Lecture #12

→ We've been thru the basics, culminating in a derivation of KKT. | mentioned briefly another deriv based on $\min f(x) + \max_i f_i(x)$

~~Example~~: ~~min~~ $\sum_i \log(1+e^{a_i^T x})$ ~~s.t.~~ $x^T x = 1$

NNLS: $\min \frac{1}{2}\|Ax-b\|^2$ s.t. $x \geq 0$

$\phi(x) = \max\{f(x) - f^*, h_1(x), \ldots, h_m(x)\}$

$$L(x,\lambda) = \frac{1}{2}\|Ax-b\|^2 - \lambda^T x$$

Finding direct solution not easy!

$$\begin{cases} \frac{\partial L}{\partial x} : & A^T(Ax-b) - \lambda = 0 \quad -(1) \\ & \lambda_i x_i = 0 \quad \text{(nonlinear!)} \\ & x \geq 0, \ \lambda \geq 0 \end{cases}$$

→ Usually, KKT ~~solution~~ not easy to solve directly, iterative approaches are taken

→ Even most simple: consider a func $f(x)$:

$$f(x) = \frac{1}{2}x^T A x - b^T x, \quad A \succeq 0$$

- $\nabla f(x) = Ax - b = 0 \Rightarrow$ solve linear syst $Ax = b$.
- If $b \in \text{Ran}(A)$, syst has a solution

Direct methods: Chol. decomp of A: $\frac{1}{3}n^3$ flops

Iterative methods: (Vast area within numerical linear algebra since well over 40 years!)

Many key Opt. algorithmic ideas originally stems from trying to solve $Ax = b$!

→ At an abstract level:

$\min f(x)$

(1) $\Rightarrow \nabla f(x) = 0 \Rightarrow x = (\nabla f)^{-1}(0) = \nabla f^*(0)$

(2) $\nabla f(x) = 0 \Rightarrow -\alpha \nabla f(x) = 0 \quad \forall \alpha > 0$ too

$\Rightarrow x - \alpha \nabla f(x) = x$ must hold

Suggests an algorithm! $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$!

1.5

$$-\alpha \nabla f(x) = 0$$

$$\Rightarrow -\alpha S \nabla f(x) = 0$$ this holds if S is invertible.

we can try $x - \alpha (\nabla^2 f(x))^{-1} \nabla f(x)$: Newton method

Other "creative" possibilities sometimes

Eg $f(x) - g(x)$

$\nabla f(x) - \nabla g(x) = 0$

$\nabla f(x) = \nabla g(x)$

Numerous possibilities to play with these

(Soon we'll see a practical choice)

→ this is the "algebraic" derivation of Subdifferential ⑤

→ Say $f(x)$ is non-diff. Then,

$$0 \in \partial f(x)$$

Now, actually $(\partial f)^{-1}(0)$ is a multivalued map: a set.

How about

$$x \in x - \alpha \partial f(x)$$

as before? Less clear. because this generates (!)

$$x_{k+1} \in x_k - \alpha_k \partial f(x_k)$$

(With careful choice of α_k, using any $g_k \in \partial f(x_k)$ works!)

→ $0 \in \partial f(x) \Rightarrow x \in x + \alpha \partial f(x)$

$\Rightarrow (Id + \alpha \partial f)(x) \ni x$

$\Rightarrow x = (Id + \alpha \partial f)^{-1}(x)$

$\Rightarrow x_{k+1} = (Id + \alpha \partial f)^{-1}(x_k)$

(Fixed-point iteration).

This is actually Rockafellar's famous "proximal point algorithm" → a method that ultimately includes all sorts of popular iterations (by a proper choice of ∂f)

→ Q: A-priori no reason for above "guess" to work → but they are a good starting point.

Q: Say we have $\phi(x) = f(x) + g(x)$

(a) $x_{k+1} = x_k - \alpha_k \nabla \phi(x_k)$

Seems to be ok if $\nabla \phi(x_k) = \nabla f(x_k) + \nabla g(x_k)$

(b) Suppose $\phi(x) = f(x) - g(x)$ where both f, g, cvx.

