

Yang Li yangli@sz.tsinghua.edu.cn

October 22, 2021

Introduction

Today's Lecture

Supervised Learning (Part V)

- Soft margin SVM
- Kernel SVM
- Some other kernel methods

Written Assignment 1 is due today.

Midterm is on November 5.

Q & A

Question

What's the difference between OLS and least absolute deviation (LAD) and their geometric interpretation? *(WA1)*

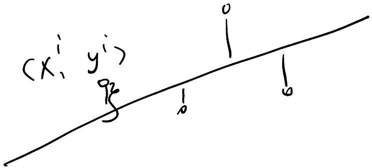
Q & A

Question

What's the difference between OLS and least absolute deviation (LAD) and their geometric interpretation? *(WA1)*

$- \frac{1}{\tau} \sum_{i=1}^{m} |y^i - \theta^T x^i - \mu| = 0$

	OLS	LAD
Loss	$J(\theta) = \frac{1}{2} \sum_{i=1}^{m} \|\|y - \theta^T x\|\|^2$	$J(\theta) = \sum_{i=1}^{m} \|y^i - \theta^T x^{(i)}\|$



Q & A

Question

What's the difference between OLS and least absolute deviation (LAD) and their geometric interpretation? *(WA1)*

	OLS	LAD
Loss	$J(\theta) = \frac{1}{2}\sum_{i=1}^{m} \|\|y - \theta^T x\|\|^2$	$J(\theta) = \sum_{i=1}^{m} \|y - \theta^T x^{(i)}\|$
Maximum likelihood $p(y\|x)$	$\mathcal{N}(0, \sigma^2)$	$Laplace(0, \tau)$

Q & A

Question

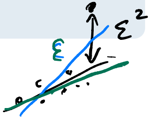
What's the difference between OLS and least absolute deviation (LAD) and their geometric interpretation? *(WA1)*

	OLS	LAD
Loss	$J(\theta) = \frac{1}{2}\sum_{i=1}^{m} \|\|y - \theta^T x\|\|^2$	$J(\theta) = \sum_{i=1}^{m} \|y - \theta^T x^{(i)}\|$
Maximum likelihood $p(y\|x)$	$\mathcal{N}(0, \sigma^2)$	$Laplace(0, \tau)$
Geometric meaning	sum of residual (error) squares	sum of absolute errors

Q & A

Question

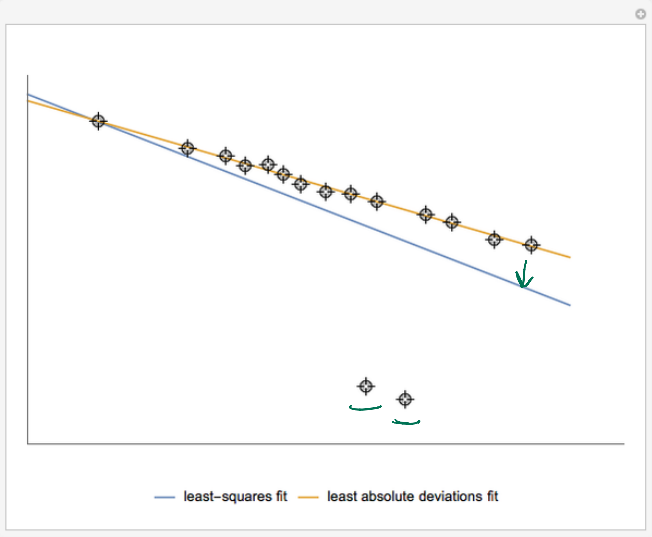
What's the difference between OLS and least absolute deviation (LAD) and their geometric interpretation? *(WA1)*



	OLS	LAD
Loss	$J(\theta) = \frac{1}{2}\sum_{i=1}^{m} \|\|y - \theta^T x\|\|^2$	$J(\theta) = \sum_{i=1}^{m} \underbrace{\|y - \theta^T x^{(i)}\|}_{\text{L1-norm}}$
Maximum likelihood $p(y\|x)$	$\mathcal{N}(0, \sigma^2)$	$Laplace(0, \tau)$
Geometric meaning	sum of residual (error) squares	sum of absolute errors
Pros/Cons	has unique global solution*, sensitive to outliers	robust to outliers, instability due to multiple solutions

Q & A

Comparison between OLS and LAD:



Review: Linear SVM Dual

$(w, b) \rightarrow \alpha$

Dual optimization problem: *(Check derivation)*

$y^i(w^T x^i + b) \geq 1.$

dual :

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$s.t. \ \alpha_i \geq 0, i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

$$\begin{cases} \min_{w,b} \frac{1}{2}\|w\|^2 \\ s.t. \ y^i(w^T x^i + b) \geq 1 \end{cases}$$

for $i = 1, \ldots, m$

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i \left(y^i (w^T x^i + b) - 1 \right)$$

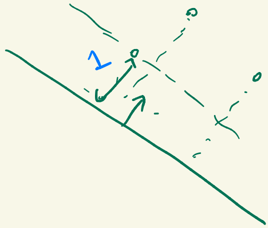
primal $\Rightarrow$ $\min_{w,b} L(w,b,\alpha)$

$$\frac{\partial L}{\partial w} = 0. \quad w^* = \sum_{i=1}^{m} \alpha_i y^i x^i$$

b^*

$g_i(w) \le 0$.

