

# CMPT 409/981: Optimization for Machine Learning

## Lecture 7

---

Sharan Vaswani

September 26, 2024

# Recap

**Polyak Momentum:** Compute the gradient at  $w_k$  and then extrapolate:

$$v_k = w_k + \beta_k(w_k - w_{k-1}); w_{k+1} = v_k - \eta \nabla f(w_k).$$

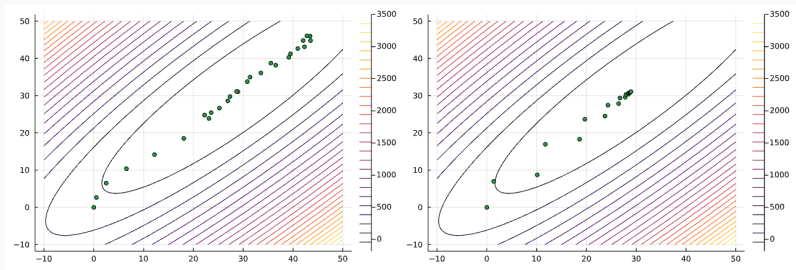
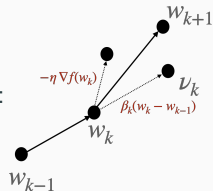


Figure 1: Comparing GD vs HB momentum (with theoretical  $(\eta, \beta)$ ) on a strongly-convex quadratic

# Minimizing strongly-convex quadratics with HB momentum

**Update:**  $w_{k+1} = w_k - \eta \nabla f(w_k) + \beta(w_k - w_{k-1})$

**Claim:** For  $L$ -smooth,  $\mu$ -strongly convex quadratics s.t.  $f(w) = \frac{1}{2}w^\top A w - bw + c$  where  $A$  is symmetric, positive semi-definite, HB momentum with  $\eta = \frac{4}{(\sqrt{L} + \sqrt{\mu})^2}$  and  $\beta = \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^2$

converges as:  $\|w_T - w^*\| \leq \sqrt{2} \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} + \epsilon_T\right)^T \|w_0 - w^*\|$ , where,  $\lim_{T \rightarrow \infty} \epsilon_T \rightarrow 0$ .

**Proof:**

$$\begin{aligned} \begin{bmatrix} w_{k+1} - w^* \\ w_k - w^* \end{bmatrix} &= \begin{bmatrix} w_k - w^* - \eta \nabla f(w_k) + \beta(w_k - w_{k-1}) \\ w_k - w^* \end{bmatrix} \\ &= \begin{bmatrix} w_k - w^* - \eta A(w_k - w^*) + \beta(w_k - w^*) - \beta(w_{k-1} - w^*) \\ w_k - w^* \end{bmatrix} \\ &\quad \text{(Since } \nabla f(w) = Aw, Aw^* = b\text{)} \\ \Rightarrow \begin{bmatrix} w_{k+1} - w^* \\ w_k - w^* \end{bmatrix} &= \begin{bmatrix} (1 + \beta)I_d - \eta A & -\beta I_d \\ I_d & 0 \end{bmatrix} \begin{bmatrix} w_k - w^* \\ w_{k-1} - w^* \end{bmatrix} \end{aligned}$$

If  $\beta = 0$ , we can recover the same equation as GD.

# Minimizing strongly-convex quadratics with HB momentum

$$\underbrace{\begin{bmatrix} w_{k+1} - w^* \\ w_k - w^* \end{bmatrix}}_{:= \Delta_{k+1} \in \mathbb{R}^{2d}} = \underbrace{\begin{bmatrix} (1 + \beta)I_d - \eta A & -\beta I_d \\ I_d & 0 \end{bmatrix}}_{:= \mathcal{H} \in \mathbb{R}^{2d \times 2d}} \underbrace{\begin{bmatrix} w_k - w^* \\ w_{k-1} - w^* \end{bmatrix}}_{:= \Delta_k \in \mathbb{R}^{2d}} \implies \Delta_{k+1} = \mathcal{H} \Delta_k$$

Recurring from  $k = 0$  to  $T - 1$ , and taking norm,

$$\|\Delta_T\| = \|\mathcal{H}^T \Delta_0\| \leq \|\mathcal{H}^T\| \left\| \begin{bmatrix} w_0 - w^* \\ w_{-1} - w^* \end{bmatrix} \right\| \quad (\text{By definition of the matrix norm})$$

Define  $w_{-1} = w_0$  and lower-bounding the LHS,

$$\|w_T - w^*\| \leq \sqrt{2} \|\mathcal{H}^T\| \|w_0 - w^*\|$$

Hence, we have reduced the problem to bounding  $\|\mathcal{H}^T\|$ .

