LECTURE 6

GRADIENT METHODS

- 1st order methods (better suited than 2nd order ones for large scale problems)
- we can often prove convergence to the optimal value and calculate the rate of convergence (i.e. how fast we are approaching the optimal value during the iteration)
- they can be interpreted (as other optimization methods) in terms of an operator $T: \mathbb{R}^n \to \mathbb{R}^n$ mapping an iterate into the following one

$$x^{k+1} = T x^k$$

the minimizer is then a fixed point of T.

- they can handle constrained problems (projected gradient method), non-differentiable objective functions (subgradient method), and they can converge to the optimal value in a finite number of steps (conjugate gradient method).

GENERAL CONCEPTS

All the methods discussed here take the form

$$\underbrace{x^{k+1}}_{\text{next iterate}} = \underbrace{x^k}_{\text{current iterate}} + \underbrace{t^k}_{\text{stepsize}} \underbrace{d^k}_{\text{direction}}$$

The goal is generally to design t^k and d^k such that the sequence $\{x^k\}_{k\geq 0}$ converges to the optimal solution of:

$\min f(x)$ with some desirable property.

If f is differentiable, we can introduce the concept of DESCENT DIRECTION.

Define $f'(x,d) = \nabla f(x)^T d$

Then, d is a descent direction of f at x if:

$$f'(x,d) < 0$$

This is an important property of d since, if that holds, we have (when f is differentiable)

DIRECTIONAL DERIVATIVE of f at x in direction d: $\lim_{t\to 0^+} \frac{f(x+td)-f(x)}{t} \overset{!}{=} f'(x,d) < 0$

Therefore, $\exists \varepsilon$ such that $\forall t \in (0, \varepsilon]$

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

That is, taking small enough stepsizes along a descent direction will always lead to a decrease of the objective function.

Typical choices for t:

- **constant stepsize**: $t^k = \bar{t} \quad \forall k$
  - not obvious which is a fixed value that will be good at every iteration ($t \leq \frac{1}{L}$ iff f is L-Lipschitz smooth)
  - large t might lead to non decreasing sequences (recall the previous result), while small t can have slow convergence
  - Main advantage: simplicity
- **exact line search**: t^k is given by

$$t^k = \arg\min_{t \in \mathbb{R}} f(x^k + t d^k)$$ ⚠

  - t^k minimizes f along the line $x^k + t d^k$
  - it seems the most obvious thing to do, but it is not always possible to determine the solution to ⚠
- **backtracking**: t^k is initialized with $s > 0$. Then, $t^k = s\beta^{i^k}$, where i^k is the smallest non negative integer for which this holds:

✅ $f(x^k) - f(x^k + s\beta^{i^k} d^k) \geq -\alpha s \beta^{i^k} f'(x^k, d^k)$

and $\alpha \in (0,1)$, $\beta \in (0,1)$ are fixed parameters.

  - This can be seen as a compromise between the previous 2 approaches. Indeed, if ✅ holds, then the function is guaranteed to be decreasing:

$$f(x^k) - f(\underbrace{x^k + t^k d^k}_{x_{k+1}}) \geq \underbrace{-\alpha t^k f'(x^k, d^k)}_{>0 \text{ (or } =0 \text{ if } x_{k+1} = x_k)}$$

GRADIENT METHODS

2x2:
- GRADIENT DESCENT
- STEEPEST DESCENT

$$d^k = -\nabla f(x^k)$$

- if $\nabla f(x_k) \neq 0$, then d_k is always a descent direction

$$f'(x^k, -\nabla f(x^k)) = -\|\nabla f(x^k)\|^2 < 0$$

- Moreover, this choice of d^k gives the steepest descent direction (Homework 1(b))
- Convergence is $O(\frac{1}{k})$ if is L-smooth and t^k is fixed, but can be improved to $O(\frac{1}{k^2})$ + accelerated schemes

When f is σ-strongly convex, convergence is linear.