$-y^i(w^T x^i + b) + 1 \le 0$.



when $y^i = 1$, $\quad -(w^T x^i + b) \le -1$

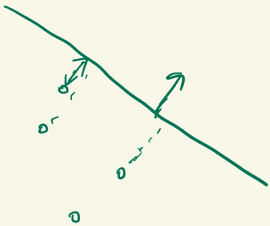
$$w^T x^i + b \ge 1$$

if $\alpha_i > 0$, there is some x^i

$$w^T x^i + b = 1.$$

(w is w^*)

$$\min_{\substack{b \\ y^i = 1}} w^T x^i + b = 1. \quad (1)$$



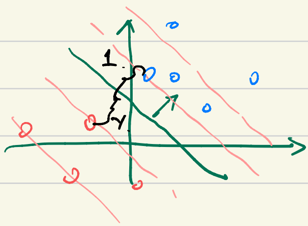
when $y^i = -1$. $\quad w^T x^i + b \le -1$.

$$\max_{\substack{b \\ y^i = -1}} w^T x^i + b = -1. \quad (2)$$

$$(2)\ \max_{\substack{b \\ y_i = -1}} w^T x^i + b \ + \ \overset{(1)}{\min_{y^i = 1} w^T x^i + b}$$

$$(-1) \quad + \quad (1) \ = \ 0.$$

$$b = \frac{1}{2}\left(\max_{y^i = -1}(w^T x^i + b) + \min_{y^i = 1}(w^T x^i + b)\right).$$



w^*

Review: Linear SVM Dual

Dual optimization problem:*(Check derivation)*

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$s.t.\ \alpha_i \geq 0, i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

Solution to the primal problem:

$$w^* = \sum_i \alpha_i^* y^{(i)} x^{(i)}$$

$$b^* = -\frac{1}{2} \left(\max_{i:y^{(i)}=-1} {w^*}^T x^{(i)} + \min_{i:y^{(i)}=1} {w^*}^T x^{(i)} \right)$$

Review: Linear SVM Dual

Dual optimization problem: *(Check derivation)*

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

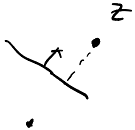
$$s.t.\ \alpha_i \geq 0, i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

Solution to the primal problem:

$$w^* = \sum_{i} \alpha_i^* y^{(i)} x^{(i)}$$

$$b^* = -\frac{1}{2} \left(\max_{i: y^{(i)}=-1} {w^*}^T x^{(i)} + \min_{i: y^{(i)}=1} {w^*}^T x^{(i)} \right)$$



For a new sample z, the SVM prediction is sign $\left[{w^*}^T z + b \right]$

$w^T z + b = \sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, z \rangle + b$

Review: Linear SVM Summary

- Input:: m training samples $(x^{(i)}, y^{(i)}), y^i \in \{-1, 1\}$
- Output: optimal parameters w^*, b^*
- Step 1: solve the dual optimization problem

$$\alpha^* = \max_{\alpha} W(\alpha)$$

$$s.t.\ \alpha_i \geq 0, \sum_{i=1}^{m} \alpha_i y^{(i)} = 0, i = 1, \ldots, m$$

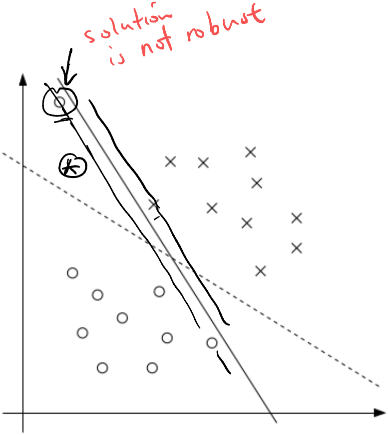
- Step 2: compute the optimal parameters w^*, b^*

$$w^* = \sum_i \alpha_i^* y^{(i)} x^{(i)}$$

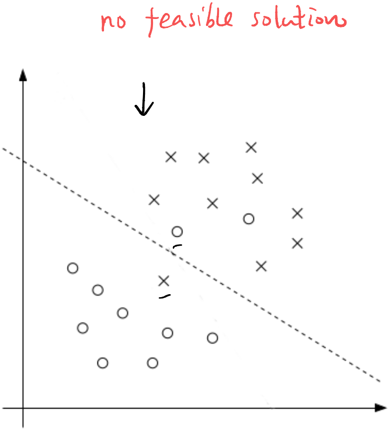
$$b^* = -\frac{1}{2}\left(\max_{i:y^{(i)}=-1} {w^*}^T x^{(i)} + \min_{i:y^{(i)}=1} {w^*}^T x^{(i)} \right)$$

Soft Margin SVM

Limitations of the basic SVM



Outliers



Non-linearly separable cases

Soft Margin SVM

$|\xi_i|$

Functional margin $1 - \xi_i \leq 1$:

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + \underbrace{C \sum_{i=1}^{m} \xi_i}$$

slack variable

2m
$$\left\{ \begin{array}{l} s.t.\ y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i \\ \xi_i \geq 0, i = 1, \ldots, m \end{array} \right.$$

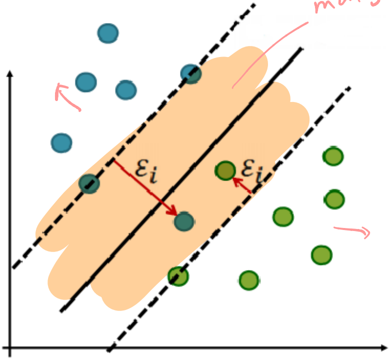
$\rightarrow g(w,b,\xi) \leq 0$

$\rightarrow \bar{g}(w,b,\xi) \leq 0$

inside margin

$y^i(w^T x^i + b) \geq 1$

- C: relative weight on the regularizer
- L_1 regularization let most $\xi_i = 0$, such that their functional margins $1 - \xi_i = 1$



