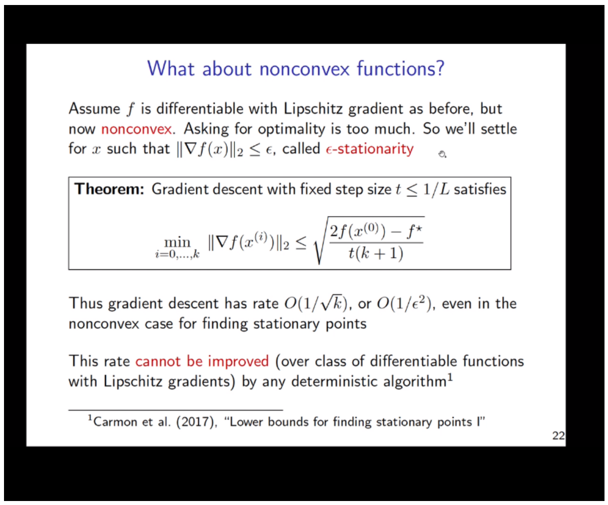
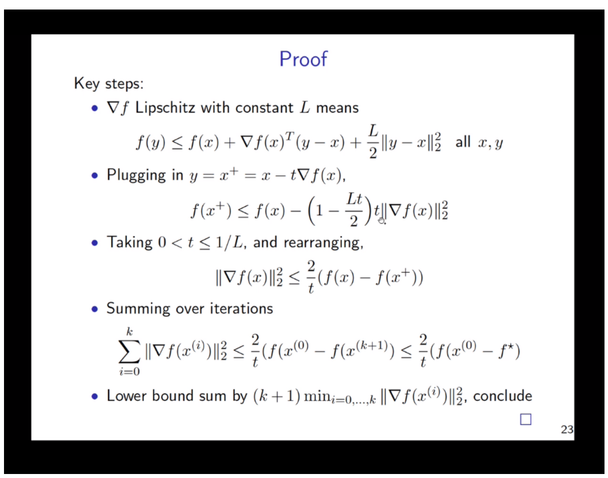
Canonical result in gradient descent convergence analysis

- if you have a function that has a Lipschitz continuous gradient
- take a step size ⩽ 1/lipschitz constant
- then after k iterations, the criterion value is improved by a factor of k

What if we have non-convex f(x)?





In the non convex case, we can't ask for optimality - so we settle for x st $||\nabla f(x)||_2 \ \leqslant \ \epsilon$, called $\epsilon - stationarity$.

Theorem: $min_{i=0, \dots, k}||\nabla f\left(x^{(i)}\right)||_2 \ \leqslant \ \sqrt{\frac{2(f\left(x^{(0)} - \ f^*\right)}{t(k+1)}}$

- the minimum L2 norm across all the gradients you observe across k iterates of the gradient descent is upper bounded by the original criterion value weighted by a factor of 1/sqrt(k+1)

- So gradient descent has a rate $O\left(1/\sqrt{k}\right)$

For $\epsilon - stationarity$, you want $||\nabla f(x)||_2 \leqslant \epsilon$

- So the number of steps k it will take to guarantee this result will be
- $\epsilon = \sqrt{\frac{2(f\left(x^{(0)} - f^*\right)}{t(k+1)}} \implies \epsilon^2 = c/k \implies k = c/\epsilon^2$
- so gradient descent has a rate

For a small ϵ this can be quite large.

