

Chapter 11 Quasi-Newton Methods

- **Motivation** It takes $\mathcal{O}(d^3)$ to calculate $\nabla^2 f(x)^{-1}$ or solve for $\nabla^2 f(\mathbf{x}_t) \Delta \mathbf{x} = -\nabla f(\mathbf{x}_t)$.

The secant method

- **Motivation** $\frac{f(x_t) - f(x_{t-1})}{x_t - x_{t-1}} \approx f'(x_t)$, so $x_{t+1} := x_t - \frac{f(x_t)}{f'(x_t)} \approx x_{t+1} := x_t - f'(x_t) \frac{x_t - x_{t-1}}{f'(x_t) - f'(x_{t-1})}$
- In optimization regime, we want to find a similar matrix s.t. $\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1}) = H_t(\mathbf{x}_t - \mathbf{x}_{t-1})$ and so $\mathbf{x}_{t+1} = \mathbf{x}_t - H_t^{-1} \nabla f(\mathbf{x}_t)$.
 - This is called *secant condition*

Quasi-Newton methods

- **Definition** If H_t symmetric, and follows secant condition, the update method is *quasi-newton*.
- **Lemma Exercise 71** $f \in C^2$ and $\nabla^2 f \neq 0$, then Newton's method is a Quasi-Newton method iff $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top M \mathbf{x} - \mathbf{q}^\top \mathbf{x} + c$ with M invertable symmetric.
 - **Proof**
 - Newton is quasi $\Leftrightarrow \nabla f(y) - \nabla f(x) = \nabla^2 f(y)(y - x), \forall x, y$, take derivative w.r.t x , we get $\nabla^2 f(x) = \nabla^2 f(y), \forall x, y$ and this means f is a quadratic function, its invertable since every secant condition there is a solution. The other direction is straightforward.

Greenstadt's Approach

- We already have $H_{t-1}^{-1}, x_{t-1}, x_t$, we need H_t^{-1} , idea is $H_t^{-1} = H_{t-1}^{-1} + E_t$, and we want to minimize the general change $\|AEA^\top\|_F^2$.
- Denote $H := H_{t-1}^{-1}, H' := H_t^{-1}, E := E_t, \sigma := \mathbf{x}_t - \mathbf{x}_{t-1}, \mathbf{y} = \nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1})$, and $\mathbf{r} = \sigma - H\mathbf{y}$,
 - then the update formula is $H' = H + E$, such that $H'\mathbf{y} = \sigma$ or equivalently $E\mathbf{y} = \mathbf{r}$,
 - so the overall minimization is **minimize** $\frac{1}{2} \|AEA^\top\|_F^2$, **subject to** $E\mathbf{y} = \mathbf{r}$ and $E^\top - E = 0$.

Solving with Lagrange multiplier

- **Fact 11.2** If $f(E) := \frac{1}{2} \|AEB\|_F^2$, then $\nabla f(E) = A^\top AEBB^\top$ if define $\nabla f(E) = \left(\frac{\partial f(E)}{\partial E_{ij}} \right)$.
 - **Proof**
 - By this def, we have $\nabla_E \text{Tr}(AE) \nabla_E = \text{Tr}(E^\top A^\top) = A^\top$, so $f(E) := \frac{1}{2} \|AEB\|_F^2 = \frac{1}{2} \text{Tr}(B^\top E^\top A^\top AEB)$, then $\nabla f(E) = \nabla_E \frac{1}{2} \text{Tr}(E^\top A^\top AEBB^\top) + \nabla_E \frac{1}{2} \text{Tr}(BB^\top E_0^\top A^\top AE) = A^\top AEBB^\top / 2 + (BB^\top E^\top A^\top A)^\top / 2 = A^\top AEBB^\top$
- Denote $\lambda \in \mathbb{R}^d$ as the multiplier for d constraints of $E\mathbf{y} = \mathbf{r}$, and $\Gamma \in \mathbb{R}^{d \times d}$ as the multiplier for $d \times d$ constraints of $E^\top - E = 0$.
- For each equation of $\partial_{E_{ij}} f = \lambda^\top f_1 + \text{Tr}(\Gamma f_2)$, λ part yields a term of $\lambda_i y_i$ and Γ yields a term of $\Gamma_{ji} - \Gamma_{ij}$
- **Lemma 11.3** The above equation gives the optimal conditional of $WE^*W = \lambda \mathbf{y}^\top + \Gamma^\top - \Gamma$, where $W := A^\top A$ is symmetric and positive definite.