# Minimizing strongly-convex quadratics with HB momentum

Recall that for symmetric matrices,  $\|B\|_2 = \rho(B)$ . Unfortunately, this relation is not true for general asymmetric matrices, and  $\|B\| \geq \rho(B)$ .

**Gelfand's Formula:** For a matrix  $B \in \mathbb{R}^{d \times d}$  such that  $\rho(B) := \max_{i \in [d]} |\lambda_i|$ , then there exists a sequence  $\epsilon_k \geq 0$  such that  $\lim_{k \rightarrow \infty} \epsilon_k = 0$  and,

$$\|B^k\| \leq (\rho(B) + \epsilon_k)^k.$$

Using this formula with our bound,

$$\|w_T - w^*\| \leq \sqrt{2} (\rho(\mathcal{H}) + \epsilon_T)^T \|w_0 - w^*\|$$

Hence, we have reduced the problem to bounding  $\rho(\mathcal{H})$ .

# Minimizing strongly-convex quadratics with HB momentum

Similar to the GD case, let  $A = U\Lambda U^\top$  be the eigen-decomposition of  $A$ , then,  $(1 + \beta)I_d - \eta A = USU^\top$  where  $S_{i,i} = 1 + \beta - \eta\lambda_i$ . Hence,

$$\mathcal{H} = \begin{bmatrix} U^\top & 0 \\ 0 & U^\top \end{bmatrix} \underbrace{\begin{bmatrix} (1 + \beta)I_d - \eta\Lambda & -\beta I_d \\ I_d & 0 \end{bmatrix}}_{:=H} \begin{bmatrix} U & 0 \\ 0 & U \end{bmatrix}$$

Since  $U$  is orthonormal,  $\rho(\mathcal{H}) = \rho(H)$ . Hence we have reduced the problem to bounding  $\rho(H)$ .

# Minimizing strongly-convex quadratics with HB momentum

Let  $P$  be a permutation matrix such that:

$$P_{i,j} = \begin{cases} 1 & i \text{ is odd, } j = i \\ 1 & i \text{ is even, } j = d + i \\ 0 & \text{otherwise} \end{cases} \quad B = P H P^\top = \begin{bmatrix} H_1 & 0 & \dots & 0 \\ 0 & H_2 & \dots & 0 \\ \vdots & \ddots & & \\ 0 & & 0 & H_d \end{bmatrix}$$

where,

$$H_i = \begin{bmatrix} (1 + \beta) - \eta \lambda_i & -\beta \\ 1 & 0 \end{bmatrix}$$

Note that  $\rho(H) = \rho(B)$  (a permutation matrix does not change the eigenvalues). Since  $B$  is a block diagonal matrix,  $\rho(B) = \max_i [\rho(H_i)]$ . Hence we have reduced the problem to bounding  $\rho(H_i)$ .

## Minimizing strongly-convex quadratics with HB momentum

For a fixed  $i \in [2d]$ , let us compute the eigenvalues of  $H_i \in \mathbb{R}^{2 \times 2}$  by solving the characteristic polynomial:  $\det(H_i - uI_2) = 0$  w.r.t  $u$ .

$$u^2 - (1 + \beta - \eta\lambda_i)u + \beta = 0 \implies u = \frac{1}{2} \left[ (1 + \beta - \eta\lambda_i) \pm \sqrt{(1 + \beta - \eta\lambda_i)^2 - 4\beta} \right]$$

Let us set  $\beta$  such that,  $(1 + \beta - \eta\lambda_i)^2 \leq 4\beta$ . This ensures that the roots to the above equation are complex conjugates. Hence,

$$1 + \beta - \eta\lambda_i \geq -2\sqrt{\beta} \implies (\sqrt{\beta} + 1) \geq \sqrt{\eta\lambda_i} \implies \beta \geq (1 - \sqrt{\eta\lambda_i})^2$$

