Today + Thursday: last new material before midterm.

HW #3 due Tuesday

Midterm on next Thursday.

- heavily based on HW.
- solutions for HW #1, 2, posted
- solutions for HW #3 won't be available for midterm.
- next Tues: "review session"

Descent methods and gradient descent

Lecture ~~08~~ 09

September 28, 2021

Beck, sections 4.1-4.2

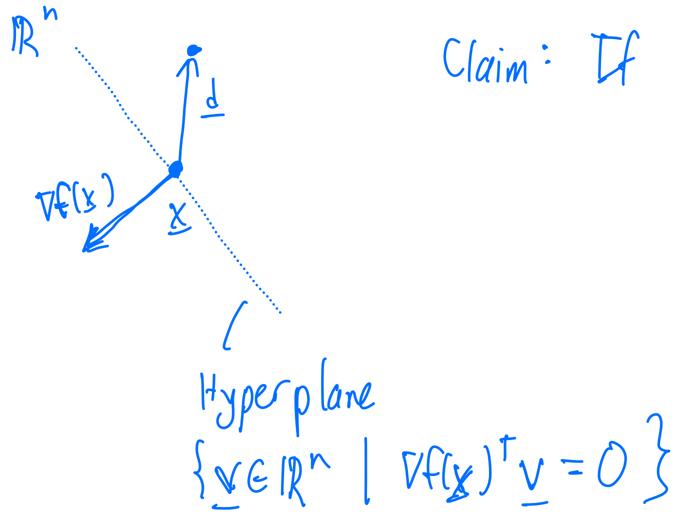
Minimization and descent

Goal: $\min_{\underline{x} \in \mathbb{R}^n} f(\underline{x})$, $\quad f : \mathbb{R}^n \to \mathbb{R}$

Ideas: (1) Try to computationally find the minimum by traveling "downhill" in $\mathbb{R}^n$

(2) Probably in a good spot ("solution") if we descend far enough so that $\nabla f = \underline{0}$..

Suppose we're starting at a point $\underline{x} \in \mathbb{R}^n$



Claim: If $\nabla f(\underline{x})^T \underline{d} < 0$, then moving in direction $\underline{d}$ decreases value of f.

"Proof" of claim: $\nabla f(\underline{x})^T \underline{d} = f'(\underline{x}; \underline{d}) = \lim_{t \downarrow 0} \dfrac{f(\underline{x}+t\underline{d}) - f(\underline{x})}{t}$

if $\nabla f(\underline{x})^T \cdot \underline{d} < 0 \implies$ for sufficiently small t,

$$\frac{f(\underline{x}+t\underline{d}) - f(\underline{x})}{t} < 0$$

$\implies f(\underline{x}+t\underline{d}) < f(\underline{x})$ (for t sufficiently small)

Definition: Assume $f \in C^1$. Fix $\underline{x} \in \mathbb{R}^n$. Any vector $\underline{d} \neq \underline{0}$ that satisfies

$$\nabla f(\underline{x})^T \underline{d} = f'(\underline{x}; \underline{d}) < 0$$

is called a descent vector or descent direction.

Descent directions

(See above)

Algorithms for descent directions

Descent directions immediately reveal an algorithm:

Given "starting point"/"initialization" $\underline{x}_0 \in \mathbb{R}^n$, then:

1.) Compute $\nabla f(\underline{x}_k)$, and choose a descent direction $\underline{d}_k$.

2.) Choose a value of $t_k \in \mathbb{R}_{++}$, called a "stepsize"

3.) $\underline{x}_{k+1} = \underline{x}_k + t_k \underline{d}_k$

4.) Decide to stop if $\underline{x}_{k+1}$ is a local minimum, otherwise, set $k \leftarrow k+1$, go back to step 1.

Challenges:

- how to initialize? ($\underline{x}_0$)
- which descent direction to choose? ($\underline{d}_k$)
- what stepsize? (t_k)
- when to stop? (termination criterion)

Stopping/Termination criterion: gradient norm

When to stop?

Recall: we're looking for a stationary point, so

let's stop when $\|\nabla f(\underline{x}_k)\|_2$ is "small" enough.

