Data Science for Actuaries (ACT6100)

Arthur Charpentier

Rappels # 4.3 (Convex Optimization)

automne 2020

https://github.com/freakonometrics/ACT6100/

Convex Optimization Problem

$$\min_{\boldsymbol{x}}\{f(\boldsymbol{x})\}$$

with f convex, and differentiable.

Algorithm 1: Gradient Descent

```
initialization : $\boldsymbol{x}^{(0)}$;
for $t$=1,2,... do
    $\boldsymbol{x}^{(t)} \leftarrow \boldsymbol{x}^{(t-1)} - h_t \nabla f\big(\boldsymbol{x}^{(t-1)}\big)$
```

Heuristics: Taylor expansion

$$f(\boldsymbol{y}) \sim f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top\big(\boldsymbol{y}-\boldsymbol{x}\big) + \frac{1}{2h}\|\boldsymbol{y}-\boldsymbol{x}\|^2$$

Convergence

If f is convex, differentiable and such that ∇f is Lipschitz continuous with some constant $\gamma > 0$, i.e.

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\| \leq \gamma \|\boldsymbol{x} - \boldsymbol{y}\|$$

then if $h < 1/\gamma$,

$$f(\boldsymbol{x}^t) - f^\star \leq \frac{\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^\star\|^2}{2ht}$$

i.e. gradient descent converges at rate $1/t$, or we can find some ϵ-suboptimal point in $1/\epsilon$ iterations.
If f is non-convex, differentiable and such that ∇f is Lipschitz continuous with some constant $\gamma > 0$, gradient descent converges at rate $1/\sqrt{t}$

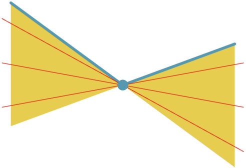
Non Differentiable Case

If f is convex and differentiable

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top (\boldsymbol{y} - \boldsymbol{x})$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \text{dom}(f)$.
If f is convex and non-differentiable, for all $\boldsymbol{x}$, there is $\boldsymbol{g}$ such that

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}^\top (\boldsymbol{y} - \boldsymbol{x})$$



for all $\boldsymbol{y} \in \text{dom}(f)$.

$\boldsymbol{g}$ is called subgradient at point $\boldsymbol{x}$.
If f differentiable at $\boldsymbol{x}$, $\boldsymbol{g}$ is unique and $\boldsymbol{g} = \nabla f(\boldsymbol{x})$

Non Differentiable Case

The set of subgradients of a convex function f is the subdifferential,

$$\partial f(\boldsymbol{x}) = \left\{\boldsymbol{g} \in \mathbb{R}^n : g \text{ is a subgradient at } \boldsymbol{x}\right\}$$

Note that $\partial f(\boldsymbol{x})$ is a convex set, and if f is differentiable at point $\boldsymbol{x}$, $\partial f(\boldsymbol{x}) = \{\nabla \partial f(\boldsymbol{x})\}$
Proposition: for any f, $f(\boldsymbol{x}^\star) = f^\star$ if and only if $\mathbf{0} \in \partial f(\boldsymbol{x})$.

Convex Optimization Problem

$$\min_{\boldsymbol{x}}\{f(\boldsymbol{x})\}$$

with f convex, but nondifferentiable.

Algorithm 2: Subgradient 'Descent'

```
initialization : $\boldsymbol{x}^{(0)}$;
for $t$=1,2,... do
    $\boldsymbol{g}^{(t-1)} \in \partial f(\boldsymbol{x}^{(t-1)})$;
    $\boldsymbol{x}^{(t)} \leftarrow \boldsymbol{x}^{(t-1)} - h_t\ \boldsymbol{g}^{(t-1)}$
```

Note that it is not necessarily a descent, so pick

$$\boldsymbol{x}^\star = \text{argmin}\{f(\boldsymbol{x}^{(0)}), f(\boldsymbol{x}^{(1)}), f(\boldsymbol{x}^{(2)}), \cdots\}$$

From Gradient Descent to Newton's Method

$$\min_{\boldsymbol{x}}\{f(\boldsymbol{x})\}$$

with f convex, twice differentiable.

