(*) constrained optimization problem :-

8th. Feb 23

$f_0 : R^n \to R$

$\min f_0(x)$

such that

$f_i(x) \leq 0 \quad i \in [1, m]$

$f_j(x) = 0 \quad j \in [1, l]$

Lagrangian form of C.O.P

$$f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{l} \mu_j f_j(x)$$

original function

$\|Ax - b\|_2^2$ s.t. $\|x\|_2^2 \leq R$ $\Rightarrow$ $\|Ax - b\|_2^2 + \lambda \|x\|_2^2$ $\quad \lambda > 0$

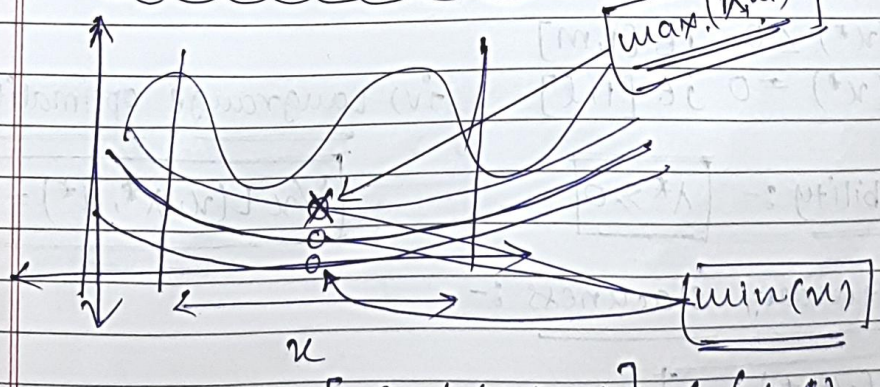
λ_i, μ_j is Lagrangian multiplier.

$$L(x, \lambda, \mu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{l} \mu_j f_j(x)$$

what if x is feasible point ?

$\Rightarrow$ For feasible x, $L(x, \lambda, \mu) \leq f_0(x)$

$\min_x L(x, \lambda, \mu) \leq f_0(x^*)$ s.t. $\lambda \geq 0$



$$\max_{\substack{\lambda, \mu \\ (\lambda, \mu) \in \text{domain}}} \left[\min_x L(x, \lambda, \mu) \right] \leq f_0(x^*)$$

8th Feb 2023

$$\min_x L(x, \lambda, u) = g(\lambda, u)$$ ⟸ dual function.

$$g(\lambda, u) = \min_x L(x, \lambda, u)$$

$$\max_{(\lambda,u)} g(\lambda, u) \leq f_0(x^*) \quad \lambda \geq 0$$

⇒ (λ^*, u^*) are point of ~~maximum~~ maxima

⇒ $g(\lambda, u) \leq g(\lambda^*, u^*)$

⇒ $g(\lambda, u) \leq g(\lambda^*, u^*) \leq f_0(x^*)$

maximum value of the dual function is ~~equal to~~ less than or equal to primal function.

$f_0(x^*) - g(\lambda^*, u^*) \geq 0$ } weak duality theorem

duality gap.

⇒ Primal Problem is convex problem. (usually) ⟹ strong duality hold (not always)

$f_0(x^*) - g(\lambda^*, u^*) = 0$

constraint should satisfy some condition

(*) KKT conditions :-

x^* ← point of primal optimal
λ^*, u^* ← point of dual optimal with zero duality gap.

(i) Primal Feasibility :-

$f_i(x^*) \leq 0 \quad i \in [1, m]$
$f_j(x^*) = 0 \quad j \in [1, l]$

(ii) Dual feasibility :- $\lambda^* \geq 0$

(iii) complementary slackness :-

$\lambda_i^* f_i(x^*) = 0 \quad \forall i$

(iv) Lagrange optimality :-

$\nabla_x L(x, \lambda^*, u^*) = 0$

8th Feb 2023

Any tuple (x, λ, u) is called KKT point
($x \in R^n$, $\lambda \in R^m$, $u \in R^l$)

if it satisfies the KKT conditions

when you have $(x^*, \lambda^*, u^*) \Rightarrow$ KKT point.