- Consider for example $f(x) = x^T A x$

Then one can prove (Kantorovich inequality)

$$f(x^k) \leq c^k f(x^0)$$

$$c = \left(\frac{\text{cond}(A) - 1}{\text{cond}(A) + 1}\right)^2$$

$$\text{cond}(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$$ CONDITION NUMBER of A

$\uparrow$ cond $\Rightarrow \uparrow c \Rightarrow$ slower convergence to x^*

$1 \leq \text{cond} < \infty$ $\quad 0 \leq c < 1$

- For non-quadratic objective functions, what matters is $\text{cond}(\nabla^2 f)$

condition number of the Hessian.

In particular, if $\text{cond}(\nabla^2 f(x^*))$ is very large, the convergence to the optimal solution will be slow.

Example 1) $f(x_1, x_2) = 100\,(x_2 - x_1^2)^2 + (1 - x_1)^2$

$(x_1^*, x_2^*) = (1, 1)$, $f^* = 0$

$$\nabla f(x) = \begin{bmatrix} -400\, x_1 (x_2 - x_1^2) - 2(1 - x_1) \\ 200\,(x_2 - x_1^2) \end{bmatrix}$$

$$\nabla^2 f(x) = \begin{bmatrix} -400\, x_2 + 1200\, x_1^2 + 2 & -400\, x_1 \\ -400\, x_1 & 200 \end{bmatrix}$$

$$\nabla^2 f(1,1) = \begin{bmatrix} 802 & -400 \\ -400 & 200 \end{bmatrix}$$

$\lambda_{max} = 1001.6$

$\lambda_{min} = 0.3984$

$\rightarrow cond(\nabla f) \simeq 2.5 \cdot 10^3$

If we apply the gradient method, t^k is chosen with backtracking ($s = 2$, $\alpha = 0.25$, $\beta = 0.5$), and we start from $x^0 = (2, 5)$, we obtain

k	$f(x^k)$	$\lvert \nabla f(x^k) \rvert$
1	3.22	118.25
2	1.496	0.723
⋮	⋮	⋮
6889	0.00000	0.000018
6890	0.00000	0.000009

If $\lvert \nabla f \rvert < \varepsilon (= 10^{-5})$ was the stopping criterion, we would stop at $k = 6890$

SCALING → possible approach to pre-conditioning the problem

$S \in \mathbb{R}^{n \times n}$ nonsingular, $x = Sy$

An equivalent problem can be formulated as

$$\min g(y) (= f(Sy))$$

$$\nabla g(y) = S^T \nabla f(Sy) = S^T \nabla f(x)$$

Thus, the gradient method in the new variables is:

$$y^{k+1} = y^k - t_k S^T \nabla f(Sy^k)$$

while in the old variables is:

$$x^{k+1} = x^k - t_k \underbrace{S^T S}_{D} \nabla f(x^k)$$

$D \succ 0$ for any choice of S

$\Downarrow$

$-D\nabla f$ is a descent direction

$$\nabla g(y^k) = D^{\frac{1}{2}} \nabla f(D^{\frac{1}{2}} y^k) = D^{\frac{1}{2}} \nabla f(x^k)$$

$$\nabla^2 g(y^k) = D^{\frac{1}{2}} \nabla^2 f(D^{\frac{1}{2}} y^k) D^{\frac{1}{2}} = D_k^{\frac{1}{2}} \nabla^2 f(x^k) D_k^{\frac{1}{2}}$$

we can take a different scaling matrix at each k

How should we choose D_k?

Ideally $D_k^{\frac{1}{2}} \nabla^2 f(x^k) D_k^{\frac{1}{2}} \simeq I$

when $\nabla^2 f(x_k) \succ 0$, this is achieved with $D_k = \left(\nabla^2 f(x^k)\right)^{-1}$

grad. method + this choice of D → Newton's method

However, $\nabla^2 f$ might be not known and D^k costly to evaluate.

DIAGONAL SCALING

with pre-computed diagonal matrices D_i (simpler option)

SUBGRADIENT METHOD

Natural generalization of gradient method to the non-smooth case

$$d^k = -g^k, \quad g^k \in \partial f(x^k)$$

- Unlike $-\nabla f(x^k)$, the direction $-g^k$ is not necessarily a descent direction
- Unlike in the gradient method, we cannot always choose t^k such that the value of f always decreases

Example 2 $\quad f(x_1, x_2) = |x_1| + 2|x_2|$

What is the subdifferential at $x_1 = 1, x_2 = 0$?

