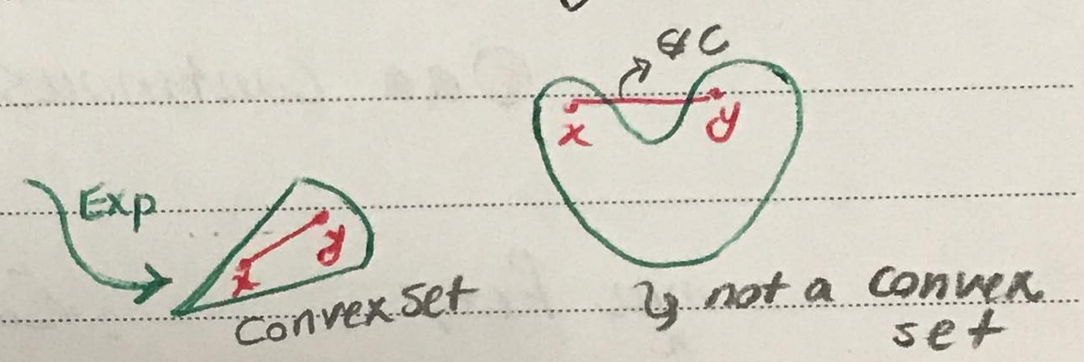
جلسه دهم دکتر نستری : ۲۵ آبان

در این جلسه در مورد convexity صحبت کردیم و اینکه مزیت اصلی تابع convex این است که بهینهٔ global دارد و دارای local optimum نیست اما در مورد توابع concave این اتفاق نمی‌افتد. حال ابتدا مجموعهٔ محدب را تعریف می‌کنیم و سپس به سراغ تعریف تابع محدب می‌رویم.

convex set: A set $\underline{C}$ is a convex set if $x, y \in C$ and

$0 \leq \theta \leq 1 \Rightarrow \theta x + (1-\theta) y \in C$

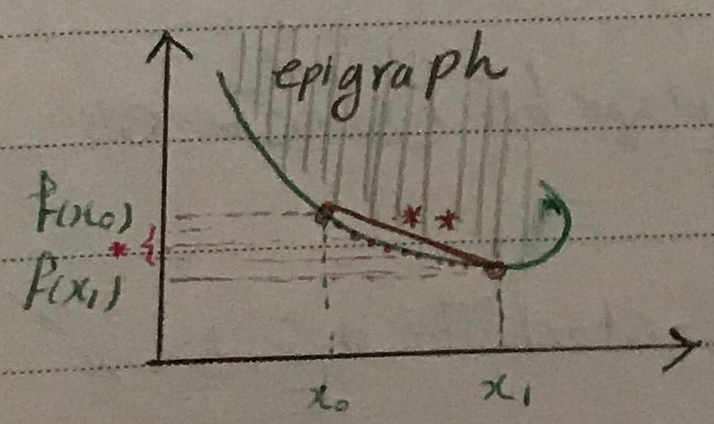


در واقع یعنی هر دو نقطه‌ای که در مجموعه در نظر گرفتیم را اگر با یک خط به هم متصل کنیم کل خط داخل مجموعه باشد.

این مجموعه‌ها نواحی هستند که توسط قیدهای مسئله نشان داده می‌شوند.

convex function: A function F is a convex function if for $x, y \in D_F$

and $0 \leq \theta \leq 1$: $\overbrace{f(\overbrace{\theta x + (1-\theta) y})}^{*} \leq \overbrace{\theta f(x) + (1-\theta) f(y)}^{**}$



یعنی مقادیر تابع برای نقاط روی خط $\theta x + (1-\theta) y$ همگی کمتر باشند از خط بین $f(x)$ و $f(y)$

epigraph یک تابع محدب یک مجموعهٔ محدب است.

Subject:

Year. Month. Date. ( )

Remarks. A concave function is the negative of a convex function.