if duality gap = 0

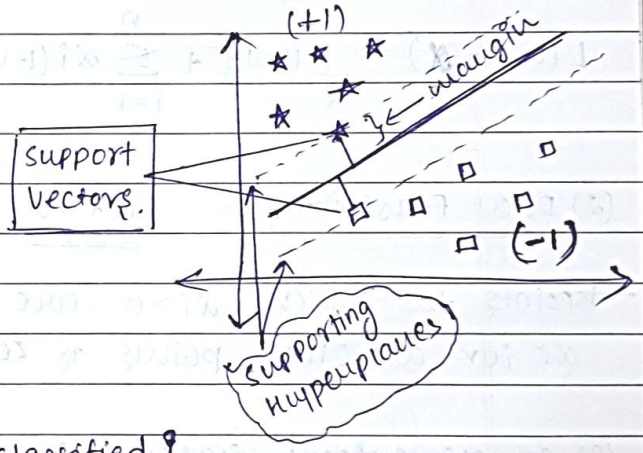
x^* } primal optimal
λ^* } dual optimal
u^* }

$\Rightarrow$ (x^*, λ^*, u^*) is KKT point

$\Leftarrow$ If Problem is convex

9th Feb. 2023

(*) Hard Margin SVM :-



$$\underset{(w, w_0)}{\text{Min}} \ \frac{1}{2}\|w\|_2$$

s.t

$$y_i(w^T x_i + w_0) \geq 1 \quad \forall i \ i \in [1, n]$$

- How new data point will be classified?

$\Rightarrow$ $x_{new} \leftarrow$ new data point.

$\rightarrow$ if $(w^T x_{new} + w_0)$ is positive then positive class.
else negative class...

- How to detect misclassification?

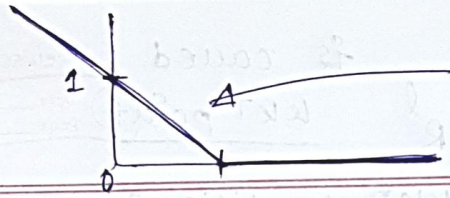
if $y_i(w^T x_i + w_0) < 1$ then point is misclassified...

$\hookrightarrow$ $0 < 1 - y_i(w^T x_i + w_0)$.

Hinge

$$l\left[(x, y), (w, w_0)\right] = \max\left(0, 1 - y_i(w^T x_i + w_0)\right)$$

9th Feb. 2023



Hinge LOSS

$$\min_{(w,w_0)} \frac{1}{2}\|w\|_2^2$$

$$\text{s.t. } y_i(w^T x_i + w_0) \geq 1 \quad \forall i \in [1,n]$$

} Primal form of Hard Margin.

= KKT conditions :-

(1) Primal Feasibility :-

$$1 - y_i(w^T x_i + w_0) \leq 0$$

w^*, w_0^*

Primal variables.

$$L(w, w_0, \alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i(w^T x_i + w_0))$$

(2) Dual Feasibility :- $\alpha_i \geq 0 \quad \forall i \in [1,n]$

→ points for which $\alpha_i > 0$ are the support vectors. α_i for all other points is zero.

(3) Complementary slackness :-

$$\alpha_i (1 - y_i(w^T x_i + w_0)) = 0 \quad \forall i \in [1,n].$$

(4) Lagrange Optimality

$$L(w, w_0, \alpha) = \frac{1}{2}\|w\|_2^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i(w^T x_i + w_0))$$

$$= \frac{1}{2} w^T w + \sum_{i=1}^{n} \alpha_i (1 - y_i(w^T x_i + w_0))$$

$$\nabla_w (w, w_0, \alpha) = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0$$

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

9th Feb 2023

$$\nabla_{w_0} L(w, w_0, \alpha) = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\therefore \boxed{\sum_{i=1}^{n} \alpha_i y_i = 0}$$