If we ensure that  $\beta \geq (1 - \sqrt{\eta\lambda_i})^2$

$$\begin{aligned} u &= \frac{1}{2} \left[ (1 + \beta - \eta\lambda_i) \pm i\sqrt{4\beta - (1 + \beta - \eta\lambda_i)^2} \right] \\ \implies |u|^2 &= \frac{1}{4} \left[ (1 + \beta - \eta\lambda_i)^2 + 4\beta - (1 + \beta - \eta\lambda_i)^2 \right] = \beta \implies |u| = \sqrt{\beta}. \end{aligned}$$

Hence, if  $\beta \geq (1 - \sqrt{\eta\lambda_i})^2$ ,  $\rho(H_i) = \sqrt{\beta}$  and  $\rho(B) = \max_i [\rho(H_i)] = \sqrt{\beta}$ .



# Minimizing strongly-convex quadratics with HB momentum

Using the result from the previous slide, if we ensure that for all  $i$ ,  $\beta \geq (1 - \sqrt{\eta\lambda_i})^2$ , then,  $\rho(B) = \sqrt{\beta}$ . Hence, we want that,

$$\beta = \max_i \{(1 - \sqrt{\eta\lambda_i})^2\} \leq \max_{\lambda \in [\mu, L]} \{(1 - \sqrt{\eta\lambda})^2\} = \max\{(1 - \sqrt{\eta\mu})^2, (1 - \sqrt{\eta L})^2\}$$

Similar to GD, we equate the two terms in the max,

$$1 + \eta\mu - 2\sqrt{\eta\mu} = 1 + \eta L - 2\sqrt{\eta L} \implies \eta = \frac{4}{(\sqrt{L} + \sqrt{\mu})^2}.$$

With this value of  $\eta$ ,  $\rho(\mathcal{H}) = \rho(H) = \rho(B) \leq \sqrt{\beta} = \sqrt{\left(1 - \frac{2\sqrt{\mu}}{(\sqrt{L} + \sqrt{\mu})}\right)^2} = \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}$ .

Putting everything together,

$$\|w_T - w^*\| \leq \sqrt{2} \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} + \epsilon_T \right)^T \|w_0 - w^*\|$$

Questions?

# Gradient Descent and Newton's method

For  $L$ -smooth,  $\mu$ -strongly convex functions,

- Gradient Descent (GD) results in an  $O(\exp(-T/\kappa))$  rate.
- Nesterov acceleration can speed up the convergence and results in an  $\Theta(\exp(-T/\sqrt{\kappa}))$  rate.
- Lower-Bound: Without additional assumptions, no first-order algorithm (one that only relies on gradient information) can attain a dimension-free rate faster than  $\Omega(\exp(-T/\sqrt{\kappa}))$ .

Next, we will use second-order (Hessian) information to minimize twice differentiable,  $L$ -smooth and  $\mu$ -strongly convex functions and get faster rates under additional assumptions.

# Gradient Descent and Newton's method

Recall the GD update:  $w_{k+1} = w_k - \eta \nabla f(w_k)$ . This can also be written as:

$$w_{k+1} = \arg \min_w \left[ \underbrace{f(w_k) + \langle \nabla f(w_k), w_k - w \rangle}_{\text{First-order Taylor series approximation}} + \underbrace{\frac{1}{2\eta} \|w_k - w\|^2}_{\text{Stay close to } w_k} \right]$$

i.e., approximate the function by a first-order Taylor series expansion, and minimize it while staying close (in the Euclidean norm) to the current point.

If  $f$  is twice-differentiable, and we approximate it by a second-order Taylor series expansion,

$$w_{k+1} = \arg \min_w \left[ \underbrace{f(w_k) + \langle \nabla f(w_k), w - w_k \rangle + \frac{1}{2} (w - w_k)^\top \nabla^2 f(w_k) (w - w_k)}_{\text{Second-order Taylor series approximation}} \right]$$
$$\implies w_{k+1} = w_k - [\nabla^2 f(w_k)]^{-1} [\nabla f(w_k)] \quad (\text{Newton Update})$$

## Digression - Preconditioned Gradient Descent

Recall that GD achieves an  $O\left(\kappa \log\left(\frac{1}{\epsilon}\right)\right)$  convergence rate, and the condition number  $\kappa \geq 1$  is the measure of problem difficulty.