How should we go about minimizing such a ϕ?

Key idea in OPT:
- Start with original iterate and a guess, say x_k:
- Build an approx. to $\phi(x)$, centered at x_k, with the aim that approx is easy to optimize; call it $m(x; x_k)$
- Obtain $x_{k+1} = \arg\min_x m(x; x_k)$

- Gradient descent: $\phi(x+\delta) \approx \underbrace{\phi(x) + \langle \nabla \phi(x), s \rangle + \frac{1}{2\alpha_k}\|s\|^2}_{m(x,s)}$

$$x_{k+1} = \arg\min_{\delta} x_k + \delta.$$

$$s = \arg\min_s \phi(x) + \langle \nabla \phi(x), s \rangle + \frac{1}{2\alpha}\|s\|^2$$

$$= \nabla \phi(x) + \frac{1}{\alpha} s = 0$$

$$\Rightarrow s = -\alpha \nabla \phi(x)$$

So $\boxed{x_{k+1} = x_k - \alpha_k \nabla \phi(x_k)}$ ∴ G.D.

- $\phi(x) = f(x) - g(x)$; f, g, cvx

$g(x+\delta) \geq g(x) + \langle \nabla g(x), s \rangle$

So $\phi(x+\delta) = f(x+\delta) - g(x+\delta)$

$$\leq \underbrace{f(x+\delta) - g(x) - \langle \nabla g(x), s \rangle}_{m(x+\delta)}$$

$x_{k+1} = x_k + \delta$

$s = \arg\min_s f(x+\delta) - \langle \nabla g(x), s \rangle$

3.5

Example of CCCP

Suppose we wish to minimize

$$\phi(X) = \min_{X>0} \sum_i \log \frac{\det\left(\frac{X+a_i}{2}\right)}{\sqrt{\det(X)\det(a_i)}}$$

This problem is called Fréchet-mean:

$$\phi(X) = \sum_i \left[\ld\left(\frac{X+a_i}{2}\right) - \frac{1}{2}\ld(X a_i)\right]$$

$$= \sum_i \left[-\frac{1}{2}\ld(X a_i) - \left(-\ld\left(\frac{X+a_i}{2}\right)\right)\right]$$

$$f(X) - g(X)$$

Obtain linearization:

$$m(X_k, X) = \min_{S} \sum_i -\frac{1}{2}\ld\left(\left(\frac{X_k+S}{2}\right)a_i\right) - \frac{1}{2}\left\langle \left(\frac{X_k+a_i}{2}\right)^{-1}, S\right\rangle$$

min it we get $\sum_i \left(-\frac{1}{2}(X_k+S)^{-1} - \frac{1}{2}\left(\frac{X_k+a_i}{2}\right)^{-1}\right) = 0$

$$\Rightarrow n(X_k+S)^{-1} = \sum_i \left(\frac{X_k+a_i}{2}\right)^{-1}$$

$$\Rightarrow (X_k+S)^{-1} = \frac{1}{n}\sum_i \left(\frac{X_k+a_i}{2}\right)^{-1}$$

$$\Rightarrow X_{k+1} = X_k + S = \left[\quad\right]^{-1}$$

\# Does this work? Converge? Etc.?

Actually this particular example does have global optimum!

~~# Beyond CC~~

~~# However: Not always~~ do.

The art of using CCCP lies in decomposing a given nonconvex function into nice parts that allow easier optimization.

It ensures ↓ etc.

Reading assignment: "On the convergence of the Convex-Concave Procedure"
Sriperumbudur, Lanckriet

More generally, a fairly large heuristic area: D.C. Programming
exists → we might revisit it, depending on importance.

~~Here's another example~~ For several examples of use of CCCP in Stats & ML.

(a) "The Concave-Convex Procedure" — Yuille & Rangarajan

(b)

Note: This method is often a useful baseline to beat when dealing with nonconvex problems!

This repeated model-/bound minimization is called Convex-Concave Programming!

→ (One-member of a class of methods called Majorize-Minimize, ch.) (We'll see some more of this later in class) → But really handy tool to have in repertoire.