α_i for $i \in C^+$

α_j for $j \in C^-$

$$\boxed{\sum \alpha_i = \sum \alpha_j}$$

- **Support vectors :-**

$$\{x_i \mid \alpha_i^* > 0\}$$

$$\min_{(w, w_0)} \frac{1}{2} \|w\|_2^2$$

such that $1 - y_i(w^T x_i + w_0) \leq 0 \quad \forall i \quad i \in [1, n]$

$$L(w, w_0, \alpha) = \frac{1}{2} \|w\|_2^2 + \sum_{i=1}^{n} \alpha_i (1 - y_i (w^T x_i + w_0))$$

$$\boxed{g(\alpha) = \min_{(w, w_0)} L(w, w_0, \alpha)}$$

$$w_i^* = \sum_{i=1}^{n} \alpha_i y_i x_i \qquad w_0 \text{ satisfies } \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$g(\alpha) = \frac{1}{2} w^{*T} w_0^* + \sum_{i=1}^{n} \alpha_i \left(1 - y_i (w^{*T} x_i + w_0^*)\right)$$

$$= \frac{1}{2} w^{*T} w_0^* + \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \alpha_i y_i w^{*T} x_i - \sum_{i=1}^{n} \alpha_i y_i w_0^* \quad (= 0)$$

$$= \frac{1}{2} w^{*T} w_0^* + \sum_{i=1}^{n} \alpha_i - w^{*T} w^*$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} w^{*T} w^*$$

$$= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \left(\left(\sum_{i=1}^{n} \alpha_i y_i x_i \right)^T \left(\sum_{j=1}^{n} \alpha_j y_j x_j \right) \right)$$

9th Feb 2023

$$\max g(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

↑ Dual form of Hard margin SVM

s.t. $\alpha_i \geq 0$

$\forall i \sum_{i=1}^{n} \alpha_i y_i = 0$

10th Feb 23

(*) Kernel SVM :-

$x \in R^d$

$x \longrightarrow z \in R^h$

s.t. $n > d$

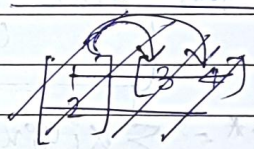
$x_i \Rightarrow z_i \Rightarrow \phi(x_i)$

$$\text{label}(x_{test}) = \text{sign}\left[w^{*T} \phi(x_{test}) + w_0^* \right]$$

$$w^* = \sum_{i=1}^{m} \alpha_i y_i \phi(x_i)$$

$$\text{label}(x_{test}) = \text{sign}\left[\sum_{i=1}^{m} \alpha_i y_i \phi(x_i)^T \phi(x_{test}) + w_0^* \right]$$

- How to implement SVM in higher dimension without actually getting $\phi(x)$:-



Mercer's Theorem :-

Kernel fun :-

$$K : X \times X \longrightarrow R$$

A symmetric function $K(x_i, x_j) = K(x_j, x_i)$ can be represented as $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ for some (ϕ).

IFF,

$$K = \begin{pmatrix} K(x_1,x_1) & K(x_1,x_2) & \cdots & K(x_1,x_n) \\ \vdots & \vdots & & \vdots \\ K(x_m,x_1) & K(x_m,x_2) & \cdots & K(x_m,x_n) \end{pmatrix}_{(m \times n)}$$

Gram Matrix.

- Example of Kernels :-

- Linear Kernel :-

$$K(x_i, x_j) = x_i^T x_j$$

classmate
Date
Page
10th Feb 2023

- Polynomial Kernel. :-

$$K(x_i, x_j) = (1 + x_i x_j)^t \; ; \; t > 0$$

- Gaussian Kernel. (RBF Kernel) :-

$$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|_2^2}{2\sigma^2}\right)$$

NOTE: A necessary and sufficient condition to make our custom kernel is that the Gram Matrix should be positive definite.

(*) Soft SVM :-

L1-SVM :- $\underset{(w, w_0, \xi_i)}{\text{Min}} \; \frac{\lambda}{2}\|w\|_2^2 + \frac{1}{2}\sum_{i=1}^{n} \xi_i$ (Regularization term)

L2-SVM :- $\underset{(w, w_0, \xi_i)}{\text{Min}} \; \frac{\lambda}{2}\|w\|_2^2 + \frac{1}{2}\sum_{i=1}^{n} \xi_i^2$

L1 C SVM :- $\underset{(w, w_0, \xi_i)}{\text{min}} \; \frac{C}{n}\sum_{i=1}^{n} \xi_i + \frac{\lambda}{2}\|w\|_2^2$

L2 C SVM :- $\underset{(w, w_0, \xi_i)}{\text{Min}} \; \frac{C}{n}\sum_{i=1}^{n} \xi_i^2 + \frac{\lambda}{2}\|w\|_2^2$