Also, this does not guarantee that you end up a local minima! Could be a saddle point or a max - any $\epsilon - stationary$ point

- however recent research suggests you are more likely to end up in minima than other stationary points

Anatomy of a convergence rate proof

1. Write down the function and all assumptions being made

2. Start with some quadratic upper or lower bound on f(y) around f(x) (where x is the current iterate)

eg: if gradient is Lipschitz with constant L: upper bound on f(y)

strong convexity: lower bound on f(y)

3. Establish some 'sufficient descent' property between f(x+) (the next iterate) and the current iterate

With strong convexity, typically this bound is on the iterates themselves i.e. it takes the form $||x^+ - x^*||_2^2 \leqslant ...$

Without strong convexity, typically get a bound in terms of the criterion function evaluations, i.e. something of the form $f\left(x^+\right) \leqslant f(x) - \frac{t}{2}||\nabla f(x)||_2^2$

4. Iterate/recurse the property to get a global statement about $f\left(x^{(k)}\right)$ or $x^{(k)}$

Gradient Boosting

Idea:

- do GD on a loss fn - eg classificaiotn or regression - supervised learning task
  - smooth loss funciton, as in GD
- The trick is you replace gradient with the closest approximation you can make using a tree
  - map gradients to a vector of predictions given by a tree (or any other simple

method) - "weak learner"

- constrained problem: force the gradients to be part of the constraint set.

Example: find the best linear combo of trees to minimize classification errro

- but this is too hard to parameterize
  - for example, the space of all d=5 trees is enormous
- So ignore that you want trees
- compute the gradient - if you could fit anything to the data, what is the fastest descent you can get on the criterion
- then find a tree that comes close to the gradient in its predictions, and add that to our collection
- repeat until you get some collection of trees
- https://en.wikipedia.org/wiki/Gradient_boosting#:~:text=Gradient%20boosting%20is%20a%20machine

------

A. Subgradients

A1. Motivation:

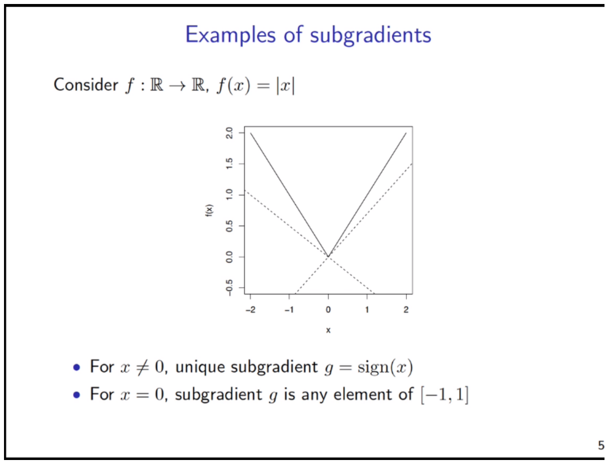
Recall that for a convex function $f(y) \geqslant f(x) + \nabla f(x)^T(y-x)$, that is, the linear approximation always underestimates f

A subgradient of f at point x is g such that the same property holds ie
$f(y) \geqslant f(x) + g^T(y-x)$ for all y

- for a convex function, a subgradient always exists on the relative interior of dom(f)
  - relative interior = if the domain is not full dim, then it is the interior wrt some subspace where the domain lies
    - these are points far away from where the function is infinte
  - Proof: comes from supporting hyperplane theorem
- if a gradient exists, then it is the only subgradient
- the same definition is true for non convex f, however g may not exist

Example 1: Absolute value

- $f: R \rightarrow R,\ f(x) = |x|$
- this is a convex function
- not differentiable everywhere
  - at x = 0 it is not
- for x neq 0, unique subgradient g = gradient = sign(x)
- for x = 0, subgradient is any element of [-1, 1]
  - you will underestimate f(0) with a line of any slope between -1 and 1!
-



Example 2: L2 norm

for x neq 0, unique subgradient $g = x/||x||_2$

for x = 0, subgradient g is any element of $\{z: ||z||_2 \leqslant 1\}$

- unit ball!

