ML_11

Constrained optimization

constrained optimization مبنای بسیاری از روش های یادگیری ماشین است.

مثلاً در SVM یا PCA، linear programming، metabolic networks، ....

یک روش کلی optimize کردن است مشابه سایر روش های unconstrained opt

(مثل gradient descent، Newton، ...)

* ابتدا لازم است با یک سری تعاریف ریاضی آشنا شویم:

در opt هایی که انجام می دهیم، convexity خیلی مهم است.

convex optimization → is important → because it is a global answer.
(global optimum)

Convex set

Def: A set C is a convex set if any $x, y \in C$ & $0 \leq \theta \leq 1$.

$$\Downarrow$$

$$\boxed{\theta(x) + (1-\theta) Y \in C}$$

Example: A convex set

A non_Convex set

θ اگر یک باشد نقطه‌ی جدید روی X است.

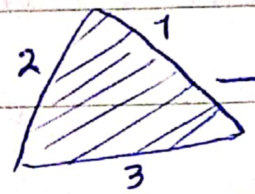
θ بین صفر و یک بین آنها.

θ برابر صفر ← روی Y

استفاده‌ی convex set در ML چیست؟

وقتی می‌خواهیم یک search در یک محدوده‌ی خاص انجام دهیم، convex set ابزاری است که امکان label زدن آن region را می‌دهد و این مجموعه‌ی constrained محدوده‌ای از فضای search را مشخص می‌کند.

Usually region denoted by constraints.



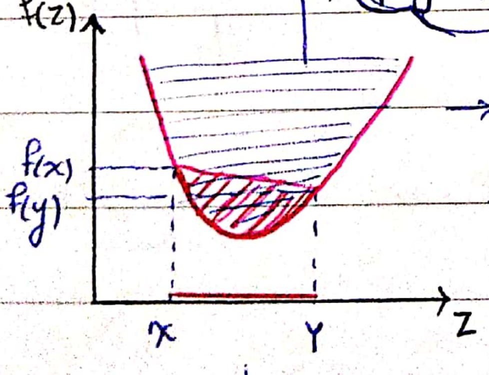
→ this shape has 3 constraint

Convex function

Def: a function f is a convex function if for all x, y in the domain of f and $0 \leq \theta \leq 1$, we have:

$$\boxed{f(\theta x + (1-\theta) y) \leq \theta f(x) + (1-\theta) f(y)}$$

epigraph → فضای بالای گراف است که اگر گراف convex باشد آن هم convex set است



→ اگر f هر نقطه‌ای بین x و y حساب شود، کوچکتر از خطی باشد که f(x) را به f(y) وصل می‌کند، آن تابع convex است.

→ bowl-like → you can find min very easily!

concave function

Remark: A concave function is the negative of a convex f.

↳ you can find max very easily.

از توابع convex و concave می‌توان به راحتی برای یافتن min و max استفاده کردیم. مسئله‌ای که وجود دارد توابع non-convex یا non-concave هستند.

مثال: squared loss function یک convex function است که هدف هم محاسبه‌ی min تابع در linear regression بود.

یا مثلاً در likelihood، function به گونه‌ای design می‌کنیم که convex یا concave باشد.

روش‌هایی هم در بیوانفورماتیک داریم مثل Neural net که non-convex است ولی عموماً دنبال این هستیم که design را روی توابع convex یا concave انجام دهیم.

* اگر تابع convex باشد، همه‌ی local min ها global min هم هستند.

Theorem:

if $f(x)$ is twice differentiable, $f(x)$ is convex, if and only if: Hessian $\nabla^2 f$ is PSD.

مثال: $x^2 \rightarrow \Delta^2 f = 2^{+}$

/ /

<u>importance:</u>

+ for convex functions, all local min are global min.