We look for the set of $s = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$ such that

$$f(y) \geq \underbrace{f\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}\right)}_{=1} + s^T \left(y - \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) \quad \forall y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

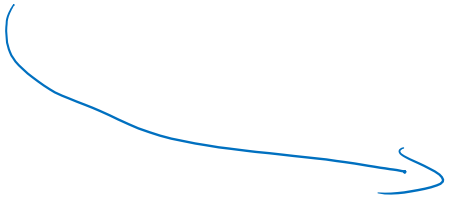
$$|y_1| + 2|y_2| \geq 1 + s_1(y_1 - 1) + s_2 y_2$$

This is valid $\forall y_1, y_2$ only if $\begin{cases} s_1 = 1 \\ |s_2| \leq 2 \end{cases}$

$$\partial f(1,0) = \left\{ \begin{bmatrix} 1 \\ b \end{bmatrix} : |b| \leq 2 \right\}$$

In particular, $\begin{bmatrix} 1 \\ 2 \end{bmatrix} \in \partial f(1,0)$.

However $-\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ is not a descent direction.



For any $t > 0$ it holds

$$f\left(\begin{bmatrix}1\\0\end{bmatrix} - t\begin{bmatrix}1\\2\end{bmatrix}\right) = f\left(\begin{bmatrix}1-t\\-2t\end{bmatrix}\right) = |1-t| + 4t$$

$$= \begin{cases} 1+3t, & t \in (0,1] \\ 5t-1, & t \geq 1 \end{cases}$$

Therefore $\forall t > 0$

$$f\left(\begin{bmatrix}1\\0\end{bmatrix} - t\begin{bmatrix}1\\2\end{bmatrix}\right) \geq 1 = f\left(\begin{bmatrix}1\\0\end{bmatrix}\right)$$

→ There is no point in the ray $\left\{\begin{bmatrix}1\\0\end{bmatrix} - t\begin{bmatrix}1\\2\end{bmatrix} : t > 0\right\}$ with a smaller function value than $\begin{bmatrix}1\\0\end{bmatrix}$

CONVERGENCE

The choice of stepsize will have here a big impact. Not only on the rate, but also on the convergence itself.

The usual assumptions are:

- f L-Lipschitz continuous
- $\|g\|_2 \leq G \quad \forall g \in \partial f$ (Homework 2(a))
- Result from slide [15]

$$f_{best}^k - p^* \leq \frac{\|x^0 - x^*\|_2^2 + G^2 \sum_{k=0}^{n} (t^k)^2}{2 \sum_{k=0}^{n} t^k}$$

- when $t_k = t$ → constant stepsize
  the RHS of ◎ tends to $\frac{G^2 t}{2}$ as $k \to \infty$
  Thus, f_{best}^k will converge to within $\frac{G^2 t}{2}$ of the optimal value.
- constant step length? (Homework 2(c))
- $\{t_k\}$ square summable but not summable: it converges → shown in the lecture
- $\{t_k\}$ converging to zero and not summable

Let $\varepsilon > 0$, $\exists N_1$ such that ☑ $t^k \le \frac{\varepsilon}{G^2}$ $\forall k > N_1$

$\exists N_2$: $\sum_{i=1}^{N_2} \alpha_i \ge \frac{1}{\varepsilon}\left(R^2 + G^2 \sum_{i=1}^{N_1} \alpha_i^2\right)$

Let $N = \max\{N_1, N_2\}$

Then for $k > N$:

$$\frac{\|x^0 - x^*\|_2^2 + G^2 \sum_{i=1}^{k} (t^i)^2}{2 \sum_{i=1}^{k} t^i} \le \frac{R^2 + G^2 \sum_{i=1}^{N_1} (t^i)^2}{2 \sum_{i=1}^{k} t^i} + \frac{G^2 \sum_{i=N_1+1}^{k} (t^i)^2}{2 \sum_{i=1}^{N_1} t^i + \sum_{i=N_1+1}^{k} t^i}$$

$$\le \frac{R^2 + G^2 \sum_{i=1}^{N_1} (t^i)^2}{\left(\frac{2}{\varepsilon}\right)\left(R^2 + G^2 \sum_{i=1}^{N_1} (t^i)^2\right)} + \frac{G^2 \sum_{i=N_1+1}^{k} \varepsilon t^i / G^2}{2 \sum_{i=N_1+1}^{k} t^i}$$

$$= \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$$

That is, the algorithm converges!
(Note indeed that as $k \to \infty$, $\varepsilon \to 0$ ☑)