Soft Margin SVM

The generalized Lagrangian function:

$$L(w,b,\xi,\alpha,r) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i - \sum_i^m \alpha_i\left[y^{(i)}(w^Tx^{(i)}+b) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i\xi_i$$

$\xi_i \geq 0 \longleftrightarrow -\xi_i \leq 0.$

$\frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{m} \alpha_i y^i x^i$

$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{m} \alpha_i y^i = 0.$

$\frac{\partial L}{\partial \xi_i} = 0. \Rightarrow C\cdot 1 - \alpha_i \cdot 1 - r_i \cdot 1 \Rightarrow C - \alpha_i - r_i = 0$ for all i

$L(w,b,\xi,\alpha,r) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m}\alpha_i(y^i(w^Tx^i+b)-1) - \sum_{i=1}^{m}\xi_i\underbrace{(C-\alpha_i-r_i)}_{=0.}$

$W(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y^iy^j\alpha_i\alpha_j x^{iT}x^j$

$\left.\begin{array}{l}\alpha_i \geq 0.\\ r_i \geq 0.\end{array}\right\}$ since $r_i = C - \alpha_i$, $\alpha_i \geq 0$, $r_i \geq 0 \Rightarrow \boxed{\alpha_i \leq C}$

$\left.\begin{array}{l}0 \leq \alpha_i \leq C\\ r_i \geq 0.\end{array}\right\}$ dual constraints.

Soft Margin SVM

The generalized Lagrangian function:

$$L(w, b, \xi, \alpha, r) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i}^{m}\alpha_i\left[y^{(i)}(w^T x^{(i)} + b) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i\xi_i$$

Dual problem:

Soft Margin SVM

The generalized Lagrangian function:

$$L(w,b,\xi,\alpha,r) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i}^{m}\alpha_i\left[y^{(i)}(w^T x^{(i)}+b) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i\xi_i$$

Dual problem:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle$$

$$s.t.\ 0 \le \alpha_i \le C, i = 1, \ldots, m$$

$$\underbrace{\sum_{i=1}^{m}\alpha_i y^{(i)} = 0}$$

w^* is the same as the non-regularizing case, but b^* has changed.

Soft Margin SVM

Dual problem:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$s.t.\ \underline{0 \leq \alpha_i \leq C}, i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

By the KKT dual-complentary conditions, for all i, $\alpha_i^* g_i(w^*) = 0$

(x^i, y^i)

$\underline{\alpha_i = 0} \iff$

$\underline{\alpha_i = C} \iff$

$\underline{0 < \alpha_i < C} \iff$

$$w = \begin{bmatrix} w \\ b \\ \xi \end{bmatrix}$$

KKT conditions

Stationary: $\frac{\partial L}{\partial w} = 0 \Rightarrow w^* = \sum_{i=1}^{m} \alpha_i^* y^i x^i$

$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{m} \alpha_i y^i = 0$

$\frac{\partial L}{\partial \xi} = 0 \Rightarrow \underline{\alpha_i = c - r_i}$ (1)

Complementary: $\alpha_i g_i(\vec{w}) = 0. \Rightarrow \alpha_i \left(y^i (w^T x^i + b) - 1 + \xi_i \right) = 0.$ (2)

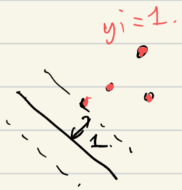
$r_i \bar{g}_i(\vec{w}) = 0 \Rightarrow r_i \xi_i = 0.$ (3)

dual feasibility: $\alpha_i \geq 0$ (4)

$r_i \geq 0$ (5)

primal feasibility: $g_i(\vec{w}) \leq 0. \Rightarrow y^i (w^T x^i + b) - 1 + \xi_i \geq 0$ (6)

$\bar{g}_i(\vec{w}) \leq 0 \Rightarrow \xi_i \geq 0.$ (7)



Case 1. $\alpha_i = 0.$

By (1), $\alpha_i = c - r_i$, then $r_i = c$

Since $c > 0 \Rightarrow r_i > 0.$

By (3) $r_i \xi_i = 0. \Rightarrow \xi_i = 0$

$y^i (w^T x^i + b) - 1 \geq 0.$

$y^i (w^T x^i + b) \geq 1$

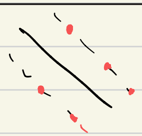
$x^{(i)}$ is on the correct side of the margin!

Case 2 $\alpha_i \neq 0$, & $\alpha_i < c$

$r_i = c - \alpha_i$, $\alpha_i < c$, $r_i > 0.$

By (3), $r_i \xi_i = 0$, $\Rightarrow \xi_i = 0$, since $\alpha_i \neq 0$, $\alpha_i \geq 0.$

$\alpha_i > 0$. by (2), $y^i (w^T x^i + b) - 1 + \cancel{\xi}^{\,0} = 0.$

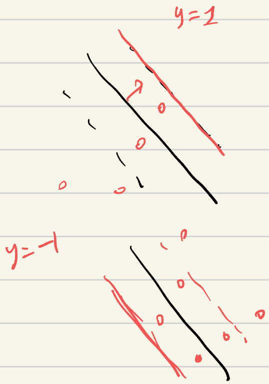


$y^i (w^T x^i + b) = 1.$

$x^{(i)}$ on the margin!