Theorem: if $f(x)$ is twice differentiable, $f(x)$ is convex if and only if Hessian $\nabla^2 f$ is posetive semi definite

for convex function, all local minimum are global.

Continous optimization

$\min_x f(x)$ حالت اول) بهینه‌سازی بدون داشتن هیچ قیدی است:

EX1. $\min_x x^4 + 2x^2 - 12x + 3$

EX 2. a quadratic function:

$$\min_{x_1, x_2} \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 2 \\ 2 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$f(x) = \frac{1}{2} x^T Q x + c^T x \Rightarrow \nabla f = Qx + c$$

برای بدست آوردن نقاط بهینه این مساله ما نظر که قبلاً صحبت کردیم میتوان از روشهای gradient descent

که میتوانیستم از stochastic و batch استفاده کنیم و همینطور از روش نیوتن استفاده کنیم.

→ constrain optimization

در این حالت مسئله بهینه سازی مورد نظر دارای محدودیت های مساوی یا نامساوی یا هر دو است:

$$\min_x f(x)$$

s.t $g_i(x) \leq 0 \quad ; i=1,\dots,m \Rightarrow$ m قید نامساوی داریم

$h_j(x) = 0 \quad ; j=1,\dots,t \Rightarrow$ t قید تساوی داریم

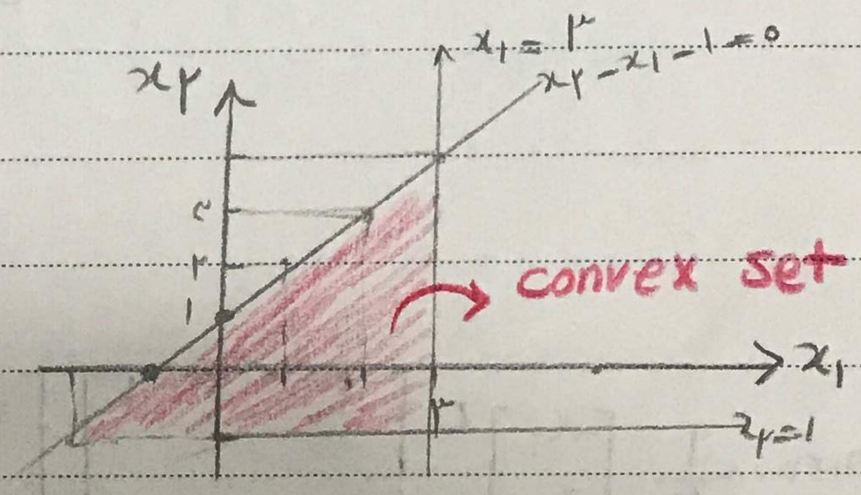
EX1. $\min \; 2x_1 + x_2$

s.t $x_1 \leq 3$

$x_2 \geq -1$

$-x_1 + x_2 - 1 \leq 0$

(a linear program)



✓ مسئله بهینه سازی محدب، مسئله ای است که f و g محدب هستند و $h_j(x) = 0$ یک مجموعه محدب باشد. در دسته خیلی مهم این مسائل بهینه سازی محدب دارند: ← linear ، آنهایی که f و g و h خطی اند

quadratic ←

یک مسئله بهینه سازی با تمام قیدهایش را میتوان به فرم زیر نوشت:

$$\min \; c^T x$$

s.t $Ax \leq b$

$A \in \mathbb{R}^{m \times d}$

$b \in \mathbb{R}^{m}$ ~ تعداد قیدها

$x \in \mathbb{R}^{d}$

* برای مثال این فرم برای مثال EX1 به شکل زیر بدست میاید:

Subject:

Year. Month. Date. ( )

$$c^T x = \begin{bmatrix} 2 \\ 1 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \Rightarrow 2x_1 + x_2$$

$$A = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & -1 \\ -1 & 1 \end{bmatrix}}_{3\times 2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq b = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

اگر قید مساوی داشتیم آن را تبدیل به $\leq$ و $\geq$ میکنیم و دوباره این ماتریس را ایجاد میکنیم.