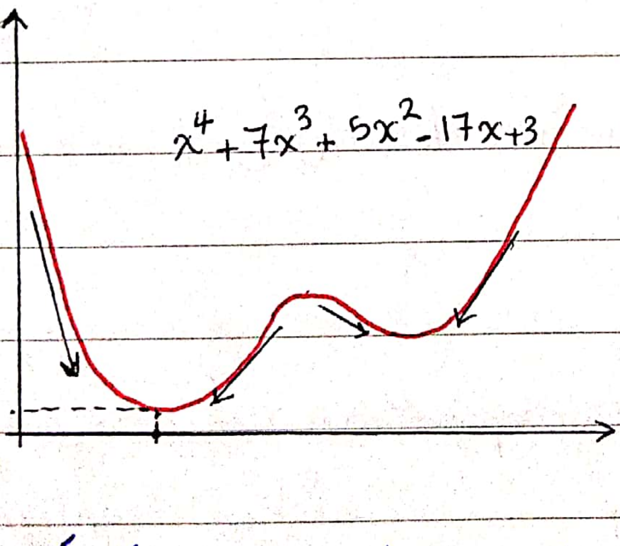
+ easy to optimize.

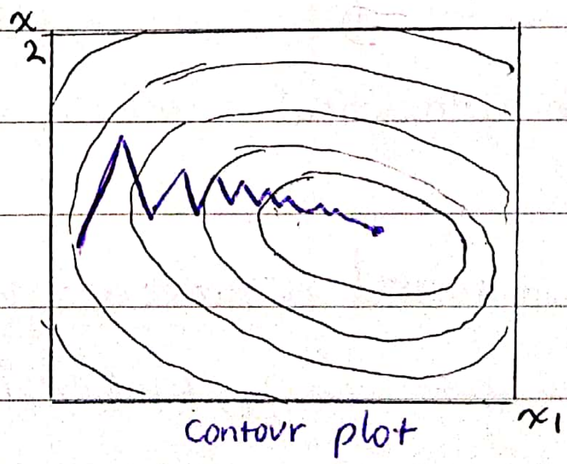
+ objective functions in ML → <u>designed to be convex.</u>

---

continues optimization (حقیقی) →
1) Unconstrained opt
2) constrained opt

Review- Unconstrained optimization





contour plot

در واقع نمودار سمت چپ در opt دیده نمی‌شود و در هر نقطه مشخص می‌شود که به چه سمتی باید حرکت کنیم. و حرکت در خلاف جهت گرادیان است و مثلاً فقط می‌دانیم نرم تابع convex است پس حرکت می‌کنیم تا به min برسیم. اما اگر مانند شکل سمت چپ non-convex باشد.

$\min f(x)$

ex1: $\min\limits_{x} \; x^4 + 5x^2 - 7x + 3 \xrightarrow{\text{مشتق‌گیری}} \dots \Longrightarrow$ min در یک بعد

ex2: a quadratic function:

$$\min_{x_1, x_2} f(x_1, x_2) = \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}^T \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 5 \\ 3 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$\Downarrow$ حداکثر درجه‌ی x_1 و x_2 درجه $\underline{\underline{2}}$ است.

$$\boxed{f(x_1, x_2) = \frac{1}{2} x^T Q x + c^T x}$$ quadratic form

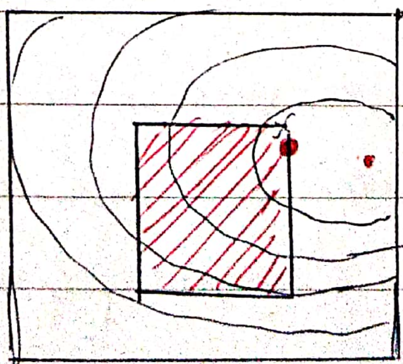
$\downarrow$

$$\boxed{\nabla f = Qx + c}$$

می‌توان با روش‌های gradient descent، Newton یا Analytical، min کرد.

Constrained optimization

دنبال پیدا کردن یک min هستیم با این فرض که جواب‌ها فقط می‌توانند در یک ناحیه‌ی highlight شده باشند.



پس در آن ناحیه نقطه‌ی optimal را پیدا می‌کنیم، یعنی نقطه‌ای که بیشترین مقدار را نسبت به objective fun دارد.

Scanned with CamScanner

LASSO دو جزء داشت که در یک جزئش در واقع constrainted opt بود.

$$\min_{x} f(x)$$
$$\text{s.t} \begin{cases} g_i(x) \leqslant 0 \\ h_j(x) = 0 \end{cases}$$
for $i = 1, 2, \ldots, m$

for $j = 1, 2, \ldots, t$

$\rightarrow$ constrained ها دو نوع هستند:

1) نامساوی که به کوچکتر مساوی تبدیل می‌شود.