A few words about gradient-descent methods:

Say $\min f(x)$ s.t. $x \in \mathbb{R}^n$

- x s.t. $\nabla f(x) \in 0$;
- Look at $x_\alpha = x - \alpha \nabla f(x)$

Methods without explicit derivatives

$$\min_{x \in \mathbb{R}^n} f(x) = f(x_1, \ldots, x_n)$$

at each iteration optimize over just 1 coordinate (or more if feasible):

Called coordinate descent or Block-CD

If $f(x) = f(u, v)$, $u \in \mathbb{R}^{n_1}$, $v \in \mathbb{R}^{n_2}$

Then $u_{k+1} \leftarrow \arg\min_u f(u, v_k)$

$v_{k+1} \leftarrow$ " $f(u_{k+1}, v)$

Observe that $\{f(u_k, v_k)\}$ is $\downarrow$.

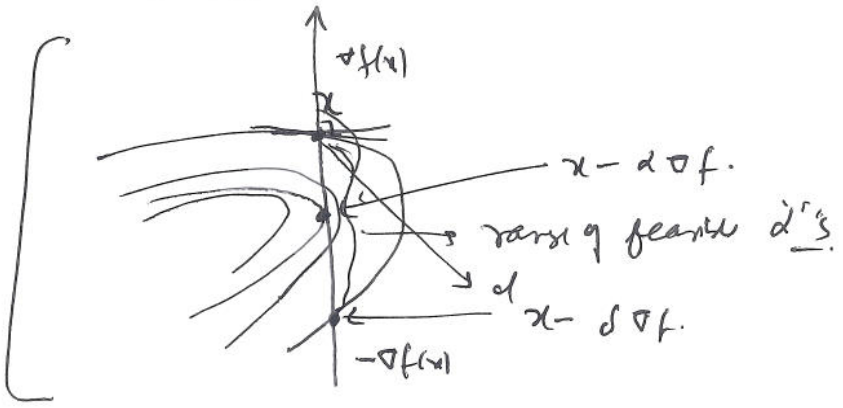
The two block case of alternating minimization is special → we will return to it later in the course.

Back to gradient-descent methods:

$x \leftarrow x - \alpha \nabla f$ — for small enough $\alpha > 0$ we have descent

more generally $x \leftarrow x + \alpha d$ ← descent direction

This picture is important.



Code: while ¬done

$x_{new} \leftarrow x_k + \alpha_k d_k$

Update α_k, d_k Somehow.

end.

α_k chosen to ensure $f(x_{new}) < f(x_k)$ (Unless already stationary)

d_k : descent direct. $\langle \nabla f(x_k), d_k \rangle < 0$

Read "Numerical Optimization" by Nocedal & Wright to learn much more about different strategies for selecting α and d

How about breaking the rules a little-bit?

DO NOT insist on $f(x_{new}) < f(x_k)$

allows more flexibility in choosing α_k's.

But why? maybe Empirical speedups!

Observe: $m(x, \delta) = f(x) + \langle \nabla f(x), \delta \rangle + \frac{1}{2\alpha} \|\delta\|^2$ was ① model

How about this model.

$m(x, 1) = f(x) + \langle \nabla f(x), 1 \rangle + \frac{1}{2\alpha} \|\delta\|^2$ ← α was fixed

But now optimizer over $\alpha \neq 0$

- $x_{k+1} = x_k - \alpha S_k \nabla f(x_k)$: Newton, Quasi-Newton
- ~~What if case~~ Crux is : How to select S_k efficiently... ...(LBFGS, etc.)
- Let us instead ~~choose~~ Set $\alpha = 1$; $S_k = \beta I$;

  Find a good β!