- Quadratic programming

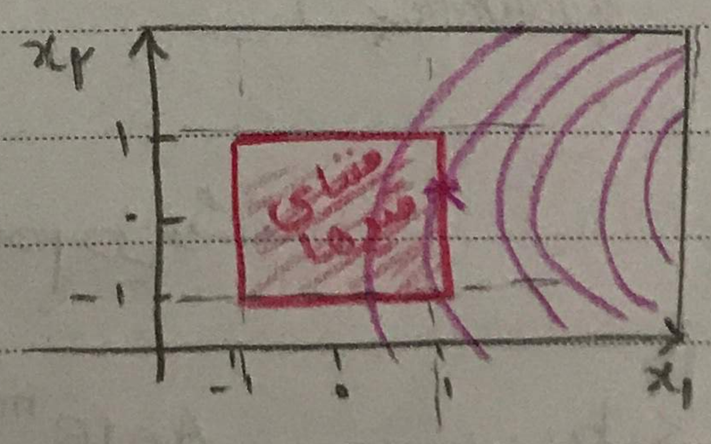
$$\min_x \frac{1}{2} x^T \underbrace{Q}_{PD} x + c^T x$$

$$s.t \quad Ax \leq b$$

$$\min_x \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 3 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

s.t

$$\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \qquad \begin{cases} x_1 \leq 1 \\ -x_1 \leq 1 \rightarrow x_1 \geq -1 \\ x_2 \leq 1 \\ -x_2 \leq 1 \rightarrow x_2 \geq -1 \end{cases}$$



حال به سراغ این مبحث رفتیم که چگونه یک مسئله بهینه‌سازی با constrain را تبدیل به یک مسئله بدون constrain کنیم:

Subject: 71

Year. Month. Date. ( )

Consider $\min_x f(x)$

s.t $g_i(x) \leq 0 \quad ; \; i=1,\dots,m$

Simple (crazy) idea

$$J(x) = f(x) + \sum_{i=1}^{m} \delta(g_i(x)) \Rightarrow \min_x J(x)$$

$$\delta(z) = \begin{cases} 0 & z \leq 0 \\ \infty & \text{o.w} \end{cases} \qquad J(x) = \begin{cases} f(x) & ; \text{all constrain are satisfy} \\ \infty & ; \text{o.w} \end{cases}$$

Lagrangian:

idea: replace δ with a linear function

$$L(x, \alpha) = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x) \longrightarrow \begin{cases} \text{if } g_i(x) \leq 0 \Rightarrow f(x) \\ \text{if } g_i(x) > 0 \Rightarrow \infty \end{cases}$$

$(\alpha_1, \dots, \alpha_m)$

$$J(x) = \max_{\alpha_i \geq 0} L(x, \alpha)$$

ضریب لاگرانژ

Simple case:

$$f(x) + \alpha_1 g_1(x) \begin{cases} \text{if } g_1(x) \leq 0 \xrightarrow{\alpha_1 \geq 0} \alpha_1 = 0 \\ \max L(x, \alpha) = f(x) \\ \text{if } g_1(x) = 1 \Rightarrow \text{بی‌نهایت می‌کنیم} \\ \max L(x, \alpha) = \infty \end{cases}$$

← همان تعریف $J(x)$ است.

Lagrangian duality

optimization x $\Rightarrow$ optimization in α

Sunwood

Subject:

Year. Month. Date. ( )

primal problem $\longrightarrow$ dual problem

x: primal variebles $\qquad$ α: dual variables

- Primal problem:

$$\min_{x} L_p(x)$$

$$L_p(x) = \max_{\alpha \geq 0} L(x, \alpha)$$

$$\Rightarrow \min_{x} \max_{\alpha \geq 0} L(x, \alpha)$$

The same solution as the orginal problem.