2) مساوی

$\Downarrow$

خیلی اهمیت ریاضی ندارد، چون می‌توان آن را به دو نامساوی هم تبدیل کرد.

$$h_j(x) = 0 \rightarrow \begin{cases} h_j(x) \leqslant 0 \\ h_j(x) \geqslant 0 \end{cases}$$

قبلاً با constrained مساوی در lagrangian آشنا شده بودیم.

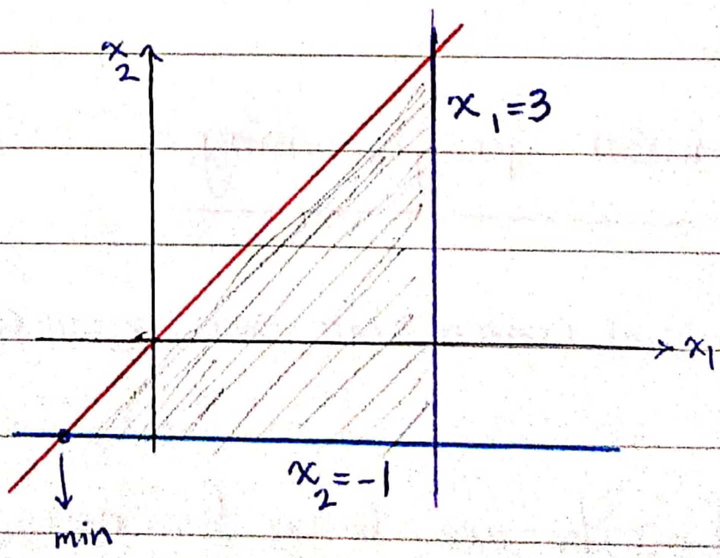
ex: $\min \; 2x_1 + x_2$

s.t $\quad x_1 \leqslant 3 \quad , x_2 \geqslant -1$

$\Downarrow$

$x_2 - x_1 - 1 \leqslant 0$

(a linear program)



$-x_2 - 1 \leqslant 0$ all inequalities $\xrightarrow[\text{convert to}]{}$ "less than" inequalities

Convex optimization problem :

$$\min_{x} f(x) \quad \text{s.t} \quad g_i(x) \leq 0, \quad h_j(x) = 0$$

→ این مسئله وقتی convex opt است که:

① * f, g_i are convex functions

یعنی علاوه بر خود تابع f که باید convex باشد،

تمام constraint ها (g_i) ی تر باید convex باشند.

② * $h_j(x) = 0$ is convex set

⇓

اهمیت این است که اگر مسئله‌ای اینگونه تعریف شود به راحتی قابل حل است.

در ML هم مثلاً در SVM این مسئله کاربرد دارد.

↳ با روشی جلوی over fitting گرفته می‌شود و بعد با روشی به convex opt problem تبدیل می‌شود.

Linear programming :

Linear programming یک مثال و زیرکلاس از convex opt prob است که در metabolic networks استفاده می‌شود.

\+ f, g, h are linear functions.

(Affine → عرض از مبدأ دارد
linear → ‶ ندارد)

↓ از نظر تکنیکی Affine هستند نه linear.

$$\boxed{\begin{aligned} &\min\; c^T x \\ &\text{s.t}\;\; Ax \leq b \end{aligned}}$$

→ همه‌ی constraint ها در قالب یک وکتور

/ /

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

$b \in \mathbb{R}^{m}$ → d → variables

$$\min \begin{bmatrix} 2 \\ 1 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

برای مثال قبل ← $m=3$ و $d=2$ ←

$$\text{s.t} \quad x_3 \leqslant 3, \quad x_2 \geqslant -1 \implies \text{s.t} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & -1 \\ -1 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{x} \leqslant \underbrace{\begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix}}_{b}$$

Quadratic programming

تعمیمی از linear programming است.

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

$$\text{s.t} \quad Ax \leqslant b$$

Q : PD matrix

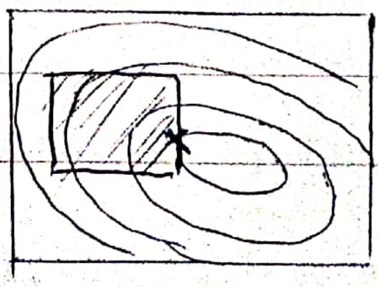
objective : Quadratic

linear constraint (Affine)

↓

$$\min_x \frac{1}{2} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 5 \\ 3 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\text{s.t} \quad \underbrace{\begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leqslant \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}}_{\text{box constrainted}}$$



/ /

* در linear programming، مستشعرجوابها در corner است.
* در 〃 quadratic جوابها می‌تواند روی یالها قرار گیرد.