Working it out, we need some g st

$f(y) \geqslant f(x) + g(y-x)$

at x = 0

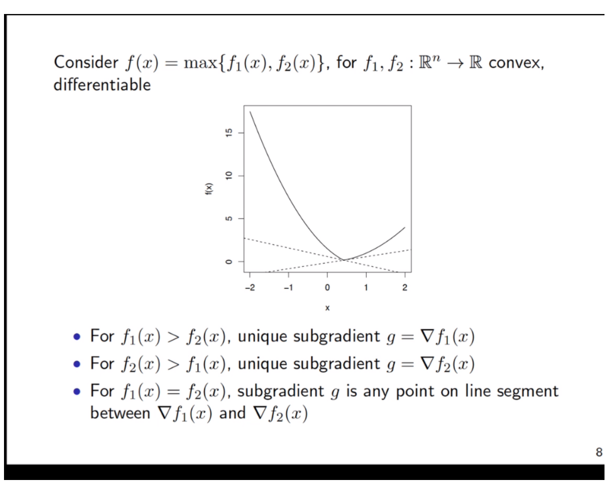
$||y||_2 \geqslant g^T y$

From cauchy-schwartz (?) this is true for any $g: ||g||_2 \leqslant 1$

Example 3: L1 norm

- not differentiable on any of the coordinate axes
- when $x_i \neq 0$, g = gradient = $sign(x_i)$
- when $x_i = 0$, subgradient is any element of [-1, 1]

Example 4: max of convex functions



B. Subdifferentials

- set of all subgradients of convex f is the subdifferential
- $\partial f(x) = \{g \in R^n : g \text{ is a subgradient of } f \text{ at } x\}$

B1. Properties

- nonempty for convex f
- $\partial f(x)$ is closed and convex, even for non convex f
- to prove that a convex function is smooth
  - characterize the subdifferential
  - if the set has one element then it must be the gradient and the fn is differentiable

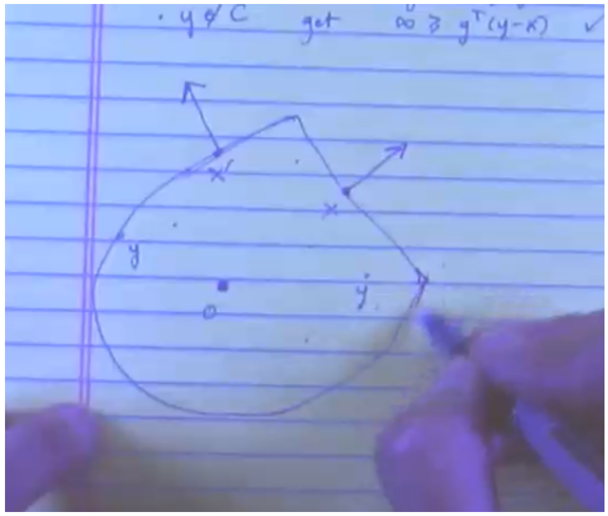
B2. Connection to convex geometry

Remember the indicator function $I_C : R^n \to R$
$I_C(x) = I\{x \in C\} = \{0 \; if \; x \in C, \; inf \; else\}$

For $x \in C$, $\partial I_c(x) = N_c(x)$, the normal cone of C at x
Recall that $N_C(x) = \{g \in R^n : g^T x \geqslant g^T y \; for \; any \; y \in C\}$

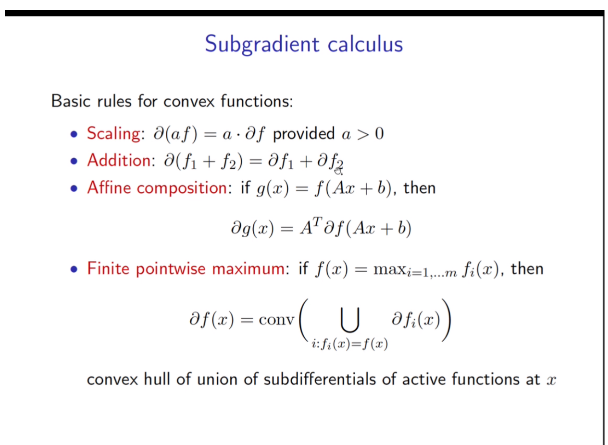
Why?
By definition of subgradient g,
$I_C(y) \geqslant I_C(x) + g^T(y-x) \; for \; all \; y$
But we know:
if $y \notin C$, $I_C(y) = inf$
if $y \in C$, $I_C(y) = 0$
so then $0 \geqslant g^T(y-x) \implies g^T x \geqslant g^T y$, which is the definition of hte normal cone for x