- dual problem:

$$\max_{\alpha \geq 0} L_D(\alpha)$$

$$L_D(\alpha) = \min_{x} L(x, \alpha)$$

$$\Rightarrow \max_{\alpha \geq 0} \min_{x} L(x, \alpha)$$

□ Relationship between primal and dual problem.

$$d^* = \max_{\alpha \geq 0} \min_{x} L(x, \alpha)$$

$$p^* = \min_{x} \max_{\alpha \geq 0} L(x, \alpha)$$

$\Rightarrow$ always $d^* \leq p^*$

اگر توابع f و g محدب باشند و برقراری شرایط (مثلاً KKT یا ...) برایشان برقرار باشد داریم: $p^* = d^*$

Sunwood

و به این حالت Strong duality می‌گویند.

$$p^* = d^* = L(\alpha^*, x^*)$$

dual solution ↩ ↪ primal solution

مفهوم دوگان یک مسئله زمانی که تعداد محدودیت ها (constraint) کمتر از تعداد ویژگی ها (Feature) باشد آسانتر است.

Subject:

Year. Month. Date. ( )

جلسه نهم (دکتر منتظری)

همانطور که در جلسه قبل صحبت کردیم هدف حل مسئله بهینه‌سازی بوده است:

$$\min f(x)$$

$$\text{s.t} \quad g_i(x) \leq 0 \quad ; \quad i = 1, \dots, m$$

$$\boxed{h_j(x) = 0 \quad ; \quad i = 1, \dots, k} \rightarrow$$ این قیدها برای سادگی حذف میشوند

حال برای این مسئله لاگرانژین را محاسبه می‌کنیم:

$$L(x, \alpha) = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x) = f(x) + \underbrace{\alpha^T g(x)}_{\text{vector}}$$

در ادامه در مورد Lagrangian duality صحبت کردیم:

$$\max L_D(\alpha)$$

$$L_D(\alpha) = \min_x L(x, \alpha) \xrightarrow[\text{duality}]{\text{strong}} p^* = d^* = L(x^*, \alpha^*)$$

حال باید بدانیم چگونه میتوان مزدوج دوگان یک مسئله را مشخص کرد:

□ Linear programming (revisited)

$$\min_x c^T x$$
$$\text{s.t} \quad Ax \leq b$$

Lagrangian: $L(x, \alpha) = c^T x + \alpha^T (Ax - b)$

$$= (c + A^T \alpha)^T x - \alpha^T b$$

dual problem: $\min_x L(x, \alpha) \Rightarrow \dfrac{\partial L(x, \alpha)}{\partial x} = (c + A^T \alpha) = 0$ *

Subject:

Year. Month. Date. ( )

با جایگذاری * در فرم لاگرانژین به فرم $L_D(\alpha)$ می‌رسیم:

$$L_D(\alpha) = -\alpha^T b = -b^T \alpha$$

$$\Rightarrow \max_{\alpha} L_D(\alpha) = \max (-b^T \alpha)$$

$$s.t \quad \alpha \geq 0$$

$$C + A^T \alpha = 0$$

پس اگر این مسئله را حل کنیم اگر همان مسئله اولیه را حل کردیم.

نکته: اینکه در یک مسئله بهتر است فرم اولیه یا دوگان حل شود به تعداد وزن‌ها و همینطور تعداد محدودیت‌ها وابسته است و اگر محدودیت‌ها از متغیرها کمتر باشد بهتر است مسئله دوگان حل شود و بالعکس.

تمرین: حل مسئله دوگان برای QP.

□□ KKT conditions:

✓ In unconstraint optimization, the optimal solution x^*

$$\Rightarrow \nabla L(x^*) = 0$$

✓ KKT condition given similar condition for constraint optimization

KKT condition: ① primal constraint : $g_i(x^*) \leq 0 \; ; i = 1, \dots, m$

② dual constraint : $\alpha_i^* \geq 0 \; ; i = 1, \dots, m$

Subject:

Year. Month. Date. ( )

③ complementray slackness : $\alpha_i^* g_i(x^*) = 0 \quad ; i=1,\dots,m$

④ Stationary point : $\nabla f(x^*) + \sum_{i=1}^{m} \alpha_i^* g_i(x^*) = 0$

if x^* and α^* are primal and dual solutions $\Rightarrow$

شرایطی که باید داشته باشند این است که strong duality برقرار باشد و f و g مشتق پذیر باشند

x^* and α^* satisfy KKT condition.

