Today: Conjugate Gradient

Goal: Let $A^T = A \in \mathbb{R}^{n\times n}$, solve $Ax = b$

A positive definite, i.e. $x^T A x > 0$, $\forall x \neq 0$

Idea: Turn $Ax=b$ into a minimization problem

Since A is positive definite, $\|x\|_A = \sqrt{x^T A x}$

Let $x_* \in \mathbb{R}^n$ be exact solution to $Ax = b$

x_* solves $Ax=b \iff x^* = \operatorname*{argmin}_{x\in\mathbb{R}^n} \|x - x_*\|_A^2$

$\iff x^* = \operatorname*{argmin}_{x\in\mathbb{R}^n} (x^T A x - 2b^T x)$

$$\boxed{\begin{aligned}&\|x-x_*\|_A^2\\ &= (x-x_*)^T A (x-x_*)\\ &= x^T A x - 2(Ax_*)^T x + \|x_*\|_A^2\end{aligned}}$$

Let $f(x) = \frac{1}{2} x^T A x - b^T x \in \mathbb{R}$.

want to find $x_* = \operatorname*{argmin}_{x\in\mathbb{R}^d} f(x) \longleftarrow$ optimization algorithms

---

• Method 1: Steepest gradient descent

Given $x_k \in \mathbb{R}^n$, try to find $x_{k+1} \in \mathbb{R}^n$

such that $x_{k+1} = x_k - \alpha_k \nabla f(x_k) \longleftarrow$ downhill direction

want $f(x_{k+1}) \leq f(x_k)$

so choose $\alpha_k = \operatorname*{argmin}_{\alpha\in\mathbb{R}} f(x_k - \alpha \underbrace{\nabla f(x_k)}_{-d_k}) \longleftarrow$ line search

Note that $\nabla f(x) = Ax - b$

$$f(x+\alpha y) = \frac{1}{2}(x+\alpha y)^T A (x+\alpha y) - b^T(x+\alpha y)$$

$$= \frac{1}{2}x^T A x - b^T x + \alpha y^T(Ax-b) + \frac{\alpha^2}{2} y^T A y$$

$$= f(x) + \alpha\, y^T(Ax-b) + \frac{\alpha^2}{2} y^T A y$$

$$d_k = -\nabla f(x_k) = b - Ax_k = r_k$$

$$f(x_k - \alpha d_k) = f(x_k) + \alpha\, d_k^T r_k + \frac{\alpha^2}{2} d_k^T A d_k$$

$$\Longrightarrow \quad \boxed{\alpha_k = \frac{d_k^T r_k}{d_k^T A d_k}} = \frac{r_k^T r_k}{r_k^T A r_k}$$

Note also that $r_k = b - A(x_{k-1} + \alpha_{k-1} d_{k-1})$

$$= r_{k-1} - \alpha_{k-1} A d_{k-1} = r_{k-1} - \alpha_{k-1} \underbrace{A r_{k-1}}_{w_{k-1}}$$

- Algorithm: Given $x_0 \in \mathbb{R}^n$, $r_0 = b - Ax_0 = d_0 \in \mathbb{R}^n$

