1 March 9th, 2021

1.1 Convergence of Steepest Descent

Theorem 1.1
Let $A \in \mathbb{R}^{n\times n}$ be SPD. Then $\{x_k\}$ generated by steepest descent satisfies:

$$\|x_{k+1} - x_*\|_A \leq \frac{\gamma - 1}{\gamma + 1}\|x_k - x_*\|_A$$

where: $\gamma = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$ is the condition number of A.

Proof. Due to the optimality of x_{k+1}:

$$\begin{aligned}
\|x_{k+1} - x_k\|_A &= \min_{x\in x_k + \text{span}\{r_k\}} \|x - x_*\|_A \\
&= \min_{\beta\in\mathbb{R}} \|(x_k - \beta r_k) - x_*\|_A \\
&= \min_{\beta\in\mathbb{R}} \|(x_k - x_*) + \beta A(x_k - x_*)\|_A \\
&= \min_{p\in\mathbb{P}_1, p(0)=1} \|p(A)\cdot(x_k - x_*)\|_A \qquad \leq \left(\min_{p\in\mathbb{P}_1,p(0)=1}\|p(A)\|_A\right)\|x_k - x_*\|_A.
\end{aligned}$$

Where $\mathbb{P}_1$ is the set of all polynomial of degree 1. Note that minimizing over β is minimizing over $\mathbb{P}_1$ where $p(0) = 1$, since we can take $\{1 + \beta t : \beta \in \mathbb{R}\}$.

Lemma 1.2
Let $B \in \mathbb{R}^{n\times n}$. Then:

$$\|B\|_A = \|A^{\frac{1}{2}}BA^{-\frac{1}{2}}\|_2$$

where $A^{\frac{1}{2}}$ is the square root of A (using eigenvalue decomposition).

Proof.

$$\|B\|_A^2 = \max_{x\in\mathbb{R}^n}\frac{\langle ABx, Bx\rangle}{\langle Ax, x\rangle} = \max_{y\in\mathbb{R}^n}\frac{\left\langle A^{\frac{1}{2}}BA^{-\frac{1}{2}}y, A^{\frac{1}{2}}BA^{\frac{1}{2}}y\right\rangle}{\langle y, y\rangle} = \|A^{\frac{1}{2}}BA^{-\frac{1}{2}}\|_2^2$$

where $y = A^{\frac{1}{2}}y \iff x = A^{-\frac{1}{2}}y$ □

Then

$$|p(A)\|_A = \|A^{\frac{1}{2}}p(A)A^{-\frac{1}{2}}\|_2 = \|p(A)\|_2 = \max_{i=1,\dots,}|p(\lambda_i)|,$$

where λ_i are eigenvalues of A. Therefore:

$$\begin{aligned}
\|x_{k+1} - x_*\|_A &\leq \left(\min_{p\in\mathbb{P}_1,p(0)=1}\|p(A)\|_A\right)\|x_k - x_*\|_A \\
&\leq \left(\min_{p\in\mathbb{P}_1,p(0)=1}\max_{i=1,\dots,n}|p(\lambda_i)|\right)\|x_k - x_*\|_A \\
&\leq \left(\min_{p\in\mathbb{P}_1,p(0)=1}\max_{\lambda\in[\lambda_{\min},\lambda_{\max}]}|p(\lambda)|\right)\|x_k - x_*\|_A.
\end{aligned}$$

Remark 1.3 — This is taking the infinity norm, which will be discussed in further detail later.

This minimax occurs when:

$$|p(\lambda_{\min})| = |p(\lambda_{\max}) \text{ and } p\left(\frac{\lambda_{\min} + \lambda_{\max}}{2}\right) \implies p(t) = 1 - \frac{2}{\lambda_{\min} + \lambda_{\max}} t$$

Thus:

$$\min_{p \in \mathbb{P}_1, p(0)=1} \max_{\lambda \in [\lambda_{\min}, \lambda_{\max}]} |p(\lambda)| = 1 - \frac{2}{\lambda_{\min} + \lambda_{\max}} \lambda_{\min} = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{max} + \lambda_{\min}} = \frac{\gamma - 1}{\gamma + 1}$$

□

Remark 1.4 — Thus, the convergence speed is $\frac{\gamma-1}{\gamma+1}$.