← به طور کلی به این شرایطی که باید برقرار باشند qualification condition میگویند مثلاً شرایط Slater.

□ sufficiency of KKT :

$\tilde{x}$ and $\tilde{\alpha}$ satisfy KKT conditions $\Rightarrow$ $\tilde{x}$ and $\tilde{\alpha}$ are primal and dual optimal point

$\boxed{\text{f and g convex}}$

□ practical implications:

+ convergence check
+ Some α_i are zero due to KKT (non_binding conditions)

○○○ coordinate wise optimization:

یک روش بهینه سازی میگیرد که به نوعی برای حل مشکل دوگان در SVM از این الگوریتم استفاده میشود.

2→ subgradient :

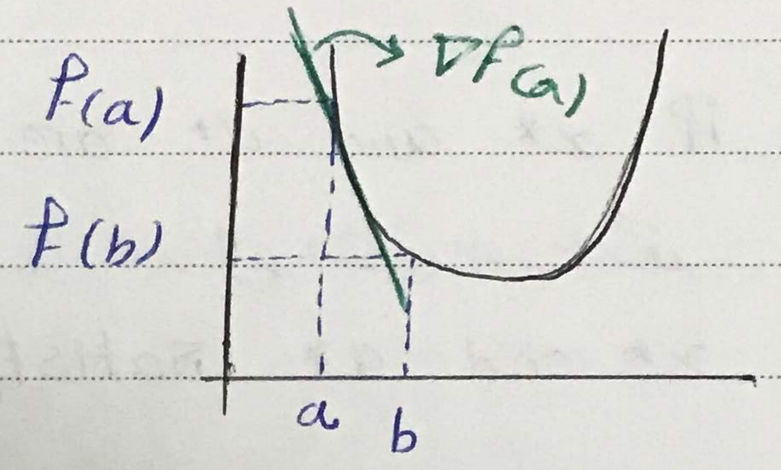
Sunwood

Subject:

Year. Month. Date. ( )

تعمیمی از gradient است برای زمانی که تابع در یه سری از نقاط مشتق پذیر نباشد.

اگر تابعی به شکل زیر را داشته باشیم آنگاه همواره رابطهٔ زیر را داریم:



$$f(b) \geq f(a) + \nabla f(a)^T (b-a)$$

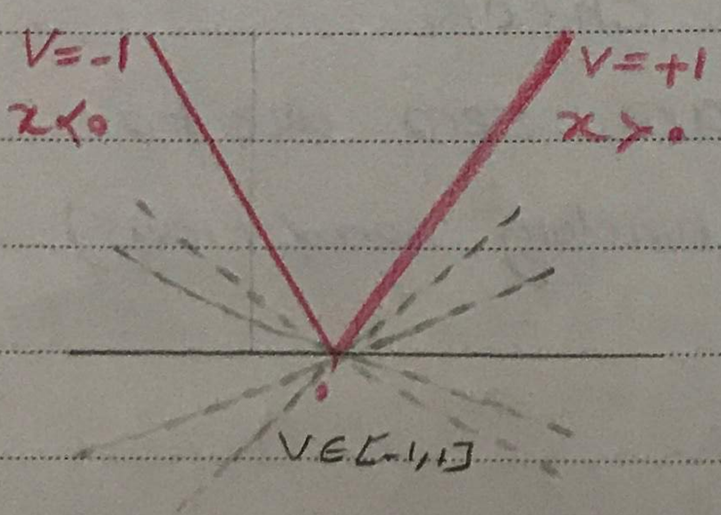
linear approximization alwayse understatment F