Algorithm 3: Newton's Method

```
initialization : $\boldsymbol{x}^{(0)}$;
for $t$=1,2,... do
    $\boldsymbol{H}_t \leftarrow \nabla^2 f\left(\boldsymbol{x}^{(t-1)}\right)$;
    $\boldsymbol{x}^{(t)} \leftarrow \boldsymbol{x}^{(t-1)} - \boldsymbol{H}_t^{-1}\,\nabla f\left(\boldsymbol{x}^{(t-1)}\right)$
```

Instead of

$$f(\boldsymbol{y}) \sim f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top(\boldsymbol{y}-\boldsymbol{x}) + \frac{1}{2h}\|\boldsymbol{y}-\boldsymbol{x}\|^2$$

use a better quadratic approximation – $\frac{1}{h}\mathbb{I} \;\rightarrow\; H$,

$$f(\boldsymbol{y}) \sim f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top(\boldsymbol{y}-\boldsymbol{x}) + \frac{1}{2}(\boldsymbol{y}-\boldsymbol{x})^\top H(\boldsymbol{y}-\boldsymbol{x})$$

Newton (1685) - Raphson (1690)

Let $g(\boldsymbol{x}) = \nabla f(\boldsymbol{x})$. Assume that $g(\boldsymbol{x} + \vec{\boldsymbol{u}}) = 0$, then

$$0 = g(\boldsymbol{x} + \vec{\boldsymbol{u}}) \sim g(\boldsymbol{x}) + \nabla g(\boldsymbol{x})\vec{\boldsymbol{u}}$$

i.e. $\vec{\boldsymbol{u}} \sim -\nabla g(\boldsymbol{x})^{-1} g(\boldsymbol{x})$, which yields

$$\boldsymbol{x}^{(t)} \leftarrow \boldsymbol{x}^{(t-1)} + \vec{\boldsymbol{u}},\ \text{with } \vec{\boldsymbol{u}} = -\boldsymbol{H}_t^{-1}\, \nabla f\big(\boldsymbol{x}^{(t-1)}\big)$$

If computing the Hessian matrix $\boldsymbol{H}_t$ is complicated, one can approximate $\boldsymbol{H}_t$ by some (positive definite) matrix: quasi-Newton. Heuristically, use $\boldsymbol{H}'$ close to $\boldsymbol{H}_t$, symmetric, e.g.
$\boldsymbol{H}' = \boldsymbol{H}_t + a\boldsymbol{u}\boldsymbol{u}^\top$ (symmetric rank one update) or
$\boldsymbol{H}' = \boldsymbol{H}_t + a\boldsymbol{u}\boldsymbol{u}^\top + b\boldsymbol{v}\boldsymbol{v}^\top$ (symmetric rank two update), called
Broyden Fletcher Goldfarb Shanno (BFGS) method

Coordinate Descent

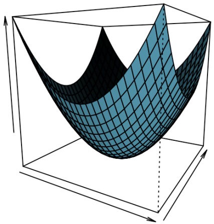
Let $\{\vec{\boldsymbol{e}}_1, \cdots, \vec{\boldsymbol{e}}_n\}$ denote the standard basis in $\mathbb{R}^n$,

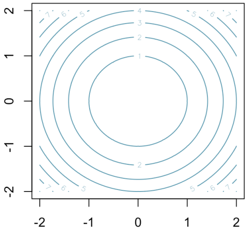
$$\vec{\boldsymbol{e}}_i = (0, \cdots, 0, 1, 0, \cdots, 0) \in \mathbb{R}^n$$

Proposition If $f : \mathbb{R}^n \to \mathbb{R}$ is convex, differentiable,

$$f(\boldsymbol{x}) \leq f(\boldsymbol{x} + \delta \vec{\boldsymbol{e}}_i),\ \forall i \Longrightarrow f(\boldsymbol{x}) = \min\{f\}$$

i.e. if we are at a point $\boldsymbol{x}$ such that $f(\boldsymbol{x})$ is minimized along each coordinate axis, then we have found a global minimizer.



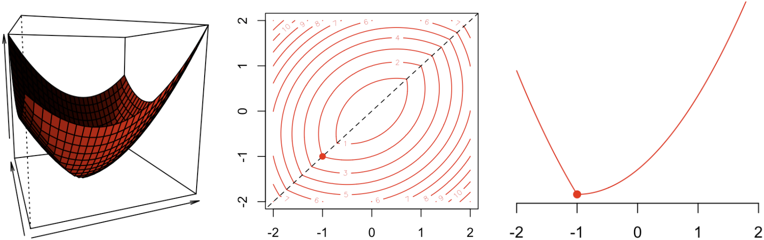


Coordinate Descent

Proposition If $f : \mathbb{R}^n \to \mathbb{R}$ is convex, but not differentiable,

$$f(\boldsymbol{x}) \le f(\boldsymbol{x} + \delta\vec{\boldsymbol{e}}_i),\ \forall i \not\Longrightarrow f(\boldsymbol{x}) = \min\{f\}$$

i.e. if we are at a point $\boldsymbol{x}$ such that $f(\boldsymbol{x})$ is minimized along each coordinate axis, then we have not found a global minimizer.