Example :- Let $x \in R$.

$$K(x_i, x_j) = \exp\left(\frac{-(x_i - x_j)^2}{2\sigma^2}\right)$$

$$= \exp\left(\frac{-x_i^2}{2\sigma^2} - \frac{x_j^2}{2\sigma^2} + \frac{2x_i x_j}{2\sigma^2}\right)$$

$$= \exp\left(\frac{-x_i^2}{2\sigma^2}\right) \cdot \exp\left(\frac{-x_j^2}{2\sigma^2}\right) \cdot \exp\left(\frac{2x_i x_j}{2\sigma^2}\right)$$

$$= \exp\left(\frac{-x_i^2}{2\sigma^2}\right) \exp\left(\frac{-x_j^2}{2\sigma^2}\right) \cdot \exp\left(1 + \frac{x_i x_j}{\sigma^2} + \frac{x_i^2 x_j^2}{2!(\sigma^2)^2} + \cdots\right)$$

10th Feb. 2023

$$= \exp\left(\frac{-x_i^2}{\sigma^2}\right) \cdot \exp\left(\frac{-x_j^2}{\sigma^2}\right) \cdot \left[1 \;\; \frac{x_i}{\sigma} \;\; \frac{x_i^2}{\sqrt{2}\sigma^2} \cdots\right] \left[1 \;\; \frac{x_j}{\sigma} \;\; \frac{x_j^2}{\sqrt{2}\sigma^2} \cdots\right]$$

$$= \exp\left(\frac{-x_i^2}{\sigma^2}\right) \left[1 \;\; \frac{x_i}{\sigma} \;\; \frac{x_i^2}{\sqrt{2}\sigma^2} \cdots\right] \cdot \exp\left(\frac{-x_j^2}{\sigma^2}\right) \left[1 \;\; \frac{x_j}{\sigma} \;\; \frac{x_j^2}{\sqrt{2}\sigma^2} \cdots\right]$$

$\phi(x_i)$ $\phi(x_j)$

(*) (*)

(*) **Low Rank Approximation :-**

$$\min_{\hat{A} \in R^{n \times d}} \|A - \hat{A}\|_F^2$$

subject to $\text{rank}(\hat{A}) = k$

$k << d$

$$A_k = \hat{A} = \sum_{i=1}^{k} \sigma_i u_i v_i^T$$

Solution

$\|Ux\|_2 = \|x\|_2$
$\|V^T x\|_2 = \|x\|_2$

↑ orthonormal.

(*) (*)

(*) **SMOTE :-** Synthetic Minority Over-Sampling Technique.

→ Avoid overfitting due to exact replicas of minority class samples.
→ Subset of minority class is taken.
→ New synthetic data samples ~~are~~ created from subsets and added.

:) No overfitting.
Reduced overfitting

:(mainly introduce noise.

(*) AdaBoost :- ~~A~~ classification

Boosting :- combining 'base' classifiers to form a "committee" whose performance is better than any of base classifiers.

classmate
Date
Page
Slide (8)
Deep learning
RNN

It works even for "weak learners".

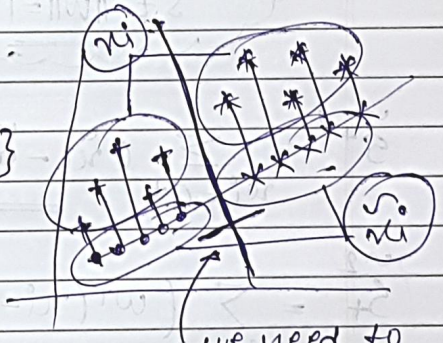
⇒ PCA only considers features into account.
But in supervised learning we have label also.

20th April.

(*) Fisher's LDA :-

• assumption :- Data coming from Gaussian distribtn.

~~Fisher's~~



we need to find

Divide data into two classes

$C^+ = \{x_i | y_i = +1\}$, $C^- = \{x_i | y_i = -1\}$

$|C^+| = n^+$ $\quad n = n^+ + n^-$

$|C^-| = n^-$

↑ Total no. of points

$\tilde{x}_i = w^T x_i \quad \forall i \in C^+ \cup C^-$

↑ projected

mean of both class.