- subgradient definition:

"is a generalization of gradient to non-differentiable point."

subgradient of a convex f is any $v \Rightarrow f(b) \geq f(a) + v(b-a)$

مثال ۱: تابع قدرمطلق:

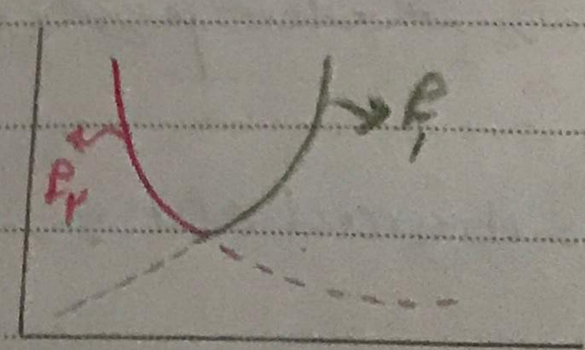


$$\Rightarrow \partial f(x) = \begin{cases} 1 & ; x>0 \\ [-1,1] & ; x=0 \\ -1 & ; x<0 \end{cases}$$

مثال ۲: توابع f_1 و f_2 توابعی محدب و مشتق پذیرند ←

$$f(x) = \max(f_1(x), f_2(x))$$

$$\partial f(x) = \begin{cases} \nabla f_1(x) & ; f_1(x) > f_2(x) \\ \text{any point on line between } \nabla f_1(x) \text{ and } \nabla f_2(x) & ; f_1(x) = f_2(x) \\ \nabla f_2(x) & ; f_2(x) > f_1(x) \end{cases}$$



Subject:

Year. Month. Date. ( )

Coordinate descent algorithm:

الگوریتم ساده و کارایی هستند که با ابعاد و پیچیدگی زیاد هم میتوانند کار کنند.

we want to solve $\min_{B} L(B_1, B_2, \dots, B_p)$

→ gradient decent این مسئله را میتوان با روشهای نیوتن یا حل کرد.

Algorithm:

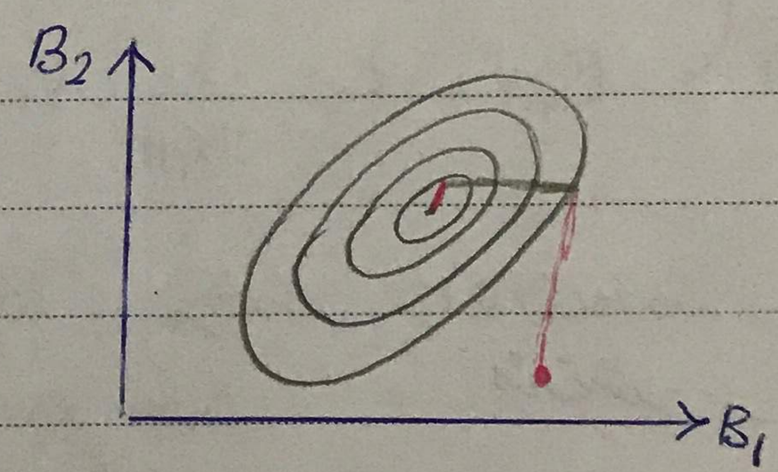
iterate until covergence {

for $i = 1, \dots, p$

$B_i := \arg\min_{b} L(B_1, \dots, B_{i-1}, b, B_{i+1}, \dots B_p)$

}

→ keep all other variebles fixed



△ algorithm result in coordinate-wise minimum ⇒ global minimum

convex and differentiable

also some non-differentiable (seperable non-smooth part)

point: one at the time not all at once.