Converting constrained problem to unconstrained problem:

$$\min_x f(x) \quad \text{s.t } g_i(x) \leq 0 \quad i=1,2,\dots,m$$

↓

simple idea: $\boxed{J(x) = f(x) + \sum_{i=1}^{m} \gamma(g_i(x))}$

به‌جای اینکه $f(x)$ را opt کنیم، $J(x)$ را opt کنیم.

$$\gamma(z) = \begin{cases} 0 & z \leq 0 \\ \infty & z > 0 \end{cases} \qquad J(x) = \begin{cases} f(x) & \text{all constraints satisfied} \\ \infty & \text{ow} \end{cases}$$

$\Downarrow$

not practical → but $\boxed{\min_x J(x)}$ → مبنایی برای روش lagrangian است.

Lagrangian:

idea: replace γ with a linear function.

$$L(x,\alpha) = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x)$$

$\alpha_i \geq 0$ are called lagrange multipliers. (also called dual variable)

$$L(x,\alpha) = f(x) + \alpha^T g(x)$$ یک وکتور فرم هم داریم :

$$J(x) = \max_{\alpha \geq 0} L(x,\alpha) = \begin{cases} f(x) & \text{all satisfied} \\ \infty & \text{ow} \end{cases}$$

Lagrangian duality :

optimization problem in x $\xrightarrow[\text{تبدیل کنیم به}]{\text{می‌خواهیم}}$ optimization problem in α

x : primal variables α : dual variables

primal problem : $\underset{x}{\text{minimize}}\ L_p(x)$

$$L_p(x) = \max L(x,\alpha)$$

$\Downarrow$

$$\boxed{\min_{x} \max_{\alpha \geq 0} L(x,\alpha)} \rightarrow$$ دوتا opt داریم

dual problem : $\max L_D(\alpha)$

$$L_D(\alpha) = \min_{x} L(x,\alpha) \rightarrow$$ lagrange dual function

concave تابع یک چیزی است $\leftarrow$ $\boxed{\max_{\alpha \geq 0} \min_{x} L(x,\alpha)}$

مشتق می‌گیریم

primal & dual problem دوتا جواب یکسانی نمی‌دهند و تحت شرایطی ممکن است جوابها یکسان باشد.

درحالت dual اگر minimization به راحتی انجام شود، max تر راحت پیدا می‌شود و حالت dual نسبت به primal راحت‌تر است.

(حالت primal جهت دوم (max) سخت است)

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

باید بفهمیم در چه حالت design و $p^*=d^* \leftarrow$ می‌شود، آن موقع به جای حل p^*، d^* را حل می‌کنیم.

$\boxed{p^*=d^*=L(x^*,\alpha^*)}$ → strong duality

+f g are convex
+other criteria } → strong duality

چرا مسئله‌ی dual خیلی مهم است:

+ if $\min_{x} L(x,\alpha)$ is easy ⇒ overal problem
↓
easy
↓ Cause
$L_D(\alpha)$ is a concave function

+easier in some problems
+ #constraints < # features → min → max

گاهی primal هم easy است، اما وقتی اتفاق تعداد constraint ها کمتر از feature ها باشد، dual راحت‌تر است