For $k = 1, 2, 3, \cdots$

$$\overbrace{x_{k-1},\ r_{k-1},\ d_{k-1},\ w_{k-1},\ \alpha_{k-1}}^{4n+1}$$

$$w_{k-1} = A d_{k-1}$$

$$\alpha_{k-1} = \frac{d_{k-1}^T r_{k-1}}{d_{k-1}^T w_{k-1}}$$

- Cost: $O(n)$ memory

$O(n)$ flops

\+ compute one Ad per iteration

$$x_k = x_{k-1} + \alpha_{k-1} d_{k-1}$$

$$r_k = r_{k-1} - \alpha_{k-1} w_{k-1}$$

$$d_{k-1} = r_{k-1}$$

search direction has no memory

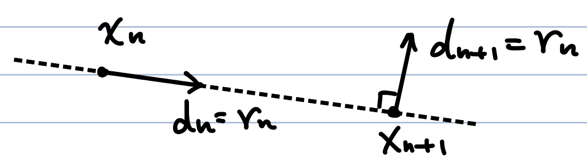
end

- Convergence of Steepest GD

1) Two consecutive search directions are orthogonal

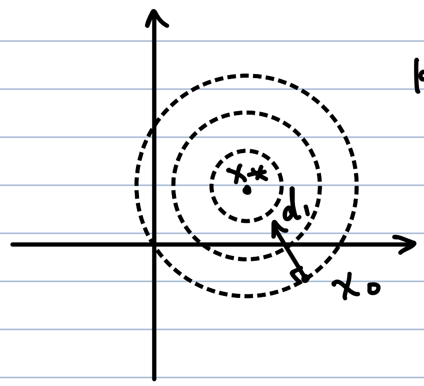
From $r_k = r_{k-1} - \alpha_{k-1} A r_{k-1}$

$$\xRightarrow{r_{k-1}^T \cdot} \quad r_{k-1}^T r_k = r_{k-1}^T r_{k-1} - \alpha_{k-1} r_{k-1}^T A r_{k-1} = 0$$



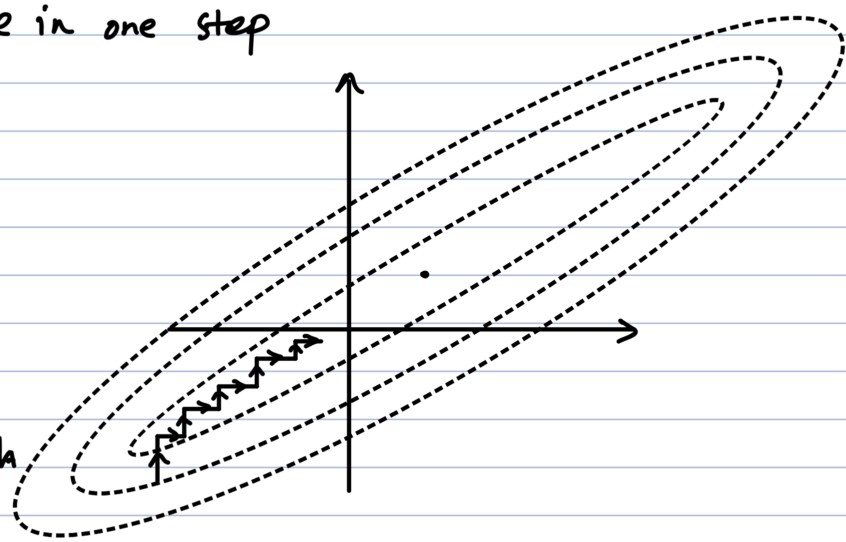
2) $\|x_k - x_*\|_A \le \left(\frac{\kappa_2(A)-1}{\kappa_2(A)+1}\right)^k \|x_0 - x_*\|_A$

Slow convergence if $\kappa_2(A) \gg 1$



$\kappa_2(A) = 1$

converge in one step



$\kappa_2(A) \gg 1$

$$\|x_k - x_*\|_A \le \left(1 - \frac{2}{\kappa_2(A)+1}\right)^k \|x_0 - x_*\|_A$$
$$\le \varepsilon$$

$$\Rightarrow \quad k \ge O\left(\frac{\log(\varepsilon)}{\log\left(1 - \frac{2}{\kappa_2(A)+1}\right)}\right) \approx O\left(\kappa_2(A) \log\left(\frac{1}{\varepsilon}\right)\right)$$