△ Linear regression:

$$\min L(B) = \frac{1}{2}(Y - XB)^T(Y - XB)$$

$Y \in \mathbb{R}^n$

$B \in \mathbb{R}^p$

$X \in \mathbb{R}^{n \times p}$

$x_1, x_2, \dots, x_p$ columns of X

Subject:

Year. Month. Date. ( )

$$\frac{\partial L}{\partial B_i} = 0 \Rightarrow -X_i^T(Y - XB) = -X_i^T(X_i B_i + X_{-i} B_{-i} - Y) = 0$$

$$\Rightarrow B_i = \frac{X_i^T(Y - X_{-i} B_{-i})}{X_i^T X_i} = C_i$$

سرعت همگرایی CD خیلی بیشتر از سرعت همگرایی GD است.

LASSO regression:

$$\min L(B) = \frac{1}{2}(Y - XB)^T(Y - XB) + \lambda \sum_{i=1}^{p} |B_i|$$

$$\frac{\partial L}{B_i} = X_i^T X_i B_i + X_i^T(X_{-i} B_{-i} - Y) + \lambda S_i$$

$$S_i \in \partial |B_i| := \begin{cases} -1 & ; B_i < 0 \\ [-1,1] & ; B_i = 0 \\ +1 & ; B_i > 0 \end{cases}$$

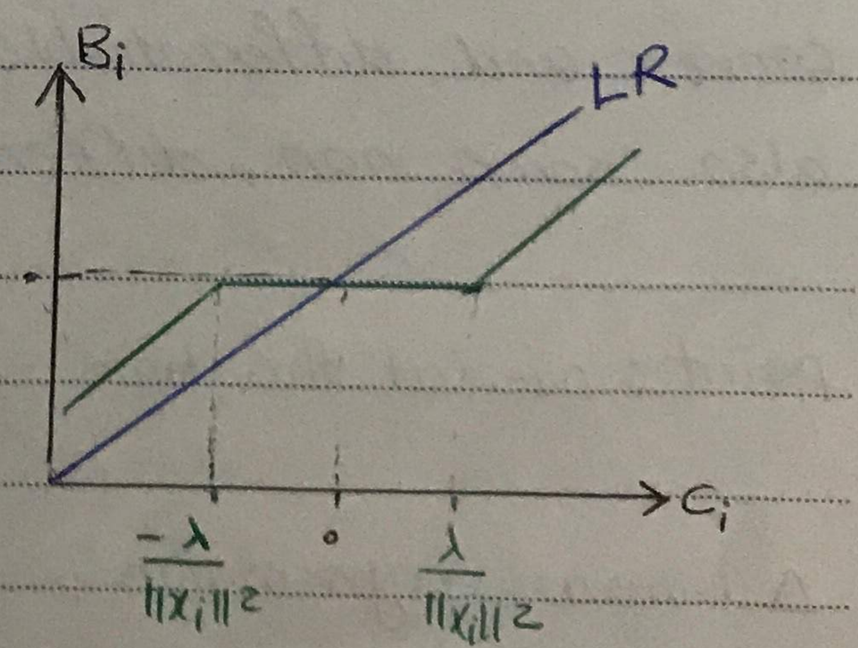
$$\xrightarrow{\frac{1}{X_i^T X_i}} B_i - C_i \frac{\lambda}{\|X_i\|^2} S_i = 0$$

Solution :$\Rightarrow$ $B_i = \delta_{\frac{\lambda}{\|X_i\|^2}}(C_i)$

اثبات شد.

soft thresholding function

$$S_T(v) := \begin{cases} v - T & ; v > T \\ 0 & ; -T \leq v \leq T \\ v + T & ; v < -T \end{cases}$$



proof:

$$B_i - C_i \frac{\lambda}{\|X_i\|^2} S_i = 0$$

Subject:

Year. Month. Date. ( )