There are alternatives: e.g. $\|\underline{x}_{k+1} - \underline{x}_k\|_2$ being "small"

or $f(\underline{x}_k)$ is "small" enough.

None of these conditions is bullet-proof.

Ex. $f(x) = 10^{-10} x^2$

$\underset{x \in \mathbb{R}}{\operatorname{argmin}} f(0) = 0$

At $x = 1000$

$f'(x) = 2 \cdot 10^{-10} x$

$f'(1000) = 2 \cdot 10^{-7}$

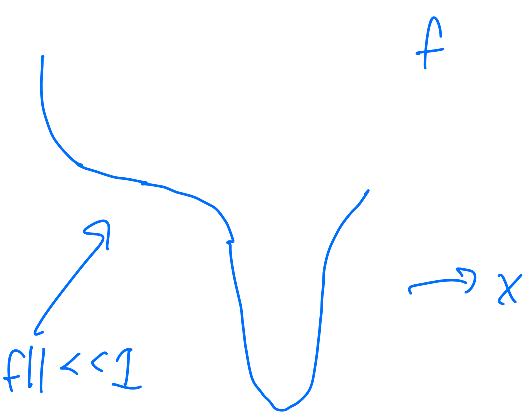
really small, but

$|1000 - 0|$ is very large

where I am.

global min

Ex. graph of f looks like:

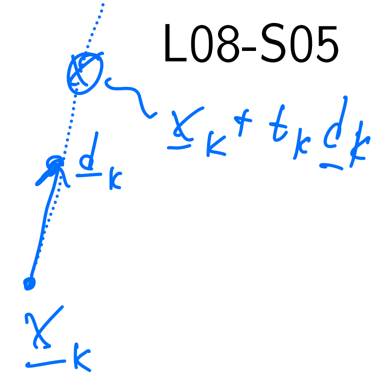


$\|\nabla f\| << 1$

So $\|\nabla f\|$, $\|\underline{x}_{k+1} - \underline{x}_k\|$ can be very small, but far away from local min.

Step size tuning: linesearch

How to choose stepsize t_k?



3 general strategies:

- "constant" stepsize: $t_k = t > 0$ for some chosen $t > 0$.

  (easy to implement, but prone to "mistakes")

- "linesearch": choose $t_k = \underset{t \in \mathbb{R}_{++}}{\arg\min}\, f(\underline{x}_k + t\underline{d}_k)$

  i.e., choose t_k as the scalar that minimizes f.

(generally much more robust, but is expensive)

- "backtracking" / "backtracking linesearch":

  Compromise between constant stepsize and linesearch.

  Specify parameters $s > 0$, $\alpha > 0$, $\beta > 0$, $\beta < 1$.

  initial stepsize

  expected decrease in f

  granularity of backtracking,

Idea: $f(\underline{x} + t\underline{d}) - f(\underline{x}) \sim t \nabla f(\underline{x})^T \underline{d}$

So: $f(\underline{x}) - f(\underline{x} + t\underline{d})$: actual "improvement" in f by taking stepsize t.

$-t \nabla f(\underline{x})^T \underline{d}$: asymptotic $(t \downarrow 0)$ improvement in f.

Step is "good enough" if

$$f(\underline{x}) - f(\underline{x} + t\underline{d}) \geq \alpha(-t \nabla f(\underline{x})^T \underline{d})$$

(typically $\alpha < 1$)

If things are "not" good enough, then

$$t \leftarrow \beta t. \quad (\beta \in (0,1))$$

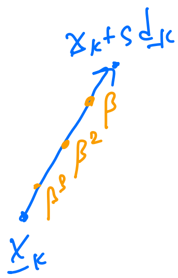
Backtracking algorithm: given $\underline{d}_k$, $\underline{x}_k$

Set $t_k = s$

While $f(\underline{x}_k) - f(\underline{x}_k + t_k \underline{d}_k) < \alpha(-t_k \nabla f(\underline{x}_k)^T \underline{d}_k)$

$t_k \leftarrow \beta t_k$

end



I.e. find smallest η (integer) s.t.

$$f(\underline{x}_k) - f(\underline{x}_k + s\beta^{\eta}\underline{d}_k) \geq -\alpha s \beta^{\eta} \nabla f(\underline{x}_k)^T \underline{d}_k.$$

Pro: avoid global linesearch,

Cons: choose β, α, s.