Case 3 $\alpha_i \neq 0$, $\alpha_i = c$

Case 3. $\alpha_i \neq 0, \alpha_i = C$



Since $\alpha_i = C$, $r_i = \underline{C - \alpha_i} = 0$.

Since $r_i \xi_i \geq 0$. $\Rightarrow$. $\xi_i \geq 0$.

Since $\alpha_i = \underline{C > 0}$,

$$y^i (w^T x^i + b) - 1 + \underline{\xi} = 0.$$

$$y^i (w^T x^i + b) \leq 1.$$

$x^{(i)}$ are on the wrong side of the margin.

Soft Margin SVM

Dual problem:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle$$

$$s.t.\ 0 \le \alpha_i \le C, i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

By the KKT dual-complentary conditions, for all i, $\alpha_i^* g_i(w^*) = 0$

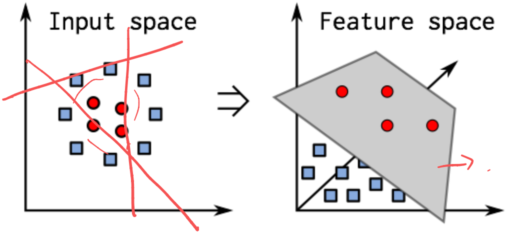
$$\begin{array}{lcll} \alpha_i = 0 & \Longleftrightarrow & y^{(i)}(w^T x^{(i)} + b) \ge 1 & \text{correct side of margin} \\ \alpha_i = C & \Longleftrightarrow & y^{(i)}(w^T x^{(i)} + b) \le 1 & \text{wrong side of margin} \\ 0 < \alpha_i < C & \Longleftrightarrow & y^{(i)}(w^T x^{(i)} + b) = 1 & \text{at margin} \end{array}$$

support vectors.

Kernel SVM

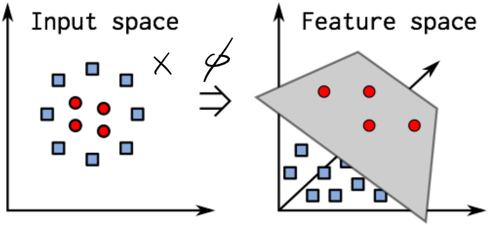
Non-linear SVM

For non-separable data, we can use the **kernel trick**: Map input values $x \in \mathbb{R}^d$ to a higher dimension $\phi(x) \in \mathbb{R}^D$, such that the data becomes separable.



Non-linear SVM

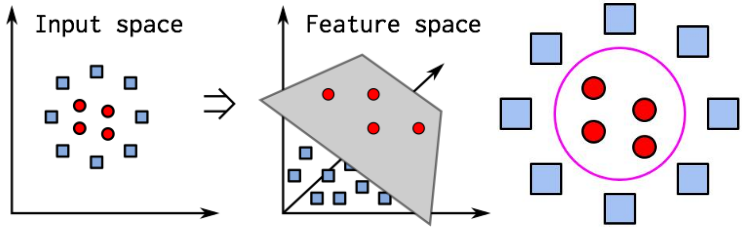
For non-separable data, we can use the **kernel trick**: Map input values $x \in \mathbb{R}^d$ to a higher dimension $\phi(x) \in \mathbb{R}^D$, such that the data becomes separable.



- ϕ is called a **feature mapping**.

Non-linear SVM

For non-separable data, we can use the **kernel trick**: Map input values $x \in \mathbb{R}^d$ to a higher dimension $\phi(x) \in \mathbb{R}^D$, such that the data becomes separable.



- ϕ is called a **feature mapping**.
- The classification function $w^T x + b$ becomes nonlinear: $w^T \phi(x) + b$

Kernel Function

Given a feature mapping ϕ, we define the **kernel function** to be

$$K(x, z) = \phi(x)^T \phi(z)$$

Kernel Function

Given a feature mapping ϕ, we define the **kernel function** to be

$$K(x, z) = \phi(x)^T \phi(z)$$

Some kernel functions are easier to compute than $\phi(x)$, e.g.

$$K(x, z) = (x^T z)^2 = \phi(x)^T \phi(z)$$ what is $\phi(x)$?

$x = (x_1, x_2)$
$z = (z_1, z_2)$.

Kernel Function

Given a feature mapping ϕ, we define the **kernel function** to be

$$K(x, z) = \phi(x)^T \phi(z)$$

Some kernel functions are easier to compute than $\phi(x)$, e.g.