$m^+ = \frac{1}{n^+} \sum_{x_i \in C^+} x_i$ $\qquad$ $m^- = \frac{1}{n^-} \sum_{x_i \in C^-} x_i$

$\tilde{m}^+ = \frac{1}{n^+} \sum_{\tilde{x}_i \in C^+} \tilde{x}_i$ $\qquad$ $\tilde{m}^- = \frac{1}{n^-} \sum_{\tilde{x}_i \in C^-} \tilde{x}_i$

$\tilde{m}^+ = w^T m^+$ $\qquad$ $\tilde{m}^- = w^T m^-$

we need to maximize the distance between mean of two classes.

$\therefore$ maximize $|\tilde{m}^+ - \tilde{m}^-|$

$\therefore$ maximize $|w^T(m^+ - m^-)|$

$$\|\tilde{m}^+ - \tilde{m}^-\|^2 = \|w^T(m^+ - m^-)\|^2$$

$$= w^T(m^+ - m^-)(m^+ - m^-)^T w$$

$(m^+ - m^-)(m^+ - m^-)^T$ ↑ matrix (rank = 1)

S_B (Between class scatter matrix)

$$= w^T S_B w$$

maximize $w^T S_B w$ ⟸ problem

s.t $\|w\| = 1$

$$\tilde{s}^{+2} = \sum_{x_i \in C^+} (\tilde{x}_i - \tilde{m}_i^+)^2 \qquad \tilde{s}^{-2} = \sum_{x_i \in C^-} (\tilde{x}_i - \tilde{m}_i^-)^2$$

$$\tilde{s}_+^2 = \sum_{x_i \in C^+} \left(w^T(x_i - m^+)\right)^2$$

scatter for projected classes.

$$\tilde{s}_+^2 = \sum_{x_i \in C^+} w^T(x_i - m^+)(x_i - m^+)^T w$$

$$= w^T \left[\sum_{x_i \in C^+} (x_i - m^+)(x_i - m^+)^T\right] w$$

↑ sum of rank 1 matrix.

$$\tilde{s}_+^2 = w^T S^+ w$$

↑ scatter matrix for +ve class.

similarly, $\tilde{s}^{-2} = w^T S^- w$

$$\tilde{s}^{+2} + \tilde{s}^{-2} = w^T (S^+ + S^-) w$$

$$= w^T (S_W) w$$

↑ within class scatter matrix.

Fisher's Linear Discriminant criteria,

$$J(W) = \frac{W^T S_B W}{W^T S_W W} \quad ; \quad \mathbb{R}^d \to \mathbb{R}$$

$$W_{Fisher} = \underset{W}{\arg\max}\ J(W)$$

$$\frac{\partial J(W)}{\partial W} = \frac{(W^T S_B W).(S_B W) - (W^T S_B W)(S_B W)}{(W^T S_W W)^2}$$

$$\left.\frac{\partial J(W)}{\partial W}\right|_{W^*} = \frac{W^T S_W W . S_B W - W^T S_B W . S_W W}{(W^T S_W W)^2} = 0$$

$$(W^{*T} S_W W^*) S_B W^* = (W^{*T} S_B W^*) S_W W^*$$

$J(W^*)$

$$S_B W^* = \frac{(W^{*T} S_B W^*) S_W W^*}{(W^{*T} S_W W^*)}$$

$$S_B W^* = \underbrace{J(W^*)}_{\text{scalar}} S_W W^*$$

(Similar to $Av = \lambda Bv$)

sowing this eqn is generalised eigen value problem.

A and B are Symmetric matrices

$$S_B v = (m_+ - m_-)\underbrace{(m_+ - m_-)^T v}_{\lambda}$$

$$S_B v = \lambda (m_+ - m_-)$$

$$J(W^*) S_W W^* \propto (m_+ - m_-)$$

$$S_W W \propto (m_+ - m_-)$$

if S_W is invertible matrix then

$$W^* \propto S_W^{-1} (m_+ - m_-)$$

24th April

* Online learning *

→ Traditional ML Methods assume that entire dataset D is available before training starts. In this case we perform "Batch learning."

→ In many cases data arrives sequentially

→ Let $\hat{\theta}_{t-1}$ is our parameter for given data points from time 1, 2, ..., t-1. We want to update our parameter in constant time when we see the t^{th} point we have to find rule.