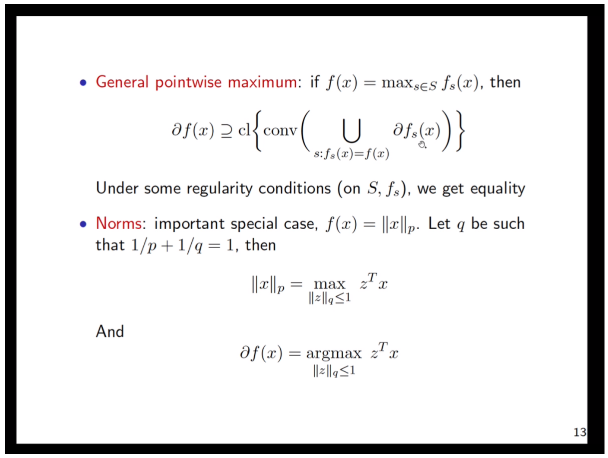
Intuition for normal cones:



For a given point x on the boundary, the normal cone will consist of multiples of the vector that is aligned in its direction
In the interior, the normal cone is {0} - can always find a point onthe boundary more aligned with the vector you have chosen

B3: Subgradient calculus
Basic rules for convex functions





The dual representation of the L_p norm:

$$||x||_p = \left(\sum |x_i|^p\right)^{1/p}$$

for $p \geqslant 1$, $\exists\, q\ st\ 1/q + 1/p = 1$

Fact: $||x||_p = max_{y:\, ||y||_q \leqslant 1}\, y^T x$

Working through this myself:

For p = 2 ⟹ q = 2

Then $||x||_2 = \sqrt{x^T x} = y^T x \rightarrow solve\ for\ y?$

Why subgradients?

1. convex analysis: relationship to duality
2. convex opt: you can minimize any convex function if you can compute a subgradient

B4. Subgradient Optimality Condition

For any f convex or not, $f\left(x^*\right) = min\ f(x) \implies 0 \in \partial f\left(x^*\right)$

i.e. x^* is a minimizer iff 0 is a subgradient of f at x^*.

Why? if $g = 0$ is a subgradient, then for all y

$f(y) \geqslant f\left(x^*\right) + 0^T\left(y - x^*\right) = f\left(x^*\right)$

B5. Derivation of first-order optimality

Example of the power of subgradients

Recall that

$min_x\ f(x)\ subject\ to\ x \in C$ is solved at x, for f convex and differentiable, iff

$\nabla f(x)^T (y - x) \geqslant 0\ for\ all\ y \in C$

Intuitively, says the gradient increases as we move away from x.
How to prove this?
1. Recast the problem as $min_x \ f(x) \ + \ I_C(x)$
2. Now apply subg optimality: $0 \ \in \partial(f(x) \ + \ I_C(x))$ at the solutions to this problem
3.
$$
\begin{aligned}
& 0 \ \in \partial(f(x) \ + \ I_C(x)) \\
& \implies 0 \ \in \{\{\nabla f(x)\} \ + \ N_C(x)\} \ (from\ subg\ calculus\ -\ additive\ rule) \\
& \implies which\ means\ -\nabla f(x) \ \in N_C(x) \\
& by\ definition\ of\ normal\ cone,\ this\ means \\
& -\nabla f(x)^T x \ \geqslant \ -\nabla f(x)^T y\ for\ all\ y \ \in C \\
& \implies \nabla f(x)^T (y - x) \ \geqslant \ 0\ for\ all\ y \ \in C
\end{aligned}
$$