**Idea:** Reparameterize the space so that the minimum function value remains the same, but condition number in the reparameterized space is smaller enabling GD to converge faster.

*Example:*  $\min_{w \in \mathbb{R}^2} f(w) = \frac{1}{2} w^\top A w$  where  $A = \begin{bmatrix} L & 0 \\ 0 & \mu \end{bmatrix}$ . For the above problem,  $w^* = 0$ ,  $f(w^*) = 0$  and  $\kappa = \frac{L}{\mu}$ .

Let us choose a **preconditioning matrix**  $Q \in \mathbb{R}^{2 \times 2}$  such that  $w = Qv$ , and write the reparameterized function  $g(v) := \frac{1}{2} [Qv]^\top A [Qv] = \frac{1}{2} v^\top Q^\top A Q v$ .

If we choose  $Q = \begin{bmatrix} \frac{1}{\sqrt{L}} & 0 \\ 0 & \frac{1}{\sqrt{\mu}} \end{bmatrix}$ ,  $Q^\top A Q = I$ ,  $g(v) = \frac{1}{2} v^\top v$ . Clearly,  $v^* = 0$  and  $g(v^*) = 0$  and  $w^* = Qv^* = 0$ . For this problem,  $\kappa = 1$  making it easier to solve using GD.

## Digression - Preconditioned Gradient Descent

Formalizing the intuition on the previous slide, define a positive definite, symmetric matrix  $Q \in \mathbb{R}^{d \times d}$  such that  $w = Qv$  and hence,  $v = Q^{-1}w$ . Define  $g(v) := f(Qv)$ .

**Q:** If  $w^* = \arg \min_w f(w)$  and  $v^* = \arg \min_v g(v)$ , is  $f(w^*) = g(v^*)$ ?

Computing the gradient of  $g(v)$ ,  $\nabla g(v) = Q^T \nabla f(Qv)$ . Running GD on  $g(v)$ , we get that,

$$\begin{aligned} v_{k+1} &= v_k - \eta \nabla g(v_k) = v_k - \eta [Q^T \nabla f(Qv_k)] = v_k - \eta [Q^T \nabla f(w_k)] \\ \implies Q^{-1}w_{k+1} &= Q^{-1}w_k - \eta [Q \nabla f(w_k)] \implies w_{k+1} = w_k - \eta [QQ^T \nabla f(w_k)] \end{aligned}$$

Define a positive definite, symmetric  $P$  such that  $P = QQ^T$ . Since  $Q$  is symmetric,  $Q = P^{\frac{1}{2}}$ . Hence, for  $w = P^{\frac{1}{2}}v$ ,

$$w_{k+1} = w_k - \eta [P \nabla f(w_k)] \quad (\text{Preconditioned GD})$$

i.e., compute the gradient, “precondition” it by matrix  $P$  and then do the GD step.

# Digression - Preconditioned Gradient Descent

Equivalent formulations of preconditioned gradient descent to minimize  $f(w)$ ,

- Reparameterizing the space using a positive definite, symmetric matrix  $P^{\frac{1}{2}}$  such that  $v = P^{-\frac{1}{2}}w$  and using GD to minimize  $g(v) := f(P^{\frac{1}{2}}v)$ .
- Use GD with the preconditioned gradient  $P\nabla f(w)$ .
- The preconditioned GD update at iteration  $k$  can be written as:

$$w_{k+1} = \arg \min \left[ \underbrace{f(w_k) + \langle \nabla f(w_k), w_k - w \rangle}_{\text{First-order Taylor series approximation}} + \underbrace{\frac{1}{2\eta} \|w_k - w\|_{P^{-1}}^2}_{\text{Stay close to } w_k} \right]$$

i.e., approximate the function by a first-order Taylor series expansion, and minimize it while staying close (in the norm induced by matrix  $P^{-1}$ ) to the current point.