$x^T z$

multiply

$$K(x,z) = (x^T z)^2 = \underbrace{\left(\sum_{i=1}^{n} x_i z_i\right)}_{x^T z} \underbrace{\left(\sum_{j=1}^{n} x_j z_j\right)}_{x^T z} = \underbrace{\sum_{i=1}^{n}\sum_{j=1}^{n} x_i, x_j, z_i, z_j}$$

$$= \phi(x)^T \phi(z)$$

Example:

2D case.

$$(x^T z)^2 = (x_1 z_1 + x_2 z_2)(x_1 z_1 + x_2 z_2)$$

$$= (x_1 z_1)^2 + (x_1 z_1)(x_2 z_2) + (x_2 z_2)(x_1 z_1) + (x_2 z_2)^2$$

$$= (x_1^2 z_1^2) + (x_1 x_2)(z_1 z_2) + (x_2 x_1)(z_2 z_1) + (x^2 z^2)$$

$$= \left\langle \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_2 x_1 \\ x_2^2 \end{bmatrix}, \begin{bmatrix} z_1^2 \\ z_1 z_2 \\ z_2 z_1 \\ z_2^2 \end{bmatrix} \right\rangle$$

Kernel Function

Given a feature mapping ϕ, we define the **kernel function** to be

$$K(x,z) = \phi(x)^T\phi(z)$$

Some kernel functions are easier to compute than $\phi(x)$, e.g.

$$K(x,z) = (x^Tz)^2 = \sum_{i=1}^{n} x_i, z_i \sum_{j=1}^{n} x_j, z_j = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i, x_j, z_i, z_j$$
$$= \phi(x)^T\phi(z)$$

where $\phi(x) = \left.\begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1 \cdot x_n \\ x_2 \cdot x_1 \\ x_nx_{n-1} \\ x_nx_n \end{bmatrix}\right\} n^2$ takes $O(n^2)$ operations to compute, while

$K(x,z)$

$= (x^Tz)^2$ only takes $O(n)$

Kernel SVM

In the dual problem, replace $\langle x_i, y_j \rangle$ with $\langle \phi(x_i), \phi(y_i) \rangle = K(x_i, x_j)$

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \underline{K(x_i, x_j)}$$

$\underline{\phi(x^i)^T \phi(x^j)}$

$$s.t.\ 0 \le \alpha_i \le C, i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

Kernel SVM

In the dual problem, replace $\langle x_i, y_j \rangle$ with $\langle \phi(x_i), \phi(y_i) \rangle = K(x_i, x_j)$

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j K(x_i, x_j)$$

$$s.t. \; 0 \le \alpha_i \le C, i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$$

No need to compute $w^* = \sum_{i=1}^{m} \alpha_i^* y^{(i)} \phi(x^{(i)})$ explicitly since

$$f(x) = w^T \phi(x) + b = \left(\sum_{i=1}^{m} \alpha_i y^{(i)} \phi(x^{(i)}) \right)^T \phi(x) + b$$

$$= \sum_{i=1}^{m} \alpha_i y^{(i)} \langle \phi(x^{(i)}), \phi(x) \rangle + b$$

$$= \sum_{i=1}^{m} \alpha_i y^{(i)} K(x^{(i)}, x) + b$$

$$= \sum_{i=1}^{m} \alpha_i y^i \phi(x^i)^T \phi(x) + b.$$

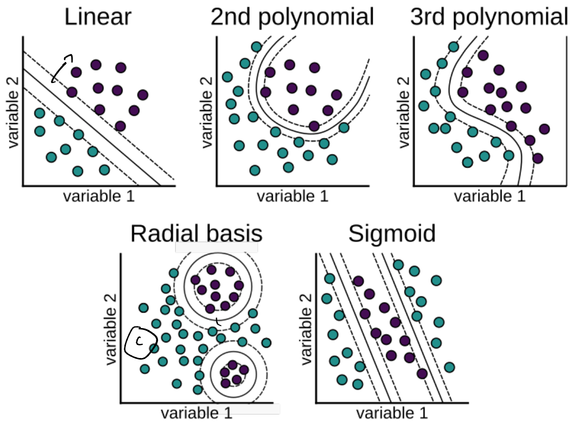
Kernel Matrix

kernel functions measure the similarity between samples x, z, e.g.

- Linear kernel: $K(x, z) = (x^T z)$
- Polynomial kernel: $K(x, z) = (x^T z + 1)^p$ $\quad p = 2$
- Gaussian / radial basis function (RBF) kernel:

$$K(x, z) = \exp\left(-\frac{||x - z||^2}{2\sigma^2}\right)$$

$$\exp(-\gamma\|x - z\|^2)$$

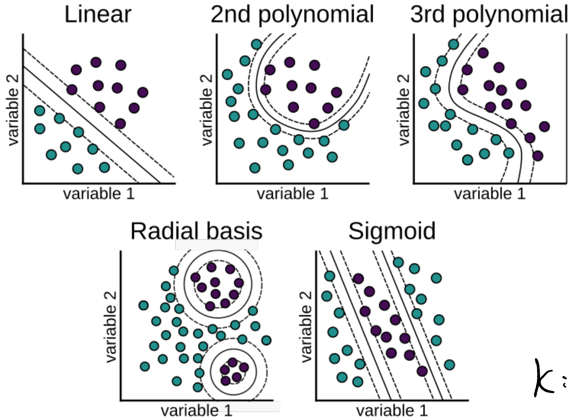


$(x^T z + 1)^2$

Kernel Matrix

kernel functions measure the similarity between samples x, z, e.g.

- Linear kernel: $K(x,z) = (x^T z)$
- Polynomial kernel: $K(x,z) = (x^T z + 1)^p$
- Gaussian / radial basis function (RBF) kernel:
  $K(x,z) = \exp\left(-\frac{||x-z||^2}{2\sigma^2}\right)$



Can any function $K(x,y)$ be a kernel function?

$K: X \times X \rightarrow \mathbb{R}$

Kernel Matrix

n : # of samples.