/ /

ML-12

Review → $\min_{x} f(x)$

$$\text{s.t. } g_i(x) \leq 0 \quad i=1,2,\dots,m$$

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

Lagrangian duality

primal problem : $\min_{x} L_P(x)$

$$\text{s.t. } L_P(x) = \max_{\alpha \geq 0} L(x,\alpha)$$

dual problem : $\max_{\alpha \geq 0} L_D(\alpha)$

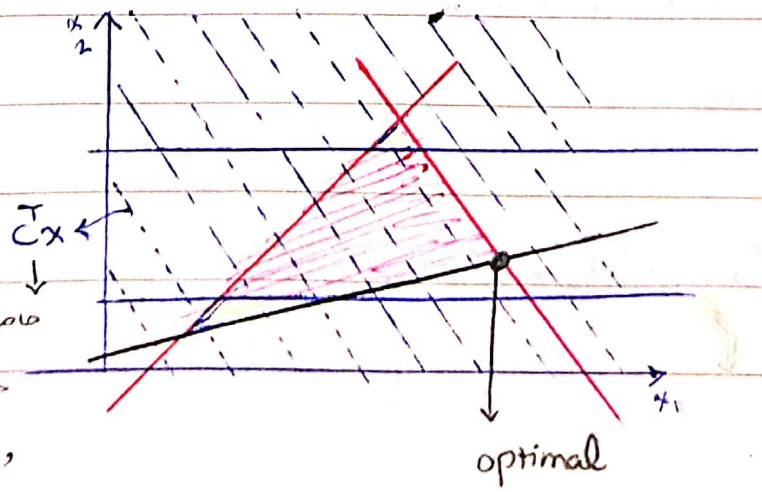
$$\text{s.t. } L_D(\alpha) = \min_{x} L(x,\alpha)$$

strong duality → $\boxed{p^* = d^* = L(x^*, \alpha^*)}$

Linear programming:

$$\min c^T x$$

$$\text{s.t. } Ax \leq b$$



حاصلضرب این دو که یک عدد است در کل این خط‌ها مقدار یکسانی است.

و هرچه c^Tx به سمت راست می‌رود این مقدار کمتر و کمتر می‌شود.

/ /

مربوط به شکل قبل :

$$\min \; -\begin{bmatrix}5\\3\end{bmatrix}^T \begin{bmatrix}x_1\\x_2\end{bmatrix}$$

subject to
$$\begin{bmatrix}2 & 2\\2 & -4\\-2 & 1\\0 & -1\\0 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} \leqslant \begin{bmatrix}33\\8\\5\\-1\\8\end{bmatrix}$$

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

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

The dual opt problem: ① $\frac{\partial L}{\partial x} = 0 \qquad \min_x L(x,\alpha) \rightarrow c + A^T\alpha = 0$

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

dual problem :
$$\boxed{\begin{aligned}&\max = -b^T\alpha\\ &\text{s.t} \begin{cases}\alpha \geqslant 0\\ c + A^T\alpha = 0\end{cases}\end{aligned}}$$

$\rightarrow$ در خیلی از مسائل این constraint را نداریم ولی مشکل quadratic داریم.

m constrained $\rightarrow$ dual

d feature $\rightarrow$ primal

HW : فرم dual را برای quadratic به دست آورید.

[مشتق می‌گیریم برابر صفر قرار می‌دهیم وقتی مشتق می‌گیریم یک x هم باقی می‌ماند که آن را در lagrangian می‌گذاریم، فرم $L_D(\alpha)$ می‌دهد]

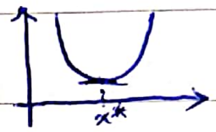
KKT Conditions (Karush-Kuhn-Tucker)

differentiable

In unconstrained optimization : x^* : optimal point $f(x)$

$\Downarrow$

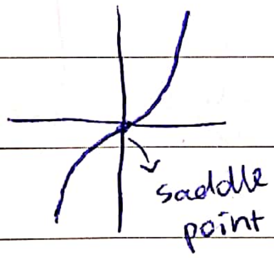
$\nabla f(x^*) = 0$



KKT conditions are similar conditions for constrained opt.

نقطه‌ای از تابع

می‌دانیم که در نقطه‌ی optimal مشتق اول صفر است (اگر مشتق‌پذیر باشد) ولی اگر مشتق اول صفر باشد، آیا لزوماً می‌توانیم بگوییم آن نقطه min یا max است؟ خیر.
تحت شرایطی می‌توانیم این موضوع را تأیید کنیم، اما همیشه این امر صادق نیست.



مثال : $f(x) = x^3$

$f'(x) = 3x^2 = 0 \rightarrow x = 0$

$f''(x) = 6x = 0 \rightarrow x = 0$

هدف KKT conditions، بیان گزاره‌ای است در مورد x^* و α^*.