Pf:
$$\begin{aligned}
\|x_k - x_*\|_A^2 &= (x_k - x_*)^T A (x_k - x_*) \\
&= f(x_k) + x_*^T A x_* \\
(\forall \alpha \in \mathbb{R}) \quad &\le f(x_{k-1} + \alpha r_{k-1}) + x_*^T A x_* \\
&= (x_{k-1} + \alpha r_{k-1} - x_*)^T A (x_{k-1} + \alpha r_{k-1} - x_*)^T \\
&= \|(I - \alpha A)(x_{k-1} - x_*)\|_A \\
&\le \max_{1 \le i \le n} |1 - \alpha \lambda_i| \ \|x_{k-1} - x_*\|_A
\end{aligned}$$

A eig. values
$0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$

Note that $\min_{\alpha \in \mathbb{R}} \max_{1 \le i \le n} |1 - \alpha \lambda_i| = \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}$

$$\Rightarrow \quad \|x_k - x_*\|_A \le \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1} \|x_{k-1} - x_*\|_A \le \left(\frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1}\right)^k \|x_0 - x_*\|_A \quad \blacksquare$$

Method 2: Conjugate Gradient (Krylov subspace method)

Each step, we still do one-dimensional search

$$x_k = x_{k-1} + \alpha_{k-1} d_{k-1}$$

Choose $\alpha_{k-1} = \underset{\alpha \in \mathbb{R}}{\operatorname{argmin}} f(x_{k-1} + \alpha d_{k-1})$

$$\Rightarrow \boxed{\alpha_{k-1} = \frac{d_{k-1}^T r_{k-1}}{d_{k-1}^T A d_{k-1}}}$$

$$\Rightarrow \boxed{r_k = r_{k-1} - \alpha_{k-1} A d_{k-1}} \Rightarrow d_{k-1}^T r_k = 0 \leftarrow \text{search direction}$$

- How to choose "best" d_k ?
  - Because we do 1D search each step, after k steps,

$$x_k = x_0 + \sum_{i=0}^{k-1} \alpha_i d_i \quad , \quad \alpha_i \in \mathbb{R}$$

  - Hope to choose $d_0, d_1, \dots, d_{k-1}$

    s.t. 1) $\operatorname{span}\{d_0, \dots, d_{k-1}\} = \mathcal{K}_k(A, r_0) =: \mathcal{K}_k$

    2) $f(x_k) = \min_{x \in x_0 + \mathcal{K}_k} f(x)$

  - Since x_k is optimal, we have

$$\partial_{\alpha_j} f\left(x_0 + \sum_{i=0}^{k-1} \alpha_i d_i\right)$$

$$= \partial_{\alpha_j} \left[f(x_0) + \left(\sum_{i=0}^{k-1} \alpha_i d_i\right)^T (Ax_0 - b) + \frac{1}{2}\left(\sum_{i=0}^{k-1} \alpha_i d_i\right)^T A \left(\sum_{i=0}^{k-1} \alpha_i d_i\right)\right]$$

$$= d_j^T (Ax_0 - b) + d_j^T A \left(\sum_{i=0}^{k-1} \alpha_i d_i\right)$$

$$= d_j^T (Ax_k - b) = -d_j^T r_k \longleftarrow$$ when $\partial_{\alpha_j} f(x_0 + \sum_{i=0}^{k-1} \alpha_i d_i) = 0$,

$\Rightarrow$ $d_j^T r_k = 0, \ \forall j = 0, \dots, k-1$ (☆)

$$= d_j^T (A x_{j+1} - b) - d_j^T A \sum_{i=j+1}^{k-1} \alpha_i d_i$$

$$= \underbrace{-d_j^T r_{j+1}}_{=0} - \underbrace{d_j^T \sum_{i=j+1}^{k-1} \alpha_i A d_i}_{\sum_{i=j+1}^{k-1} \alpha_i d_j^T A d_i}$$

α_i's are fixed as long as d_i's are chosen

so want to choose d_i s.t. $d_j^T A d_i = 0$ $(j \neq i)$

<u>Idea:</u> As we generate the residual, use Gram-Schmidt to generate search directions that are A-conjugate.