Case 1 $B_i > 0 \Rightarrow S_i = 1 \Rightarrow B_i = C_i - \frac{\lambda}{\|x_i\|^2}$

if $C_i > \frac{\lambda}{\|x_i\|^2}$

Case 2 $B_i = 0 \Rightarrow S_i \in [-1, 1] \Rightarrow C_i = \frac{\lambda}{\|x_i\|^2} [-1, 1]$

if $\frac{-\lambda}{\|x_i\|^2} < C_i < \frac{\lambda}{\|x_i\|^2}$

Case 3 $B_i < 0 \Rightarrow S_i = -1 \Rightarrow B_i = C_i + \frac{\lambda}{\|x_i\|}$

if $C_i < \frac{\lambda}{\|x_i\|^2}$

"جلسه دهم"

The sequential minimal optimization (SMO)

الگوریتمی است که با استفاده از آن مسئله دوگان SVM را حل می‌کنیم. مسئله دوگان SVM به فرم زیر بوده است:

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

s.t $\quad 0 \leq \alpha_i \leq C \longrightarrow$ box constraint : به این شرط‌ها می‌گوییم

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

← الگوریتم SMO شبیه به الگوریتم coordinate ascent است.

(blockwise coordinate ascent)

ایده این الگوریتم به این صورت است که به جای یک α، دو α را update می‌کنیم:

$\Rightarrow$ update two α's at a time

Subject:

Year. Month. Date. ( )

SMO Algorithm: Repeat until coverage {

1. select a pair α_i & α_j to update

2. $\alpha_i, \alpha_j = \underset{\alpha_i, \alpha_j}{\arg\max}\ W(\alpha)$

↳ keep other α's fixed

}

در SVM شرطی داشتیم که میگفت:

* $\alpha_1 y^1 + \alpha_2 y^2 + \dots + \alpha_n y^n = 0$

$\Rightarrow \alpha_1 = -(\alpha_2 y^2 + \dots + \alpha_n y^n) / y^{(1)}$

چون α_1 طبق بقیه αها بدست میاریم پس باید در هر گام دو α را update کنیم:

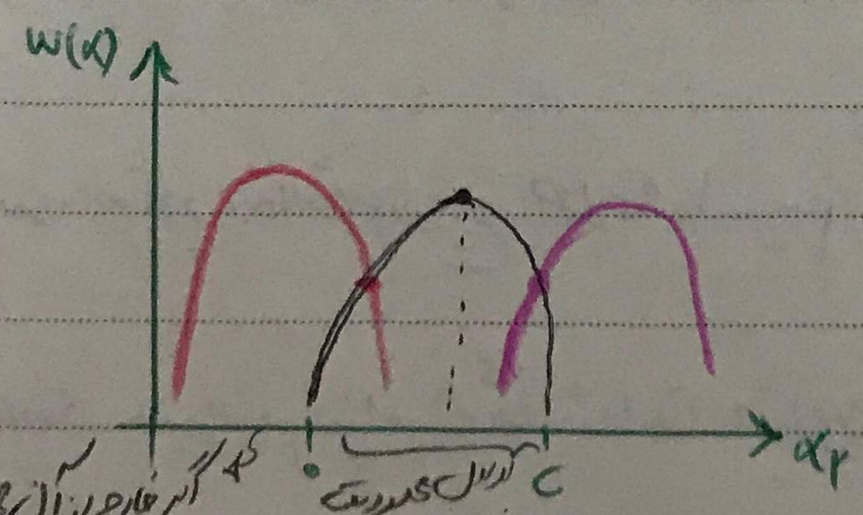
فرض کنیم هدف update کردن α_1 و α_2 باشد:

$\xrightarrow{(*)} \alpha_1 y^{(1)} + \alpha_2 y^{(2)} = k$

$\Rightarrow \alpha_1 = (k - \alpha_2 y^{(2)}) \cdot y^{(1)}$

$W(\alpha_1, \alpha_2, \dots, \alpha_n) = W((k - \alpha_2 y^{(2)}) \cdot y^{(1)}, \alpha_2, \dots, \alpha_n)$

* چون تمام αها بجز α_1 و α_2 ثابت هستند و α_1 را هم براساس α_2 نوشتیم پس W یک تابع quadratic در α_2 است.



Sunwood