Solving Greenstadt family

- The minimization has now turn into three linear equation (i) $E\mathbf{y} = \mathbf{r}$, (ii) $E^\top - E = \mathbf{0}$ and (iii) $WEW = \boldsymbol{\lambda}\mathbf{y}^\top + \Gamma^\top - \Gamma$
- To eliminate Γ , by plug (iii) into (ii) we get $M(\boldsymbol{\lambda}\mathbf{y}^\top - \mathbf{y}\boldsymbol{\lambda}^\top + 2\Gamma^\top - 2\Gamma)M = \mathbf{0}$, where $M = W^{-1}$, so $\Gamma^\top - \Gamma = \frac{1}{2}(\mathbf{y}\boldsymbol{\lambda}^\top - \boldsymbol{\lambda}\mathbf{y}^\top)$
 - then $E = \frac{1}{2}M(\boldsymbol{\lambda}\mathbf{y}^\top + \mathbf{y}\boldsymbol{\lambda}^\top)M$
- Then to eliminate $\boldsymbol{\lambda}$, we plug in the secant condition (i) we get $\boldsymbol{\lambda} = \frac{1}{\mathbf{y}^\top M\mathbf{y}}(2M^{-1}\mathbf{r} - \mathbf{y}\boldsymbol{\lambda}^\top M\mathbf{y})$
 - multiply with $\mathbf{y}^\top M$, we get $z = \boldsymbol{\lambda}^\top M\mathbf{y} = \frac{\mathbf{y}^\top \mathbf{r}}{\mathbf{y}^\top M\mathbf{y}}$, so $\boldsymbol{\lambda} = \frac{1}{\mathbf{y}^\top M\mathbf{y}}\left(2M^{-1}\mathbf{r} - \frac{(\mathbf{y}^\top \mathbf{r})}{\mathbf{y}^\top M\mathbf{y}}\mathbf{y}\right)$
- Plug this into E , we get $E = \frac{1}{2}M(\boldsymbol{\lambda}\mathbf{y}^\top + \mathbf{y}\boldsymbol{\lambda}^\top)M = \frac{1}{\mathbf{y}^\top M\mathbf{y}}\left(\mathbf{r}\mathbf{y}^\top M + M\mathbf{y}\mathbf{r}^\top - \frac{(\mathbf{y}^\top \mathbf{r})}{\mathbf{y}^\top M\mathbf{y}}M\mathbf{y}\mathbf{y}^\top M\right)$, by definition $\mathbf{r} = \boldsymbol{\sigma} - H\mathbf{y}$, we get
 - $E^* = \frac{1}{\mathbf{y}^\top M\mathbf{y}}\left(\boldsymbol{\sigma}\mathbf{y}^\top M + M\mathbf{y}\boldsymbol{\sigma}^\top - H\mathbf{y}\mathbf{y}^\top M - M\mathbf{y}\mathbf{y}^\top H - \frac{1}{\mathbf{y}^\top M\mathbf{y}}(\mathbf{y}^\top \boldsymbol{\sigma} - \mathbf{y}^\top H\mathbf{y})M\mathbf{y}\mathbf{y}^\top M\right)$

BFGS (Broyden, Fletcher, Goldfarb and Shanno)