We can also use a different preconditioner at every iteration, i.e.

$$w_{k+1} = w_k - \eta[P_k \nabla f(w_k)]$$

## Digression - Preconditioned Gradient Descent

- But what is the “best”  $P_k$  around a specific iterate for a specific problem? For this, consider the Hessian of  $g(v) = f(P^{\frac{1}{2}}v)$  and let us choose  $P$  such that  $\kappa = 1$ .

Recall that  $\nabla g(v) = P^{\frac{1}{2}} \nabla f(P^{\frac{1}{2}}v)$  and hence,  $\nabla^2 g(v) = P^{\frac{1}{2}} [\nabla^2 f(P^{\frac{1}{2}}v)] (P^{\frac{1}{2}})^{\top}$ . If  $P = [\nabla^2 f(P^{\frac{1}{2}}v)]^{-1} = [\nabla^2 f(w)]^{-1}$ , then,

$$\nabla^2 g(v) = [\nabla^2 f(P^{\frac{1}{2}}v)]^{-\frac{1}{2}} [\nabla^2 f(P^{\frac{1}{2}}v)] [\nabla^2 f(P^{\frac{1}{2}}v)]^{-\frac{1}{2}} = I_d$$

Around iterate  $w_k$ , define  $P_k := [\nabla^2 f(w_k)]^{-1}$  and using the equivalence to preconditioned gradient descent, the resulting update can be written as:

$$w_{k+1} = w_k - \eta [\nabla^2 f(w_k)]^{-1} \nabla f(w_k)$$

If  $\eta = 1$ , we have recovered the Newton method! Hence, the Newton method can be thought of as finding the best preconditioner (one that minimizes the condition number) at every iteration of preconditioned GD.



# Newton Method

Using the equivalence to preconditioned GD, the Newton method is also equivalent to:

$$w_{k+1} = \arg \min \left[ \underbrace{f(w_k) + \langle \nabla f(w_k), w_k - w \rangle}_{\text{First-order Taylor series approximation}} + \underbrace{\frac{1}{2\eta} \|w_k - w\|_{\nabla^2 f(w_k)}^2}_{\text{Stay close to } w_k} \right]$$

i.e., approximate the function by a first-order Taylor series expansion, and minimize it while staying close (in the “local norm” induced by the Hessian at  $w_k$ ) to the current point.

*Example:* Consider solving  $w^* = \arg \min f(w) := \frac{1}{2} w^\top A w - b w + c$ . We know that  $\nabla f(w) = A w - b = A(w - w^*)$  and  $\nabla^2 f(w) = A$ . Starting from point  $w_0$ , consider the Newton update with  $\eta = 1$ ,

$$w_1 = w_0 - [A^{-1}] A(w_0 - w^*) = w^*$$

i.e. the Newton method can minimize quadratics in one step. In this case,  $P_k = P = A^{-1}$  and hence,  $g(v) = f(A^{-\frac{1}{2}} v) = \frac{1}{2} [A^{-\frac{1}{2}} v]^\top A [A^{-\frac{1}{2}} v] - b [A^{-\frac{1}{2}} v] + c = \frac{1}{2} v^\top v - b A^{-\frac{1}{2}} v + c$ . Computing the Hessian of  $g(v)$ ,  $\nabla^2 g(v) = I_d$  which has  $\kappa = 1$ .

Questions?

We have seen that for quadratics, the Newton method converges to the minimizer in one step.

- Let us analyze the convergence of Newton for general  $L$ -smooth,  $\mu$ -strongly convex functions. For this, we will consider two phases for the update:

$$w_{k+1} = w_k - \eta_k [\nabla^2 f(w_k)]^{-1} \nabla f(w_k),$$

**Phase 1 (Damped Newton):** For some  $\alpha$  to be chosen later, if  $\|\nabla f(w_k)\|^2 > \alpha$  (“far” from the solution), use the Newton method with the step-size  $\eta_k$  set according to the Back-tracking Armijo line-search.

**Phase 2 (Pure Newton):** If  $\|\nabla f(w_k)\|^2 \leq \alpha$  (“close” to the solution), use the Newton method with step-size equal to 1.