- At the k^{th} step, we have compute $r_k, d_0, \dots, d_{k-1}$

then $d_k = r_k - \sum_{i=0}^{k-1} d_i \dfrac{d_i^T A r_k}{d_i^T A d_i}$ $\longleftarrow$ need to store all d_i's not a great idea!

- We can do better: Actually, $d_i^T A r_k = 0$, $i = 1, \dots, k-2$

1) Since $r_{i+1} = r_i - \alpha_i A d_i$ $\Rightarrow$ $A d_i = \dfrac{r_i - r_{i+1}}{\alpha_i}$

$$\Rightarrow d_i^T A r_k = \left(\frac{r_i - r_{i+1}}{\alpha_i} \right)^T r_k \quad (*) \quad , \quad i = 0, \dots, k-1$$

want to know $r_i^T r_k$

2) From (☆), $d_i^T r_k = 0$, $i = 0, 1, \dots, k-1$

we know $\left(r_i - \sum_{j=0}^{i-1} d_j \dfrac{d_j^T A r_i}{d_j^T A d_j} \right)^T r_k = 0 \Rightarrow r_i^T r_k = 0$, $i = 0, \dots, k-1$ (☆☆)

3) $\stackrel{(*)}{\Rightarrow}$ for $i = 0, \dots, k-2$, $d_i^T A r_k = 0$, $\longleftarrow$ $d_i \perp_A r_k$, $i = 0, \dots, k-2$

for $i = k-1$, $\dfrac{d_{k-1}^T A r_k}{d_{k-1}^T A d_{k-1}} = -\dfrac{r_k^T r_k}{d_{k-1}^T r_{k-1}} \stackrel{(☆)}{=} -\dfrac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$

– Combine everything

$$d_k = r_k + d_{k-1}\frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$$

- Algorithm: Conjugate Gradient

Given initial $x_0 \in \mathbb{R}^n$, $r_0 = b - Ax_0 \in \mathbb{R}^n$, $d_0 = r_0$

For $k = 1, 2, 3, \cdots$

$$w_{k-1} = Ad_{k-1}$$

$$\alpha_{k-1} = \frac{d_{k-1}^T r_{k-1}}{d_{k-1}^T w_{k-1}}$$

$$x_k = x_{k-1} + \alpha_{k-1} d_{k-1}$$

$$r_k = r_{k-1} - \alpha_{k-1} w_{k-1}$$

if $\|r_k\|_2 < \varepsilon$, break

$$d_k = r_k \boxed{+ d_{k-1} \underbrace{\frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}}_{\beta_k}}$$

end

only difference from Steepest Gradient Descent

$\overbrace{x_{k-1}, r_{k-1}, d_{k-1}, w_{k-1}, \alpha_{k-1}, \beta_{k-1}}^{4n+2}$

- Cost: $O(n)$ memory

compute one Ad + $O(n)$ flops per iteration

- Does $\text{span}\{d_0, \dots, d_{k-1}\} = K_k(A, r_0)$? ✓

Actually, also $K_k(A, r_0) = \text{span}\{r_0, r_1, \dots, r_{k-1}\}$

By induction, $\text{span}\{d_0\} = \text{span}\{r_0\} = K_1(A, r_0)$

Assume $\text{span}\{d_0, \dots, d_{k-1}\} = K_k(A, r_0)$

then $r_k = r_{k-1} - \alpha_{k-1} A d_{k-1} \in K_{k+1}(A, r_0)$

$d_k = r_k + d_{k-1}\beta_k \in K_{k+1}(A, r_0)$

but $r_k \perp r_i$, $i = 0, 1, \dots, k-1$ linearly indep. ($r_k \neq 0$)

$d_k \perp_A d_i$, $i = 0, 1, \dots, k-1$

$\Rightarrow K_{k+1}(A, r_0) = \text{span}\{r_0, r_1, \dots, r_{k-1}\} = \text{span}\{d_0, \dots, d_{k-1}\}$

CG is Krylov subspace method !