Represent kernel function as a matrix $K \in \mathbb{R}^{m \times m}$ where $K_{i,j} = K(x_i, x_j) = \underline{\phi(x_i)^T \phi(x_j)}$.

$$
\begin{bmatrix}
\phi(x^{(1)})^T \phi(x^{(1)}), & \cdots, & \phi(x^{(1)})^T \phi(x^{(m)}) \\
\phi(x^{(2)})^T \phi(x^{(1)}), & \cdots, & \phi(x^{(2)})^T \phi(x^{(m)}) \\
\vdots & \ddots & \\
\phi(x^{(m)})^T \phi(x^{(1)}) & \cdots & \phi(x^{(m)})^T \phi(x^{(m)})
\end{bmatrix}
$$

Kernel Matrix

Represent kernel function as a matrix $K \in \mathbb{R}^{m \times m}$ where $K_{i,j} = K(x_i, x_j) = \phi(x_i)\phi(x_j)$.

Theorem (Mercer)

Let $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ Then K is a valid (Mercer) kernel if and only if for any finite training set $\{x^{(i)}, \ldots, x^{(m)}\}$, K is symmetric positive semi-definite.

i.e. $\underbrace{K_{i,j} = K_{j,i}}$ and $x^T K x \geq 0$ for all $x \in \mathbb{R}^n$

$K = K^T$

Kernel SVM Summary

$\phi: X \to \mathbb{R}^D$.

- Input: m training samples $(x^{(i)}, y^{(i)}), y^i \in \{-1, 1\}$, kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, constant $C > 0$
- Output: non-linear decision function $f(x)$

Step 0: compute kernel matrix K for all x:

- Step 1: solve the dual optimization problem for α^*

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j K(x^{(i)}, x^{(j)})$$

$$s.t.\ 0 \leq \alpha_i \leq C, \sum_{i=1}^{m} \alpha_i y^{(i)} = 0, i = 1, \ldots, m$$

- Step 2: compute the optimal decision function

x_j

$$b^* = y^{(j)} - \sum_{i=1}^{m} \alpha_i^* y^{(i)} K(x^{(i)}, x^{(j)}) \text{ for some } 0 \leq \alpha_j \leq C$$

$$f(x) = \sum_{i=1}^{m} \alpha_i y^{(i)} K(x^{(i)}, x) + b^*$$

In practice, it's more efficient to compute kernel matrix K in advance.

SVM in Practice

Sequential Minimal Optimization: a fast algorithm for training soft margin kernel SVM

- Break a large SVM problem into smaller chunks, update two α_i's at a time
- Implemented by most SVM libraries.

SVM in Practice

Sequential Minimal Optimization: a fast algorithm for training soft margin kernel SVM

- Break a large SVM problem into smaller chunks, update two α_i's at a time
- Implemented by most SVM libraries.

original SVM for classification

$$y^i(w^T x_i + b) \geq 1 - \varepsilon_i$$

$$\downarrow$$

$$\max(0, \underbrace{|y^i - (w^T x^i + b)| - \varepsilon_i}_{\text{for regression}})$$

Other related algorithms

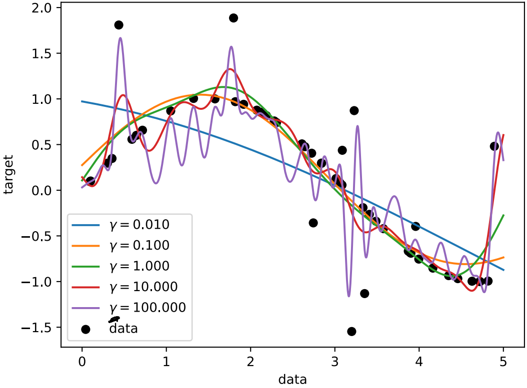
- Support Vector Regression (SVR)
- Least Square SVM (LS-SVM) homework.
- Multi-class SVM (Koby Crammer and Yoram Singer. 2002. *On the algorithmic implementation of multiclass kernel-based vector machines*. J. Mach. Learn. Res. 2 (March 2002), 265-292.)

Kernel Regularized Least Square

Other Kernel Methods

Kernel trick can be applied in many linear models, e.g.

- Kernel regularized least square regression
  - Numerical solution (gradient descent)
  - Analytically solution ← *See WA2*

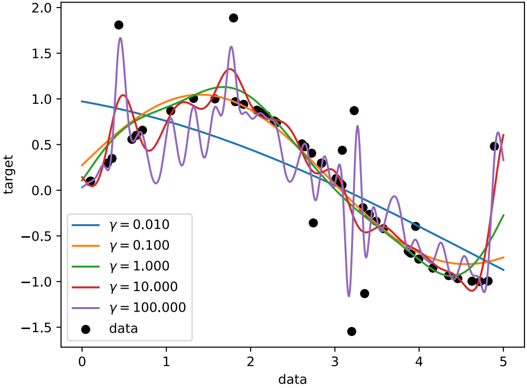


Regularized least square with RBF kernel $k(x, y) = \exp(-\gamma ||x - y||^2)$

Other Kernel Methods

Kernel trick can be applied in many linear models, e.g.

- Kernel regularized least square regression
  - Numerical solution (gradient descent)
  - Analytically solution ← *See WA2*



Regularized least square with RBF kernel $k(x, y) = \exp(-\gamma||x - y||^2)$

- Kernel PCA, Kernel CCA (in later lectures)

Review: Regularized Least Square Regression

Given $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$, $y^{(i)} \in \mathbb{R}, x^{(i)} \in \mathbb{R}^n$
Regularized least square:

$$\min_{\theta \in \mathbb{R}^n} \frac{1}{2} \underline{\sum_{i=1}^{m} ||y^{(i)} - \theta^T x^{(i)}||}_2^2 + \underbrace{\lambda \frac{1}{2} ||\theta||_2^2}$$

$|| \cdot ||$ is the L2 norm.