  $S_k \approx [\nabla^2 f(x_k)]^{-1}$;

  $H \cdot \Delta x = \Delta g$

  $H^{-1} \Delta g = \Delta x$

$$\min_{\beta} \| \beta \Delta g - \Delta x \|^2 : \quad \beta^2 \langle \Delta g, \Delta g \rangle - 2\beta \langle \Delta g, \Delta x \rangle$$

$$\Rightarrow \beta = \frac{\langle \Delta g, \Delta x \rangle}{\|\Delta g\|^2} .!$$

Alternatively we can search for

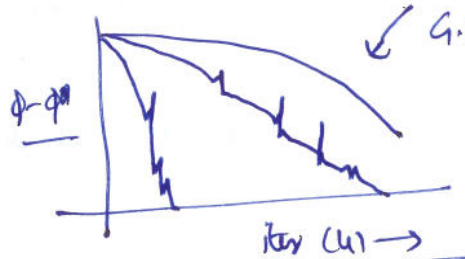
$$\| \beta \Delta x - \Delta g \|^2 \Rightarrow \beta = \frac{\langle \Delta x, \Delta g \rangle}{\|\Delta x\|^2} \Rightarrow \beta^{-1} = \frac{\|\Delta x\|^2}{\langle \Delta x, \Delta g \rangle} .!$$

Step is $\frac{1}{\beta}$

→ These steps are known as "Barzilai-Borwein" step sizes!

→ Empirically remarkable! "Barzilai & Borwein" : Two-point step size gradient methods. IMA J. Num. Anal; 1988.

Lead to picture such as :



Let's ~~start~~ begin now a more careful study of theoretically evaluating performance of methods. : Best always, for a real-problem, from compare on a computer to get a better idea! Theory is just a guideline.

7

$O(\frac{1}{\epsilon^2})$. ~~$\frac{1}{2^{2\epsilon}} = \epsilon^k$~~ $\log \epsilon = -2\epsilon \log 2$ $\log \epsilon = 2\epsilon \log(\epsilon_2)$

$\Rightarrow 2\epsilon = \frac{\log \epsilon}{\log(\epsilon_2)}$

To judge methods / their performance, we'll ~~could~~ could

~~Use~~ Use:

- ~~Oracle~~ analytical complexity:
- Arithmetic complexity:

What does it mean to solve? a problem:

Ans: obtain a point $\bar{x}$ s.t. $f(\bar{x}) \leq f^* + \epsilon$

When running depends on $\text{poly}\left(\frac{1}{\epsilon}\right)$ or $\text{poly}\left(\log\left(\frac{1}{\epsilon}\right)\right)$, $\log\log\left(\frac{1}{\epsilon}\right)$

Alternatively: $\|\nabla f(\bar{x})\| \leq \epsilon$ · maybe measured.

Or even · $\|x_{k+1} - (x_k - \alpha_k \nabla f(x_k))\|$. and so on..

Model of Computation:

Types of oracles and examples of these oracles.

Determ
- Zeroth
- First
- Second

Stoch

More generally:

Oracles with error.

that does not satisfy $\mathbb{E}[\eta] = 0$.

(Biased oracle)

Classes of functions:

$C_L^{k,p}(\mathcal{X})$: k times cont. diff func. whose p^{th} deriv is Lipschitz cont with const L i.e.

$$\|f^{(p)}(x) - f^{(p)}(y)\| \leq L\|x-y\|, \quad \forall x, y \in \mathcal{X}.$$

Useful Lemma (HW): $f \in C_L^{2,1}(\mathcal{X})$ iff $\|f''(x)\| \leq L$ $\forall x \in \mathcal{X}$.

Ques: What is l.c. of $f(x) = x^T A x$? ; $f(x) = \log(1+e^x)$

Note: $-\log x \notin C_L^{1,1}(\mathbb{R}_{++})$ for any L | $\max(0, 1-x) \in C_1^{0,\cdot}(\mathbb{R})$.

| use cond $C_L^1 \equiv C_L^{1,1}$

~~Let us finally analyze~~

Theorem: Let f be cvx, $C_L^1(\mathbb{R}^n)$; Let $\alpha_k = 1/L$.

Let $x_{k+1} = x_k - \alpha_k \nabla f(x_k)$; Then

$$f(x_{k+1}) - f(x^*) \leq \frac{2L\|x_0 - x^*\|^2}{k+1} \equiv O(1/k)$$

i.e. $O(1/\epsilon)$ iter to get ϵ' acc.