How to choose $\underline{x}_0$? (How to initialize)

No satisfactory general principles.

- "arbitrary", e.g. $\underline{x}_0 = \underline{0}$.
- "randomly", i.e. $\underline{x}_0$ is a multivariate Gaussian draw.
- pilot runs: e.g. $y_1, \dots y_m$ are some selected points in $\mathbb{R}^n$

$$\text{choose } \underline{x}_0 = \underset{j=1..m}{\operatorname{argmin}} f(y_j)$$

Unfortunately: choice of $\underline{x}_0$ has substantial impact on effectiveness of descent algorithms.

Linesearch for quadratic functions

Typically, global/exact linesearch in infeasible. But in special cases, we can compute things on paper:

Ex. (Linesearch for quadratic functions)

$$f(\underline{x}) = \underline{x}^T \underline{\underline{A}}\, \underline{x} + 2\underline{b}^T \underline{\underline{A}}\, \underline{x} + c, \quad \underline{\underline{A}} \succ \underline{\underline{0}}, \quad \underline{b}, c$$

are given.

Exact linesearch: at $\underline{x}$, identify $\underline{d}$ s.t. $\underline{d}^T \nabla f(\underline{x}) < 0$

Stepsize : $t_* = \underset{t>0}{\arg\min}\ f(\underline{x}+t\underline{d})$

define $g(t) = f(\underline{x}+t\underline{d})$

i.e., $\underset{t>0}{\arg\min}\ g(t)$

$$g(t) = (\underline{x}+t\underline{d})^T \underline{\underline{A}} (\underline{x}+t\underline{d}) + 2b^T \underline{\underline{A}} (\underline{x}+t\underline{d}) + c$$

$$= \underbrace{\underline{x}^T \underline{\underline{A}}\,\underline{x} + 2\underline{b}^T \underline{\underline{A}}\,\underline{x} + c}_{f(\underline{x})} + t^2 \underline{d}^T \underline{\underline{A}}\,\underline{d} + 2t d^T \underline{\underline{A}}\,\underline{x} + 2t \underline{b}^T \underline{\underline{A}}\,\underline{d}$$

$$= f(\underline{x}) + t^2 \underline{d}^T \underline{\underline{A}}\,\underline{d} + t d^T [2 \underline{\underline{A}}\,\underline{x} + 2 \underline{\underline{A}}^T \underline{b}]$$

$$g''(t) = 2\underline{d}^T \underline{\underline{A}}\,\underline{d} > 0 \quad (\underline{\underline{A}} \succ \underline{0},\ \underline{d} \neq \underline{0})$$

$$g'(t) = 2t\underline{d}^T \underline{\underline{A}}\,\underline{d} + d^T [2\underline{\underline{A}}\,\underline{x} + 2\underline{\underline{A}}^T \underline{b}] = 0$$

stationary pts.

$$t = \frac{-d^T[2\underline{\underline{A}}\,\underline{x} + 2\underline{\underline{A}}^T \underline{b}]}{2\underline{d}^T \underline{\underline{A}}\,\underline{d}}$$

Note: $\nabla f(\underline{x}) = 2\underline{\underline{A}}\,\underline{x} + 2\underline{\underline{A}}^T b$

$$\Longrightarrow t = \frac{-d^T \nabla f(\underline{x})}{2\underline{d}^T \underline{\underline{A}}\,\underline{d}}$$ (stationary point, global min since $g'' > 0$)

$$\underset{t \geq 0}{\operatorname{argmin}}\, g(t) = \underbrace{\frac{-\underline{d}^T \nabla f(\underline{x})}{2d^T \underline{\underline{A}}\, d}}_{\displaystyle \underset{\displaystyle t_*}{\shortparallel}} > 0.$$

Direction tuning: gradient descent

How to choose $\underline{d}_k$ in descent algorithms?