- Convergence of CG

From $x_k = x_0 + \sum_{j=0}^{k-1} \alpha_j d_j \in x_0 + \mathcal{K}_k$

$$\Rightarrow \quad f(x_k) = \min_{x \in x_0 + \mathcal{K}_k} f(x)$$

$$\Leftrightarrow \quad \| x_k - x_* \|_A = \min_{x \in x_0 + K_k} \| x - x_* \|_A$$

$$= \min_{p \in P_{k-1}} \| x_0 - x_* + p(A) r_0 \|_A$$

$$= \min_{p \in P_{k-1}} \| (I - p(A)A)(x_0 - x_*) \|_A$$

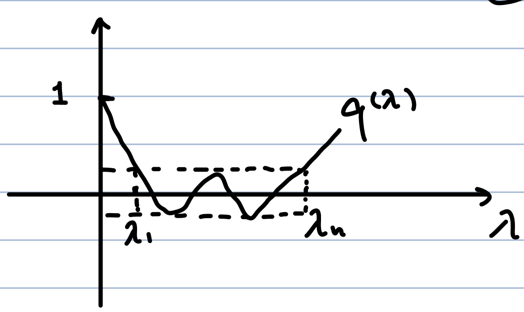
$$= \min_{\substack{q \in P_k \\ q(0)=1}} \| q(A)(x_0 - x_*) \|_A$$

Assume $A = Q^* \Lambda Q$ eigendecomposition $(A^T = A)$

$(0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n)$

$$\| x_k - x_* \|_A \le \underbrace{\min_{\substack{q \in P_k \\ q(0)=1}} \max_{1 \le i \le n} | q(\lambda_i) |}_{=: \, \mathcal{E}_k(A)} \, \| r_0 \|_A$$

if $k \ge n$, $\mathcal{E}_k(A) = 0$
CG converges in n steps



<u>Thm</u> $\displaystyle \min_{\substack{q = a_k \lambda^k + \cdots + a_0 \in P_k \\ a_k = 1}} \max_{\lambda \in [-1, 1]} | q(\lambda) | = 1$

achieved by $q_* = T_k(\lambda) \longleftarrow$ chebyshev polynomial

$$T_k(\lambda) = \begin{cases} \cos(k \arccos(\lambda)), & |\lambda| \le 1 \\ \frac{1}{2}\left(\lambda + \sqrt{\lambda^2 - 1}\right)^k + \frac{1}{2}\left(\lambda + \sqrt{\lambda^2-1}\right)^{-k}, & |\lambda| \geq 1 \end{cases}$$

Apply this to $\varepsilon_k(A)$

$$\varepsilon_k(A) \le \min_{\substack{q \in P_k \\ q(0)=1}} \max_{\lambda \in [\lambda_1, \lambda_n]} |q(\lambda)|$$

$$= \max_{\lambda \in [\lambda_1, \lambda_n]} \left| \frac{T_k\left(1 + 2\frac{\lambda - \lambda_n}{\lambda_n - \lambda_1}\right)}{T_k\left(1 + 2\frac{-\lambda_n}{\lambda_n - \lambda_1}\right)} \right|$$

$$= \frac{1}{\left|T_k\left(1 - \frac{2\lambda_n}{\lambda_n - \lambda_1}\right)\right|} = \frac{1}{\left|T_k\left(\frac{\kappa_2(A)+1}{\kappa_2(A)-1}\right)\right|}$$

Note that $\frac{\kappa_2(A)+1}{\kappa_2(A)-1} + \sqrt{\left(\frac{\kappa_2(A)+1}{\kappa_2(A)-1}\right)^2 - 1} = \frac{\sqrt{\kappa_2(A)}+1}{\sqrt{\kappa_2(A)}-1}$

$$\Rightarrow \varepsilon_k(A) \le \left(\frac{1}{2}\left(\frac{\sqrt{\kappa_2(A)}+1}{\sqrt{\kappa_2(A)}-1}\right)^k + \frac{1}{2}\left(\frac{\sqrt{\kappa_2(A)}+1}{\sqrt{\kappa_2(A)}-1}\right)^{-k} \right)^{-1}$$

$$\le 2\left(\frac{\sqrt{\kappa_2(A)}-1}{\sqrt{\kappa_2(A)}+1}\right)^k$$

When $A^* = A$ in GMRES, the error bound is similar!