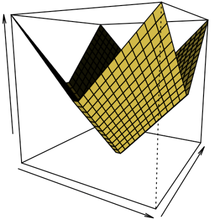
Coordinate Descent

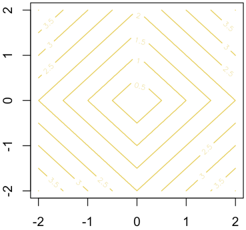
Proposition If $f:\mathbb{R}^n\to\mathbb{R}$ can be written

$$f(\boldsymbol{x})=g(\boldsymbol{x})+\underbrace{\sum_{i=1}^{n}h_i(\boldsymbol{x}_i),}_{\text{separable}}\quad\text{where}\quad\begin{cases} g\text{ convex and differentiable}\\ h_i\text{ convex and non-differentiable}\end{cases}$$

$$f(\boldsymbol{x})\leq f(\boldsymbol{x}+\delta\vec{\boldsymbol{e}}_i),\ \forall i\Longrightarrow f(\boldsymbol{x})=\min\{f\}$$

i.e. if we are at a point $\boldsymbol{x}$ such that $f(\boldsymbol{x})$ is minimized along each coordinate axis, then we have found a global minimizer.





Coordinate Descent

If we want to solve $\min\{f(\boldsymbol{x})\}$ for some $f:\mathbb{R}^n\rightarrow\mathbb{R}$ such that

$$f(\boldsymbol{x})=g(\boldsymbol{x})+\underbrace{\sum_{i=1}^{n}h_i(\boldsymbol{x}_i)}_{\text{separable}},\quad\text{where}\ \begin{cases}g\ \text{convex and differentiable}\\ h_i\ \text{convex and non-differentiable}\end{cases}$$

we can use a coordinate descent algorithm

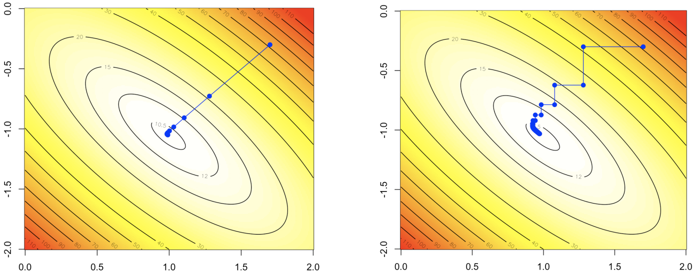
Algorithm 4: Coordinate Dscent

```
initialization : $\boldsymbol{x}^{(0)}$;
for $t$=1,2,... do
    for $j$=1,2,...,$n$ do
        $\boldsymbol{x}_j^{(t)} \leftarrow \text{argmin}\{f(\boldsymbol{x}_1^{(t)},\cdots,\boldsymbol{x}_{j-1}^{(t)},x_j,\boldsymbol{x}_{j+1}^{(t-1)},\cdots,\boldsymbol{x}_n^{(t-1)})\}$
```

Gradient vs. Coordinate Descent

Consider the problem $\min\{f(\beta)\}$ where $f(\beta)=\dfrac{1}{2}\|\boldsymbol{y}-\boldsymbol{X}\beta\|^2$

- Gradient descent, $\boldsymbol{\beta}\ \leftarrow\ \boldsymbol{\beta}+h\boldsymbol{X}^\top(\boldsymbol{y}-\boldsymbol{X}\beta)$
- Coordinate descent, $\boldsymbol{\beta}_j\ \leftarrow\ \boldsymbol{\beta}_j+\dfrac{1}{\boldsymbol{X}_j^\top\boldsymbol{X}_j}\boldsymbol{X}_j^\top(\boldsymbol{y}-\boldsymbol{X}\beta)$



Gradient vs. Coordinate Descent

to go further...

- Noisy descent

$$\boldsymbol{x}^{(t)} \leftarrow \boldsymbol{x}^{(t-1)} - h_t \, \nabla f\left(\boldsymbol{x}^{(t-1)}\right) + \varepsilon^{(t-1)}$$

where $\varepsilon^{(t-1)}$ is some zero-mean Gaussian noise, with decreasing variance.

- Simulated annealing, genetic algorithms, etc.