Recall: for small t, Taylor's theorem says:

$$f(\underline{x}_k) - f(\underline{x}_k + t\underline{d}_k) \sim -t\nabla f(\underline{x}_k)^T \underline{d}_k$$

So: let's pick $\underline{d}_k$ s.t. $-\nabla f(\underline{x}_k)^T \underline{d}_k$ is as large as possible.

Cauchy-Schwarz: $-\nabla f(\underline{x}_k)^T \underline{d}_k$ is maximized when $\underline{d}_k$ is parallel to $-\nabla f(\underline{x}_k)$.

Common choice of descent direction is

$$\underline{d}_k = -\nabla f(\underline{x}_k)$$

Algorithm: Gradient descent

$$\underset{\underline{x} \in \mathbb{R}^n}{\text{argmin}} \; f(\underline{x})$$

Given initialization $\underline{x}_0 \in \mathbb{R}^n$. Set $k=0$.

1.) Compute $\nabla f(\underline{x}_k)$.

2.) Choose a stepsize based on descent direction $\underline{d}_k = -\nabla f(\underline{x}_k)$. (E.g. t_k based on backtracking)

3.) $\underline{x}_{k+1} = \underline{x}_k - t_k \nabla f(\underline{x}_k)$

4.) If $\underline{x}_{k+1}$ is "good enough", stop, otherwise, set $k \leftarrow k+1$, go back to Step 1

Gradient descent with linesearch on quadratic functions

Recall: exact linesearch on $f(\underline{x}) = \underline{x}^T \underline{\underline{A}}\,\underline{x} + 2\underline{b}^T \underline{\underline{A}}\,\underline{x} + c$, $\underline{\underline{A}} \succ \underline{\underline{0}}$

is given by $t_* = \dfrac{-\underline{d}^T \nabla f(\underline{x})}{2\,\underline{d}^T \underline{\underline{A}}\,\underline{d}}$ (given $\underline{d}$).

Under gradient descent: $t_* = \dfrac{\|\nabla f(\underline{x})\|_2^2}{2\,\underline{d}^T \underline{\underline{A}}\,\underline{d}} = \dfrac{\|\nabla f(\underline{x})\|_2^2}{2\nabla f(\underline{x})^T \underline{\underline{A}}\,\nabla f(\underline{x})}$

$$= \frac{\|\nabla f(\underline{x})\|_2^2}{\nabla f(\underline{x})^T \nabla^2 f(\underline{x}) \nabla f(\underline{x})} = \frac{1}{R_{2\underline{\underline{A}}}(\nabla f(\underline{x}))}$$

$$= \frac{1}{R_{\nabla^2 f}(\nabla f(\underline{x}))}$$

Gradient descent: orthogonality of corrections

(Skip)

Punch line: grad descent w/ exact linesearch

$$\Longrightarrow (\underline{x}_{k+1} - \underline{x}_k)^T (\underline{x}_k - \underline{x}_{k-1}) = 0.$$

Convergence of gradient descent

$f \in C^1$ and ∇f is bounded.

Theorem: Assume f is "smooth enough", and assume $\{\underline{x}_k\}_{k=0}^{\infty}$ produced with gradient descent using any of the following stepsizes:

(a) $t_k = t > 0$ for t "small enough", $\left(t \lesssim \frac{1}{L}\right)$

(b) t_k chosen through exact linesearch

(c) t_k chosen through backtracking linesearch with any $s > 0$, $\alpha \in (0,1)$, $\beta \in (0,1)$.

(Recall: s is initial stepsize
α is tolerance relative to $-\nabla f^T \underline{d}$
β is geometric series parameter.)

Then: (i) $f(\underline{x}_{k+1}) - f(\underline{x}_k) \leq 0 \quad \forall k$, with equality iff $\nabla f(\underline{x}_k) = 0$.

iteration does decrease value of f

(ii) $\lim_{k \to \infty} \|\nabla f(\underline{x}_k)\|_2 = 0$.

converge to a stationary point.

We don't know:

- convergence to a global min
- convergence to a local min.

Computational examples