Review: Regularized Least Square Regression

Given $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$, $y^{(i)} \in \mathbb{R}, x^{(i)} \in \mathbb{R}^n$
Regularized least square:

$$\min_{\theta \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{m} ||y^{(i)} - \theta^T x^{(i)}||^2 + \lambda \frac{1}{2} ||\theta||^2$$

$|| \cdot ||$ is the L2 norm.
Gradient descent update:

$$\theta_j := (1 - \alpha\lambda)\theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)}) x_j^{(i)} \text{ for all } j = 1, \ldots, n$$

Review: Regularized Least Square Regression

Given $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$, $y^{(i)} \in \mathbb{R}, x^{(i)} \in \mathbb{R}^n$
Regularized least square:

$$\min_{\theta \in \mathbb{R}^n} \frac{1}{2} \sum_{i=1}^{m} ||y^{(i)} - \theta^T x^{(i)}||^2 + \lambda \frac{1}{2} ||\theta||^2$$

$|| \cdot ||$ is the L2 norm.
Gradient descent update:

$$\theta_j := (1 - \alpha\lambda)\theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)}) x_j^{(i)} \text{ for all } j = 1, \ldots, n$$

Vector notation:

$$\theta := (1 - \alpha\lambda)\theta + \alpha \underbrace{\sum_{i=1}^{m} (y^{(i)} - \theta^T x^{(i)})} x^{(i)}$$

Kernel Regularized Least Square Regression

Kernel Regularized Least Square (KRLS)

Given $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ with $y^{(i)} \in \mathbb{R}, x^{(i)} \in \mathbb{R}^n$ and a feature map $\phi : \mathbb{R}^n \to \mathbb{R}^d$:

$$\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{m} \frac{1}{2} ||y^{(i)} - \theta^T \phi(x)^{(i)}||^2 + \lambda \frac{1}{2} ||\theta||^2$$

Kernel Regularized Least Square Regression

Kernel Regularized Least Square (KRLS)

Given $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ with $y^{(i)} \in \mathbb{R}, x^{(i)} \in \mathbb{R}^n$ and a feature map $\phi : \mathbb{R}^n \to \mathbb{R}^d$:

$$\min_{\theta \in \mathbb{R}^n} \sum_{i=1}^{m} \frac{1}{2} ||y^{(i)} - \theta^T \phi(x)^{(i)}||^2 + \lambda \frac{1}{2} ||\theta||^2$$

Gradient descent update:

$$\theta := (1 - \alpha\lambda)\theta + \alpha \sum_{i=1}^{m} (y^{(i)} - \theta^T \phi(x)^{(i)}) \phi(x)^{(i)}$$

Kernel Regularized Least Square Regression

Kernel Regularized Least Square (KRLS)

Given $(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$ with $y^{(i)} \in \mathbb{R}, x^{(i)} \in \mathbb{R}^n$ and a feature map $\phi : \mathbb{R}^n \to \mathbb{R}^d$:

$$\min_{\theta \in \mathbb{R}^n} \sum_{i=1}^{m} \frac{1}{2} ||y^{(i)} - \theta^T \phi(x)^{(i)}||^2 + \lambda \frac{1}{2} ||\theta||^2$$

Gradient descent update:

$$\theta := (1 - \alpha\lambda)\theta + \alpha \sum_{i=1}^{m} (y^{(i)} - \underline{\theta}^T \phi(x)^{(i)}) \underline{\phi(x)^{(i)}}$$

How to use the kernel trick to solve it more efficiently?

Kernel Regularized Least Square Regression

Proposition 1.

Parameter θ can be written as a linear combination of $\phi(x^{(1)}), \ldots, \phi(x^{(m)})$: image of ϕ. on $x^1, \ldots, x^m$.

$$\frac{\partial J(\theta)}{\partial \theta} = 0.$$

$$\theta_i = \sum_{i=1}^{m} \beta_i \phi(x^{(i)}) \text{ for all } i = 1, \ldots, m$$

proof see page

Induction on k, the iteration index in GD.

$\underline{\theta^k}$: kth iteration of θ.

$k=0$, $\underline{\theta^0 = 0} = \sum_{i=1}^{m} 0 \cdot \phi(x^{(i)})$.

$k>0$, inductive hypothesis: $\theta^k = \sum_{i=1}^{m} \beta_i \phi(x^i)$

$$
\begin{aligned}
\theta^{k+1} &= (1-\alpha\lambda)\theta^k + \alpha \sum_{i=1}^{m} (y^i - \theta^T\phi(x^i))\phi(x^i) \\
&= (1-\alpha\lambda) \sum_{i=1}^{m} \beta_i \phi(x^i) + \alpha \sum_{i=1}^{m} (y^i - \theta^T\phi(x^i))\phi(x^i) \\
&= \sum_{i=1}^{m} \left[(1-\alpha\lambda)\beta_i \phi(x^i) + \alpha(y^i - \theta^T\phi(x^i))\phi(x^i) \right] \\
&= \sum_{i=1}^{m} \Big(\underbrace{(1-\alpha\lambda)\beta_i + \alpha(y^i - \theta^T\phi(x^i))}_{\beta^{k+1}} \Big)\phi(x^i) \qquad \square
\end{aligned}
$$

$\underline{\beta =} (1-\alpha\lambda)\beta_i + \underline{\alpha(y^i - \theta^T\phi(x^i))}$.