- **Definition** BFGS is when $M = H' = H_t^{-1}$, $M\mathbf{y} = H'\mathbf{y} = \boldsymbol{\sigma}$, even we don't know H' , but it never appears in the solution, so $E^* = \frac{1}{\mathbf{y}^\top \boldsymbol{\sigma}}\left(-H\mathbf{y}\boldsymbol{\sigma}^\top - \boldsymbol{\sigma}\mathbf{y}^\top H + \left(1 + \frac{\mathbf{y}^\top H\mathbf{y}}{\mathbf{y}^\top \boldsymbol{\sigma}}\right)\boldsymbol{\sigma}\boldsymbol{\sigma}^\top\right)$, where $H = H_{t-1}^{-1}$, $\boldsymbol{\sigma} = \mathbf{x}_t - \mathbf{x}_{t-1}$, $\mathbf{y} = \nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1})$.
 - Iteration cost is $O(d^2)$
- **Lemma Exercise 74.1** If f convex, $\mathbf{y}^\top \boldsymbol{\sigma} > 0$, unless $\mathbf{x}_t = \mathbf{x}_{t-1}$ or $f(\lambda\mathbf{x}_t + (1-\lambda)\mathbf{x}_{t-1}) = \lambda f(\mathbf{x}_t) + (1-\lambda)f(\mathbf{x}_{t-1})$ for all $\lambda \in (0, 1)$.
 - **Proof**
 - By property of convexity $\mathbf{y}^\top \boldsymbol{\sigma} = (\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1}))^\top (\mathbf{x}_t - \mathbf{x}_{t-1}) \geq 0$
 - If $(\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1}))^\top (\mathbf{x}_t - \mathbf{x}_{t-1}) = 0$ while $\exists \lambda$ s.t. $f(\lambda\mathbf{x}_t + (1-\lambda)\mathbf{x}_{t-1}) < \lambda f(\mathbf{x}_t) + (1-\lambda)f(\mathbf{x}_{t-1})$,
 - then by convexity $f(\lambda\mathbf{x}_t + (1-\lambda)\mathbf{x}_{t-1}) \geq f(\mathbf{x}_{t-1}) + \lambda \nabla f(\mathbf{x}_{t-1})^\top (\mathbf{x}_t - \mathbf{x}_{t-1})$ this means $f(\mathbf{x}_t) - f(\mathbf{x}_{t-1}) > \nabla f(\mathbf{x}_{t-1})^\top (\mathbf{x}_t - \mathbf{x}_{t-1})$, similarly $f(\mathbf{x}_{t-1}) - f(\mathbf{x}_t) > \nabla f(\mathbf{x}_t)^\top (\mathbf{x}_{t-1} - \mathbf{x}_t)$
 - add them together, we get $(\nabla f(\mathbf{x}_t) - \nabla f(\mathbf{x}_{t-1}))^\top (\mathbf{x}_t - \mathbf{x}_{t-1}) > 0$ contradiction.
- **Observation 11.6** $H' = \left(I - \frac{\boldsymbol{\sigma}\mathbf{y}^\top}{\mathbf{y}^\top \boldsymbol{\sigma}}\right)H\left(I - \frac{\mathbf{y}\boldsymbol{\sigma}^\top}{\mathbf{y}^\top \boldsymbol{\sigma}}\right) + \frac{\boldsymbol{\sigma}\boldsymbol{\sigma}^\top}{\mathbf{y}^\top \boldsymbol{\sigma}}$
 - **Proof**
 - $E^* + H = H + \frac{1}{\mathbf{y}^\top \boldsymbol{\sigma}}\left(-H\mathbf{y}\boldsymbol{\sigma}^\top - \boldsymbol{\sigma}\mathbf{y}^\top H + \left(1 + \frac{\mathbf{y}^\top H\mathbf{y}}{\mathbf{y}^\top \boldsymbol{\sigma}}\right)\boldsymbol{\sigma}\boldsymbol{\sigma}^\top\right) = \frac{\boldsymbol{\sigma}\boldsymbol{\sigma}^\top}{\mathbf{y}^\top \boldsymbol{\sigma}} + H\left(I - \frac{\mathbf{y}\boldsymbol{\sigma}^\top}{\mathbf{y}^\top \boldsymbol{\sigma}}\right) - \boldsymbol{\sigma}\mathbf{y}^\top H + \frac{\boldsymbol{\sigma}\mathbf{y}^\top H\mathbf{y}\boldsymbol{\sigma}^\top}{\mathbf{y}^\top \boldsymbol{\sigma}} = \text{QED}$
- **Lemma Exercise 74.2** If $H \succeq 0$ and $\mathbf{y}^\top \boldsymbol{\sigma} > 0$, then also H' is positive definite.
 - **Proof**
 - Since $C := \left(I - \frac{\mathbf{y}\boldsymbol{\sigma}^\top}{\mathbf{y}^\top \boldsymbol{\sigma}}\right) = \left(I - \frac{\boldsymbol{\sigma}\mathbf{y}^\top}{\mathbf{y}^\top \boldsymbol{\sigma}}\right)^\top$, so $H' = C^\top H C + \frac{\boldsymbol{\sigma}\boldsymbol{\sigma}^\top}{\mathbf{y}^\top \boldsymbol{\sigma}}$, $C^\top H C$ is semi positive definite and so is $\frac{\boldsymbol{\sigma}\boldsymbol{\sigma}^\top}{\mathbf{y}^\top \boldsymbol{\sigma}}$.
 - When $z \perp \boldsymbol{\sigma}^\top$, we have $Cz = z \neq 0$, so the two quadratic form will not be zero at the same time, this means positive definiteness.
- **Remark** Usually Newton or Quasi-Newton are performed with *scaled steps* $\mathbf{x}_{t+1} = \mathbf{x}_t - \alpha_t H_t^{-1} \nabla f(\mathbf{x}_t)$, either line search or backtracking line search (when $\alpha_t = 1$ is not good enough, do $\alpha_t/2$).