To achieve an ϵ-solution, it suffices to have:

$$\left(\frac{\gamma - 1}{\gamma + 1}\right)^k \|x_0 - x_*\|_A \leq \epsilon \implies k \log\left(\frac{\gamma - 1}{\gamma + 1}\right) \leq \log\left(\frac{\epsilon}{\|x_0 - x_*\|_A}\right) \implies k \geq \frac{\log\left(\frac{\|x_0 - x_*\|_A}{\epsilon}\right)}{\log\left(\frac{\gamma+1}{\gamma-1}\right)}$$

Because:

$$\log\left(\frac{\gamma + 1}{\gamma - 1}\right) = \log\left(1 + \frac{2}{\gamma - 1}\right) \approx \frac{2}{\gamma} \text{ if } \gamma \text{ is large.}$$

Thus, we have:

$$k \geq \frac{1}{2} \log\left(\frac{\|x_0 - x_*\|_A}{\epsilon}\right) \cdot \gamma$$

Remark 1.5 — The number of iterations needed is $O(\gamma)$.

1.2 Two-Dimensional Projection Method (Conjugate Gradient)

We consider the projection method where $\dim(K) = 2$. We choose:

$$K = \operatorname{span}\{r_k, d_{k-1}\}$$

where:

$$\begin{cases} r_k = b - Ax_k & \text{(potential)} \\ d_{k-1} = x_{k-1} - x_k & \text{(momentum)} \end{cases}$$

A pictorial representation of this is given in Fig 1.

Remark 1.6 — Recall that r_k is the best possible direction in 1D.

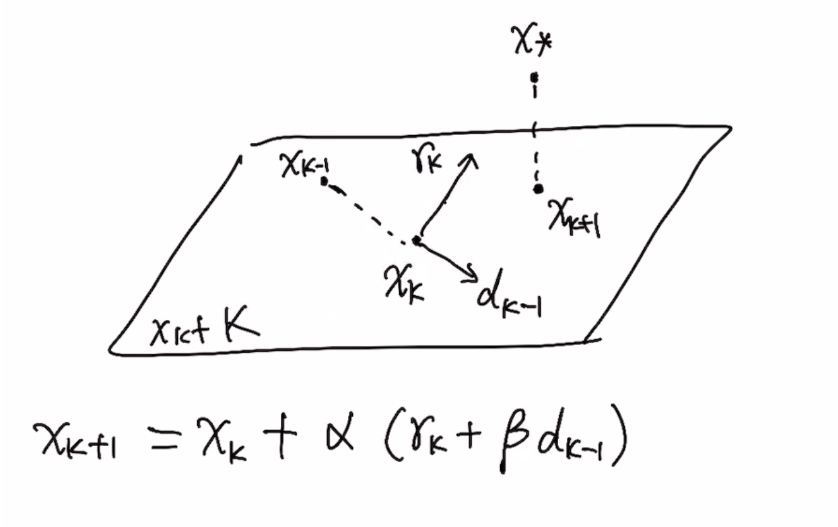

Figure 1: Pictorial Representation of $x_k + K$

Thus, we have:

$$x_{k+1} = x_k + \alpha_k(r_k + \beta_k d_{k-1}), \quad \text{where } \alpha_k, \beta_k \in \mathbb{R}$$

with:

$$(\alpha_k, \beta_k) = \arg\min_{\alpha,\beta\in\mathbb{R}} \|x_k + \alpha(r_k + \beta d_{k-1}) - x_*\|_A^2$$

Now we find the expressions of α_k, β_k:

$$\begin{aligned}\|x_k + \alpha(r_k + \beta d_{k-1}) - x_*\|_A^2 = \|x_k - x_*\|_A^2 + 2\alpha \langle x_k - x_*, r_k\rangle_A + \alpha^2\|r_k\|_A^2 \\ + 2\alpha\beta \langle x_k - x_*, d_{k-1}\rangle_A + \alpha^2\beta^2\|d_{k-1}\|_A^2 + 2\alpha^2\beta \langle r_k, d_{k-1}\rangle_A .\end{aligned}$$

Because x_k is the projection of x_* onto $x_{k-1} + \mathrm{span}\{r_{k-1}, d_{k-2}\}$, we have:

$$\langle x_k - x_*, d_{k-1}\rangle_A = 0$$

Taking derivative w.r.t α, β and setting them to 0, we have:

$$\begin{cases} 2\langle x_k - x_*, r_k\rangle_A + 2\alpha_k\|r_k\|_A^2 + 2\alpha_k\beta_k^2\|d_{k-1}\|_A^2 + 4\alpha_k\beta_k \langle r_k, d_{k-1}\rangle_A = 0 \\ 2\alpha_k^2(\beta_k\|d_{k-1}\|_A^2 + 2\langle r_k, d_{k-1}\rangle_A) = 0 \end{cases}$$

- If $\alpha_k = 0$, then: $x_{k+1} = x_k =$ projection of x_* onto $x_k + \mathrm{span}\{r_k, d_{k-1}\}$,

$$\begin{aligned} &\iff \langle x_* - x_k, r_k\rangle_A = 0 \\ &\iff \langle r_k, r_k\rangle = 0 \\ &\iff r_k = 0 \iff Ax_k = b \\ &\iff x_k = x_*. \end{aligned}$$

  x_k is the exact solution, and the algorithm stop.