CONJUGATE DIRECTION METHOD

Here $f = \frac{1}{2} x^T A x - bx$
$A \succ 0, A \in \mathbb{S}^n$

extensions to general convex (or even general nonlinear) exists, e.g. Fletcher-Reeves

$d^k = p^k$, t^k obtained with exact line search

where $p^k \in \{p^0, \dots, p^{n-1}\}$ is a set of conjugate vectors wrt A

$$p^{i\,T} A p^j = 0 \quad \forall i \neq j$$

- $\{p^0, \dots, p^{n-1}\}$ form a basis in $\mathbb{R}^n$

PROOF If $0 = \sum_{i=0}^{n-1} \alpha_i p^i$, then for $j \in \{0, \dots, n-1\}$

$$0 = p^{j\,T} A \left[\sum_{i=0}^{n-1} \alpha_i p^i\right] = \alpha_j p^{j\,T} A p^j$$

(for conjugacy)

Hence $\alpha_i = 0 \ \forall i \in \{0, \dots, n-1\}$ #

- One can thus write any $x \in \mathbb{R}^n$ as

$$x = \sum_{i=0}^{n-1} \beta_i p^i$$

This, together with conjugacy, can be used to show that the algorithm converges to x^*, f^* in at most n steps (Homework 3(a))

CONJUGATE GRADIENT METHOD (CGM)

While there are multiple ways to compute conjugate directions, CGM prescribes a precise sequence:

$r^k = \nabla f(x^k) = Ax^k - b$ is the gradient (residual)

Then $p^0 = -r^0$

$p^k = \underbrace{-r^k + \beta^k p^{k-1}}$ $(k > 1)$

linear combination of the steepest direction and the previous direction

β^k is chosen such that p^k and p^{k-1} are conjugate

$$0 = p^{k-1^T} A p^k = -p^{k-1^T} A r^k + \beta^k p^{k-1^T} A p^{k-1}$$

for conjugacy

$$\beta^k = \frac{p^{k-1^T} A r^k}{p^{k-1^T} A p^{k-1}}$$

- There is no need to know (and store) all previous directions. The only needed information is r^k and p^{k-1}, and conjugacy to the previous vectors is guaranteed.

$r^{k^T} p^i = 0 \quad \forall i = 0, \dots, k-1$ (subspace orthogonality)

ALL THESE PROPERTIES CAN BE USED IN HOMEWORK 3(b)

CONVERGENCE

- at most in n steps (but for large scale problems this can still be high)
- it depends on $\chi(A)$ in a similar way to what we have seen for gradient descent. Thus, also here preconditioning with appropriate matrices leads to improvements
- Interestingly, CGM's rate depends also on the distribution of the eigenvalues of A!

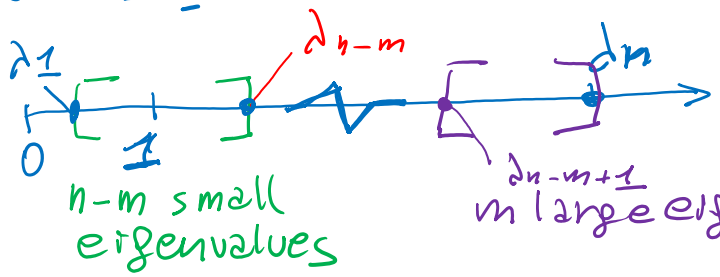
☑ If A has only r distinct eigenvalues, then the solution is found in at most r iterations

☑ If A has eigenvalues $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$, it holds:

$$\|x^{k+1} - x^*\|_A \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} + \lambda_1}\right)^2 \|x^0 - x^*\|_A$$

Implications?

Example of eigenvalue distribution



Then after m+1 steps we have

$$\|x_{m+1} - x^*\|_A \leq \underbrace{(\lambda_{n-m} - \lambda_1)}_{\varepsilon} \|x_0 - x^*\|_A$$

That is, after m+1 steps the CGM iterates will provide a good estimate of the solution (if ε is small).