L-BFGS (limited memory version)

- **Idea** Only use information from the previous m iterations, for some small value of m .
- **Lemma 11.7** If an oracle can compute $\mathbf{s} = H\mathbf{g}$ for any vector \mathbf{g} , then $\mathbf{s}' = H'\mathbf{g}'$ can be computed with one oracle call of $\mathbf{s} = H\mathbf{g}$, and $O(d)$ arithmetic operation, assuming $\boldsymbol{\sigma}, \mathbf{y}$ known.
 - **Proof**

$$\blacksquare H' \mathbf{g}' = \underbrace{\left(I - \frac{\sigma \mathbf{y}^\top}{\mathbf{y}^\top \sigma} \right) H \underbrace{\left(I - \frac{\mathbf{y}^\top}{\mathbf{y}^\top \sigma} \right) \mathbf{g}'}_{\mathbf{g}}}_{\mathbf{s}} + \underbrace{\frac{\sigma \sigma^\top}{\mathbf{y}^\top \sigma} \mathbf{g}'}_{\mathbf{h}}$$

$$\underbrace{\hspace{10em}}_{\mathbf{w}}$$

$$\underbrace{\hspace{10em}}_{\mathbf{z}}$$

▪ $\mathbf{g}, \mathbf{h}, \mathbf{s}, \mathbf{w}, \mathbf{z}$ all are computed in $O(d)$.

- The idea is that we need $H_t^{-1} \nabla f_t$, and we can borrow from $H_{t-1}^{-1} \nabla f_t$, etc, and recurse back to $t = 0$, This gives the BFGS-step:

- **Algorithm (BFGS-STEP)**

- Input (k, \mathbf{g})

- If $k = 0$ then return $H_0^{-1} \mathbf{g}'$

- Else

- Set $\mathbf{h} = \sigma \frac{\sigma_k^\top \mathbf{g}'}{\mathbf{y}_k^\top \sigma_k}$, and $\mathbf{g} = \mathbf{g}' - \mathbf{y} \frac{\sigma_k^\top \mathbf{g}'}{\mathbf{y}_k^\top \sigma_k}$

- $\mathbf{s} = \text{BFGS-STEP}(k-1, \mathbf{g})$ (*recursive call*)

- $\mathbf{w} = \mathbf{s} - \sigma_k \frac{\mathbf{y}_k^\top \mathbf{s}}{\mathbf{y}_k^\top \sigma_k}$

- $\mathbf{z} = \mathbf{w} + \mathbf{h}$

- return \mathbf{z}

- **Remark** If H_0 can be computed in $O(d)$ the total runtime is $O(td)$, this is acceptable when $t \leq d$. It's natural to think of a cut-off version

- **Algorithm (L-BFGS-STEP)**

- Input (k, l, \mathbf{g})

- If $l = 0$ then return $H_0^{-1} \mathbf{g}'$

- Else

- Set $\mathbf{h} = \sigma \frac{\sigma_k^\top \mathbf{g}'}{\mathbf{y}_k^\top \sigma_k}$, and $\mathbf{g} = \mathbf{g}' - \mathbf{y} \frac{\sigma_k^\top \mathbf{g}'}{\mathbf{y}_k^\top \sigma_k}$

- $\mathbf{s} = \text{L-BFGS-STEP}(k-1, l-1, \mathbf{g})$ (*recursive call*)

- $\mathbf{w} = \mathbf{s} - \sigma_k \frac{\mathbf{y}_k^\top \mathbf{s}}{\mathbf{y}_k^\top \sigma_k}$

- $\mathbf{z} = \mathbf{w} + \mathbf{h}$

- return \mathbf{z}