If x^*, α^* are primal & dual solutions

$\Downarrow$

x^* & α^* satisfy KKT conditions

/ /

KKT Conditions :

1. **primal constraints** $\rightarrow g_i(x^*) \leqslant 0 \quad , \quad i=1,\dots,m$

2. **Dual constraints** $\rightarrow \alpha_i^* \geqslant 0 \quad i=1,\dots,m$

3. **complentary slackness** $\alpha_i^* g_i(x^*) = 0 \rightarrow$ چه وقتی strong duality داریم این شرط برقرار است؟

* $\boxed{\text{optional HW}}$ ↑proof?

4. **stationary** $\nabla F(x^*) + \sum_{i=1}^{m} \alpha_i^* \nabla g_i(x^*) = 0$

در صورتی که هیچ constraint نداشته باشیم ← $\nabla F(x^*) = 0$

If x^* and α^* are primal & dual solutions $\rightarrow$ $\boxed{x^* \;\&\; \alpha^* \text{ satisfy KKT conditions}}$

$\left\{\begin{array}{l}\text{differentiable } f, g(i) \\ \text{strong duality}\end{array}\right]$ necessary condition

↓

وقتی که جواب پرایم و dual یکی باشند. ← و این را by design می‌توانیم حل کنیم.

↓

یک سری قضایای دیگر به نام qualification constraints داریم که فرم خیلی راحتی دارند. مثلاً قضیه‌ی slater's condition.

و طبق این قضایا می‌توان با داشتن f و g مشخص کرد که strong duality داریم یا نه.

مثلاً در linear programming قبل از اینکه حل کنیم می‌دانیم که strong duality داریم. طبق slater's condition اگر f, g convex باشند، feasible space تهی نباشد strong duality داریم.

مثلاً در SVM، یک مسئله‌ی quadratic است و تابع f و g را چون خودمان طراحی می‌کنیم، convex توصیف می‌کنیم و می‌دانیم که strong duality وجود دارد و چون strong duality وجود دارد، x^* و α^* رابطه دارند پس جوابی که برای x^* و α^* به دست می‌آوریم حتماً در KKT conditions صدق می‌کند.

---

sufficiency of KKT conditions :

$\tilde{x}$ and $\tilde{\alpha}$ satisfy KKT conditions $\Rightarrow$ $\boxed{\tilde{x}, \tilde{\alpha} \text{ are primal \& dual solutions (zero duality)}}$

+ f and g_i are convex

duality gap صفر است یعنی strong duality داریم.

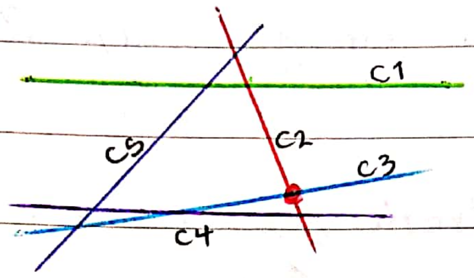
practical implications → + convergence check

+ $\alpha_i = 0$ for some constraints.

همانطور که در complementary slackness مشخص است در بعضی جاها، $\alpha_i = 0$ است. اگر به شکلی که در ابتدای جلسه مطرح شد دقت کنیم، 5 constraint داشتیم که دوتای آنها binding

شیب بهینه در نقطه‌ی optimal دخیل هستند و اگر جابجا شوند تأثیری در آن ندارند پس $\alpha_i = 0$

اما constraint هایی که در تعیین نقطه‌ی optimal دخیل اند و جابجایی آنها موثر است $\alpha \neq 0$ هست.



lagrange ضریب

for $c_{1,4,5} \rightarrow \alpha_i = 0$

for $c_{2,3} \rightarrow \alpha_i \neq 0$

$g_5(x^*) < 0$

coordinate-wise optimization :

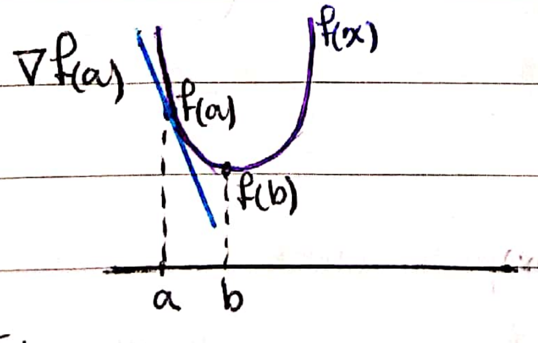
بسیاری از روش‌های جدید براساس این روش learning و optimization

parameter estimation را انجام می‌دهند و روش exted شده‌ای از این روش در SVM

استفاده می‌شود.