$$\Rightarrow \|x_k - x_*\|_A \le 2\left(\frac{\sqrt{\kappa_2(A)}-1}{\sqrt{\kappa_2(A)}+1}\right)^k \|x_0 - x_*\|_A$$

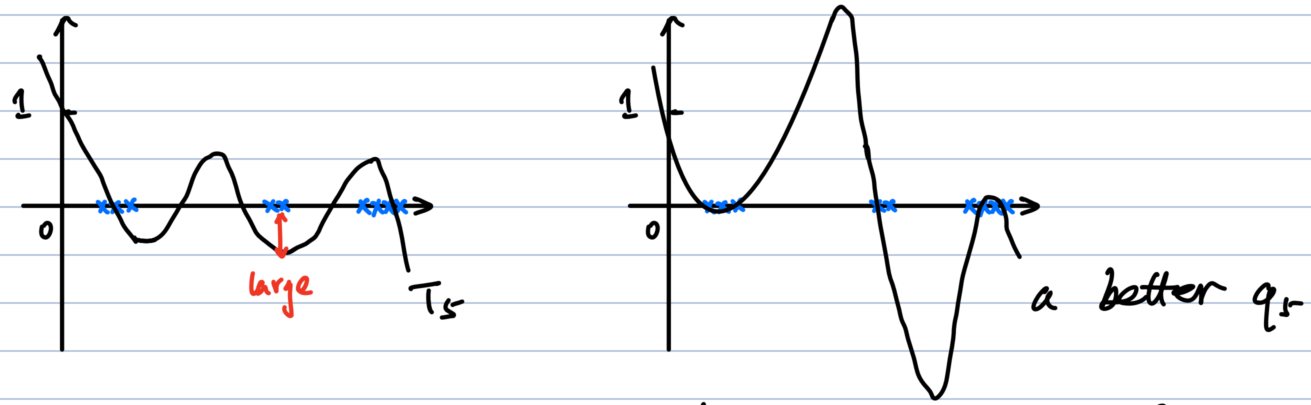
when $\kappa_2(A) \gg 1$, $\left(\frac{\sqrt{\kappa_2}-1}{\sqrt{\kappa_2}+1}\right)^k \approx \left(1 - \frac{2}{\sqrt{\kappa_2}}\right)^k \approx e^{-2k/\sqrt{\kappa_2}}$

to make $\|x_k - x_*\|_A \le \varepsilon$

we need $k \geq O\left(\sqrt{\kappa_2(A)} \log\left(\frac{1}{\varepsilon}\right)\right)$

better than steepest GD

- Remark 1: Only worst case bound, convergence can be faster



Actual convergence rate depends on the structure of the spectra of A. # cluster is important.

- Remark 2: CG is mathematically equivalent to FOM. Actually, we can derive CG from FOM directly by solving the equation $H_k y_k = \beta_1 e_1$ by LU decomposition.
(See pset 3)

- Remark 3: If A is not symmetric, can we use the idea of GMRES to extend CG? (CG is computationally saving)

In GMRES, $\|b - Ax_k\|_2 = \min_{x \in x_0 + K_k} \|b - Ax\|_2$

$$= \min_{x \in x_0 + K_k} \|A(x - x_*)\|_2$$

$$= \min_{x \in x_0 + K_k} \|x - x_*\|_{A^T A}$$

$\Rightarrow$ apply CG to $A^T A x = A^T b$ $\longleftarrow$ CGN / CGNR / Conjugate Residual...

But convergence rate is only $O\left(\left(1 - \frac{1}{\kappa_2(A)}\right)^k\right)$