- If $\alpha_k \neq 0$, then $r_k \neq 0$ and:

$$\beta_k = -\frac{\langle r_k, d_{k-1}\rangle_A}{\|d_{k-1}\|_A^2}$$

Then denote $d_k = r_k + \beta_k d_{k-1}$. So:

$$\begin{aligned} \alpha_k &= \arg\min_{\alpha\in\mathbb{R}} \|(x_k + \alpha d_k) - x_*\|_A \\ \iff \alpha_k &= \arg\min_{\alpha\in\mathbb{R}} \|x_k - x_*\|_A^2 + 2\alpha \langle d_k, x_k - x_*\rangle_A + \alpha^2 \|d_k\|_A^2. \end{aligned}$$

Taking the derivative and setting to zero, we have:

$$\begin{aligned} 2\alpha_k \|d_k\|_A^2 &= -2\langle d_k, x_k - x_*\rangle_A \\ \iff \alpha_k &= -\frac{\langle d_k, x_k - x_*\rangle_A}{\|d_k\|_A^2} = -\frac{\langle d_k, -r_k\rangle}{\langle Ad_k, d_k\rangle} \\ \iff \alpha_k &= \frac{\langle d_k, r_k\rangle}{\langle Ad_k, d_k\rangle}. \end{aligned}$$

Thus, we have Algorithm 1.

Algorithm 1 Conjugate Gradient Method

```
1: d_{-1} = 0
2: for k = 0, 1, 2, ... do
3:     r_k = b - A x_k
4:     β_k = - <r_k, A d_{k-1}> / <d_{k-1}, A d_{k-1}>
5:     d_k = r_k + β_k d_{k-1}
6:     α_k = <d_k, r_k> / <A d_k, d_k>
7:     x_{k+1} = x_k + α_k d_k
8: end for
```

Algorithm 1 requires 2 matrix-vector products + operations of $O(n)$. It can be improved to only requiring one matrix vector product, similarly to before, by introducing:

$$p_k = Ad_k.$$

Then we have:

$$\begin{aligned} Ax_{k+1} &= Ax_k + \alpha_k Ad_k \\ (b - Ax_{k+1}) &= (b - Ax_k) - \alpha_k Ad_k \\ r_{k+1} &= r_k - \alpha_k p_k. \end{aligned}$$

Remark 1.7 — Algorithm 2 only requires 1 matrix-vector product and opeartions of $O(n)$.

Definition 1.8. Algorithm 2 is called the **Conjugate Gradient (CG)** method.

Recall that the CG method is an optimization algorithm for:

$$\min_{x\in\mathbb{R}^n} f(x), \quad f(x) = \frac{1}{2}x^T Ax - x^T b.$$

For this problem, we have shown a few different variations of gradient descent:

Algorithm 2 2D Projection Method Improved

$d_{-1} = 0$
$p_{-1} = 0$
$r_0 = b - Ax_0$
for $k = 0, 1, 2, \ldots$ **do**
  $\beta_k = -\frac{\langle r_k, p_{k-1} \rangle}{\langle d_{k-1}, p_{k-1} \rangle}$
  $d_k = r_k + \beta_k d_{k-1}$
  $p_k = Ad_k$
  $\alpha_k = \frac{\langle d_k, r_k \rangle}{\langle p_k, d_k \rangle}$
  $x_{k+1} = x_k + \alpha_k d_k$
  $r_{k+1} = r_k - \alpha_k p_k$
end for

- (with exact line search) gives us Steepest Descent.
- (accelerated by momentum) gives us SOR Method.
- (simultaneously apply momentum and potential) gives us CG Method.
- (successively apply momentum then potential) gives us the **Nesterov momentum algorithm**.
- (successively apply potential then momentum) gives us gradient descent but with a larger step size (not optimal, which is exact line search).

For quadratic function $f(x)$ of the form

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

CG is preferred. For general non-linear function, Nesterov is preferred because α_k and β_k are not easy to obtain in CG. As such, Nesterov is popular in machine learning due to the complex objective function.

Remark 1.9 — If $f(x)$ is quadratic, we can get the optimal α_k and β_k, allowing us to update them simultaneously in CG.

Remark 1.10 — Best NN optimization function is probably just SGD with Nesterov acceleration.