Note that the condition $0 \ \in \partial f(x) \ + \ N_C(x)$ is a fully general condition for optimality in convex problems

- you can express any convex problem as being some objective function + indicator function on x being in some convex set C
  - incorporate all the constraints in the construction of the set C

However, this condition is not always easy to work with - KKT conditions helpful here

Example: lasso optimality conditions

The lasso problem is $min_\beta \ \frac{1}{2}||y - X\beta||_2^2 \ + \ \lambda||\beta||_1$
for some $\lambda \ \geqslant \ 0$.
Subgradient optimality:
$0 \ \in \partial\left(\frac{1}{2}||y - X\beta||_2^2 \ + \ \lambda||\beta||_1\right)$ at x^*
The first term is a convex, diff function with gradient $-X^T y \ + \ X^T XB$
The second term is a convex non diff function, but we know its subgradient form
So

$$
\begin{aligned}
& \implies 0 \ \in \ -X^T(y \ - \ X\beta) \ + \ \lambda\partial\beta \\
& \implies X^T(y - X\beta) \ = \ \lambda\partial\beta \\
& \implies X^T(y - X\beta) \ = \ \lambda v
\end{aligned}
$$

$for\ some\ v \ \in \partial\beta$
which means

$$
\begin{aligned}
& v_i \ = \ \{ \\
& \ \{1\}\ if\ \beta_i > 0, \\
& \ \{-1\}\ if\ \beta_i \ < \ 0, \\
& \ [-1,\ 1]\ if\ \beta_i \ = \ 0 \\
& \}
\end{aligned}
$$

Write the columns of X out as $X_1,\ X_2,\ ...,\ X_p$. then the condition reads

$X_i^T(y - X\beta)\ =\ \lambda.sign(\beta_i)\ if\ \beta_i\ \neq 0,$

$|X_i^T(y - X\beta)|\ \leqslant \lambda\ if\ \beta_i = 0$

This doesn't tell give you a closed form solution to the lasso problem. But it does give you conditions for lasso optimality!

You can check a priori if a given column will have a corresponding beta value or not by just checking if $|X_i^T(y - X\beta)|\ \leqslant \lambda$ or not!

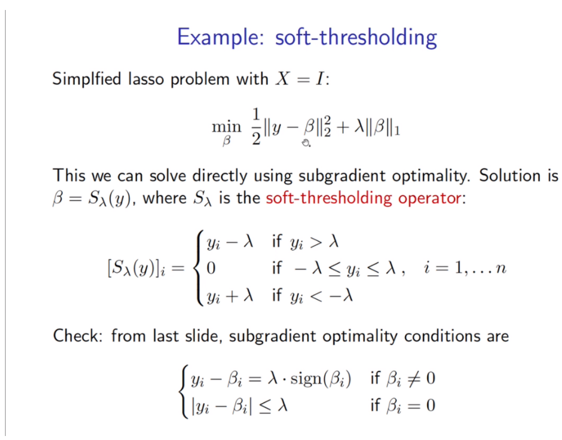
- This can be used to screen variables even before solve - i.e. drop some variables from the model
- Also, we can check solutions using these optimality conditions

One intuitive interpretation of these optimality conditions is:

If X_i is used in the regression, then its correlation with the residual will be maximal ($\lambda\ or\ -\lambda$)

If it is not, then its correlation with the residual will be less

Example 2: soft-thresholding



$\beta_i = S_\lambda(y_i)$

check: $y_i - \beta_i = \lambda \, \text{sign}(\beta_i)$ if $\beta_i \neq 0$

$|y_i - \beta_i| \le \lambda$ if $\beta_i = 0$

- $y_i > \lambda$. $\beta_i = y_i - \lambda > 0$

  $y_i - \beta_i = \lambda = \lambda \cdot \text{sign}(\beta_i)$ ✓

- $y_i < -\lambda$ similar

- $y_i \in [-\lambda, \lambda]$ $\beta_i = 0$

  $|y_i| \le \lambda$ ✓