این روش به subgradient وابسته است. (مثلاً برای حل LASSO)

subgradient :



فرض کنید تابعی به این صورت داریم:

بسط تیلور

$f(b) \geq \boxed{f(a) + \nabla f(a)^T (b-a)}$ ← تعمیم‌ای داریم که برای توابع convex صدق می‌کند

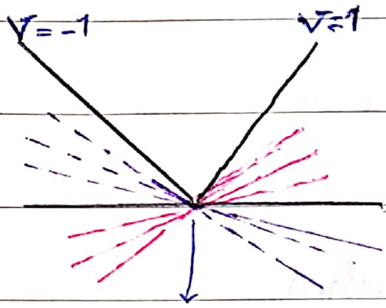
(linear approximation always underestimates f) subgradient تعمیمی است از گرادیان به نقاطی که مشتق پذیر نیستند

* subgradient is generalization of gradient to non-differentiable points.

- subgradient of convex f is any V.

$$f(b) \geq f(a) + V(b-a) \quad \text{for all } b$$

به جای گرادیان



$V \in [-1,1]$

باید زیر دو خط x, $-x$ باشد

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

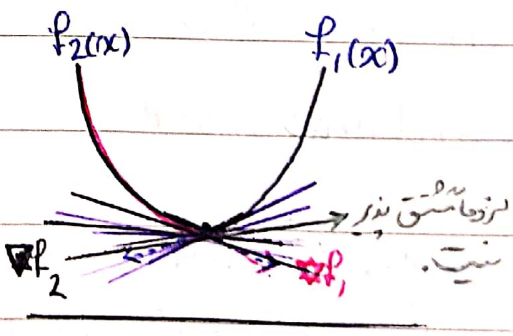
$f(x) = |x|$

اینجا کار نمی‌کند.

ex: let f_1 & f_2 be convex and differentiable.

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

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



در نقاط مشتق پذیر نیست.

Coordinate-descent algorithm : (coordinate-wise opt)

+ simple, efficient and scalable.

این روش نسبت به روش های gradient descent خیلی سریعتر است و همچنین می توان تعداد زیادی پارامتر را با آن opt کرد.

$$+ \min_{\beta} \overset{\text{loss function}}{L}(\beta_1, \dots, \beta_d) \rightarrow$$ یک مسئله ی unconstrained

Algorithm :

iterate until convergence {

for $i = 1, \dots, d$

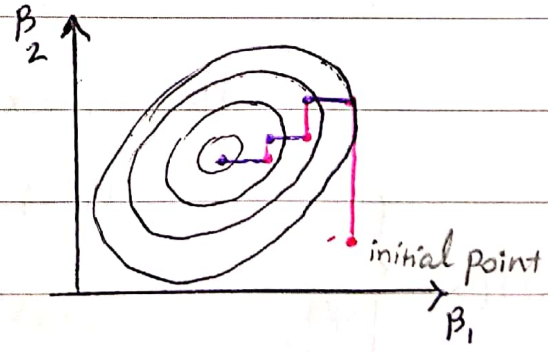
$$\beta_i = \arg\min_{b} L(\beta_1, \dots, \beta_{i-1}, b, \beta_{i+1}, \dots, \beta_d)$$

}

هر بار یک متغیر را optimize می کنیم و بقیه را ثابت می گیریم و این کار را تا متغیر آخر ادامه می دهیم.

$\Downarrow$

coordinate-wise



+ keep all other variables fixed.
+ one-at-a time not all-at-once
+ coordinate-wise min $\Rightarrow$ global min

$\Downarrow$

البته این الگوریتم همیشه global min نمی دهد، اگر تابع convex و differentiable باشد این اتفاق می افتد.

LASSO → مشتق پذیر است اما یک قسمت غیرهموار جداپذیر داردو
L1 norm دارد.

+separable non-smooth

$$\underbrace{\underbrace{(Y-X\beta)^T(Y-X\beta)}_{\text{هموار}} + \lambda \underbrace{\|\beta\|_1}_{\text{ناهموار}}}_{\text{separable}}$$