Kernel Regularized Least Square Regression

Proposition 1.

Parameter θ can be written as a linear combination of $\phi(x^{(1)}), \ldots, \phi(x^{(m)})$:

$$\theta_i = \sum_{i=1}^{m} \beta_i \phi(x^{(i)}) \text{ for all } i = 1, \ldots, m$$

Idea: do gradient descent on $\underline{\beta_1}, \ldots, \underline{\beta_m}$ instead of θ.

Kernel Regularized Least Square Regression

Proposition 1.

Parameter θ can be written as a linear combination of $\phi(x^{(1)}), \ldots, \phi(x^{(m)})$:

$$\theta_i = \sum_{i=1}^{m} \beta_i \phi(x^{(i)}) \text{ for all } i = 1, \ldots, m$$

Idea: do gradient descent on $\beta_1, \ldots, \beta_m$ instead of θ.

Gradient descent update for θ:

Kernel Regularized Least Square Regression

Proposition 1.

Parameter θ can be written as a linear combination of $\phi(x^{(1)}), \dots, \phi(x^{(m)})$:

$$\theta_i = \sum_{i=1}^{m} \beta_i \phi(x^{(i)}) \text{ for all } i = 1, \dots, m$$

Idea: do gradient descent on $\beta_1, \dots, \beta_m$ instead of θ.

Gradient descent update for θ:

$$\theta := (1 - \alpha\lambda)\theta + \alpha \sum_{i=1}^{m} (y^{(i)} - \theta^T \phi(x)^{(i)})\phi(x^{(i)})$$

$$= \sum_{i=1}^{m} \underbrace{\Big((1 - \alpha\lambda)\beta_i + \alpha(y^{(i)} - \theta^T \phi(x^{(i)})\Big)}_{\beta_i} \phi(x^{(i)})$$

Gradient descent update for $\beta_i, i = 1, \dots, m$:

Kernel Regularized Least Square Regression

Proposition 1.

Parameter θ can be written as a linear combination of $\phi(x^{(1)}), \ldots, \phi(x^{(m)})$:

$$\theta_i = \sum_{i=1}^{m} \beta_i \phi(x^{(i)}) \text{ for all } i = 1, \ldots, m$$

Idea: do gradient descent on $\beta_1, \ldots, \beta_m$ instead of θ.

Gradient descent update for θ:

$$\theta := (1 - \alpha\lambda)\theta + \alpha \sum_{i=1}^{m} (y^{(i)} - \theta^T \phi(x)^{(i)})\phi(x^{(i)})$$

$$= \sum_{i=1}^{m} \underbrace{\left((1 - \alpha\lambda)\beta_i + \alpha(y^{(i)} - \theta^T \phi(x^{(i)})\right)}_{\beta_i} \phi(x^{(i)})$$

Gradient descent update for $\beta_i, i = 1, \ldots, m$:

while not converged:

$\sum_{j=1}^{m} \beta_j \phi(x^{(j)})$

$$\beta_i := (1 - \alpha\lambda)\beta_i + \alpha(y^{(i)} - \underline{\theta^T \phi(x^{(i)})})$$

Use kernel function in the update for β_i :

$$\begin{aligned}
\beta_i &:= (1-\alpha\lambda)\beta_i + \alpha(y^{(i)} - \theta^T\phi(x^{(i)})) \\
&= (1-\alpha\lambda)\beta_i + \alpha(y^{(i)} - \sum_{j=1}^{m}\beta_j\phi(x^{(j)})^T\phi(x^{(i)})) \\
&= (1-\alpha\lambda)\beta_i + \alpha(y^{(i)} - \sum_{j=1}^{m}\beta_j k(x^{(j)}, x^{(i)}))
\end{aligned}$$

Use kernel function in the update for β_i :

$$\begin{aligned}
\beta_i &:= (1-\alpha\lambda)\beta_i + \alpha(y^{(i)} - \theta^T \phi(x^{(i)})) \\
&= (1-\alpha\lambda)\beta_i + \alpha(y^{(i)} - \sum_{j=1}^{m} \beta_j \phi(x^{(j)})^T \phi(x^{(i)})) \\
&= (1-\alpha\lambda)\beta_i + \alpha(y^{(i)} - \sum_{j=1}^{m} \beta_j k(x^{(j)}, x^{(i)}))
\end{aligned}$$

Vector form:

$$\underline{\beta := (1-\alpha\lambda)\beta + \alpha(y - K\beta)}$$

KRLS Summary

- Compute kernel matrix K for all m training samples

KRLS Summary

- Compute kernel matrix K for all m training samples
- Compute the optimal $\beta_1, \ldots, \beta_m$ through gradient descent or normal equation.

KRLS Summary

- Compute kernel matrix K for all m training samples
- Compute the optimal $\beta_1, \ldots, \beta_m$ through gradient descent or normal equation.
- Make prediction on new sample x:

$$
\begin{aligned}
\hat{y} &= \theta^T \phi(x) \\
&= \sum_{j=1}^{m} \beta_j \phi(x^{(j)})^T \phi(x) \\
&= \sum_{j=1}^{m} \beta_j K(x^{(j)}, x)
\end{aligned}
$$