## Newton Method - Phase 2

Let us first analyze the convergence rate for Phase 2. For this, we will need an additional assumption that the Hessian is Lipschitz continuous with constant  $M > 0$ :

$$\|\nabla^2 f(w) - \nabla^2 f(v)\| \leq M \|w - v\|.$$

**Claim:** In Phase 2 of the Newton method, the iterates satisfy the following inequality,

$$\|w_{k+1} - w^*\| \leq \frac{M}{2\mu} \|w_k - w^*\|^2$$

**Proof:**

$$\begin{aligned} w_{k+1} - w^* &= w_k - w^* - [\nabla^2 f(w_k)]^{-1} \nabla f(w_k) \quad (\text{Newton update with step-size 1.}) \\ &= [\nabla^2 f(w_k)]^{-1} [[\nabla^2 f(w_k)](w_k - w^*) - \nabla f(w_k)] \end{aligned}$$

$$\implies \|w_{k+1} - w^*\| = \|[\nabla^2 f(w_k)]^{-1} [[\nabla^2 f(w_k)](w_k - w^*) - \nabla f(w_k)]\|$$

$$\begin{aligned} \implies \|w_{k+1} - w^*\| &\leq \|[\nabla^2 f(w_k)]^{-1}\| \|[\nabla^2 f(w_k)](w_k - w^*) - \nabla f(w_k)\| \\ &\quad (\text{By definition of the matrix norm}) \end{aligned}$$

## Newton Method - Phase 2

Recall that  $\|w_{k+1} - w^*\| \leq \|[\nabla^2 f(w_k)]^{-1}\| \|\nabla^2 f(w_k)(w_k - w^*) - \nabla f(w_k)\|$ .

$$\begin{aligned}\|w_{k+1} - w^*\| &\leq \frac{1}{\mu} \|\nabla^2 f(w_k)(w_k - w^*) - \nabla f(w_k)\| \quad (\text{Since } \nabla^2 f(w) \succeq \mu I_d) \\ \implies \|w_{k+1} - w^*\| &\leq \frac{1}{\mu} \|\nabla^2 f(w_k)(w_k - w^*) + \nabla f(w^*) - \nabla f(w_k)\| \quad (1)\end{aligned}$$

Now let us bound  $\nabla f(w^*) - \nabla f(w_k)$ . By the fundamental theorem of calculus, for all  $x, y$ ,  $f(y) = f(x) + \int_{t=0}^1 [\nabla f(t y + (1-t)x)] (y-x) dt$ . This theorem also holds for the vector-valued gradient function,

$$\nabla f(y) = \nabla f(x) + \int_{t=0}^1 [\nabla^2 f(t y + (1-t)x)] (y-x) dt$$

Using the above statement with  $x = w^*$  and  $y = w_k$ ,

$$\implies \nabla f(w_k) - \nabla f(w^*) = \int_{t=0}^1 [\nabla^2 f(t w_k + (1-t) w^*)] (w_k - w^*) dt \quad (2)$$

## Newton Method - Phase 2

Combining eqs. (1) and (2),

$$\begin{aligned} & \|w_{k+1} - w^*\| \\ & \leq \frac{1}{\mu} \left\| [\nabla^2 f(w_k)](w_k - w^*) + \nabla f(w^*) - \nabla f(w_k) \right\| \\ & = \frac{1}{\mu} \left\| \left[ [\nabla^2 f(w_k)](w_k - w^*) - \int_{t=0}^1 [\nabla^2 f(t w_k + (1-t) w^*)] (w_k - w^*) dt \right] \right\| \\ & = \frac{1}{\mu} \left\| \left[ \int_{t=0}^1 [\nabla^2 f(w_k)](w_k - w^*) dt - \int_{t=0}^1 [\nabla^2 f(t w_k + (1-t) w^*)] (w_k - w^*) dt \right] \right\| \\ & = \frac{1}{\mu} \left\| \int_{t=0}^1 [\nabla^2 f(w_k) - \nabla^2 f(t w_k + (1-t) w^*)] (w_k - w^*) dt \right\| \\ & \leq \frac{1}{\mu} \int_{t=0}^1 \left\| [\nabla^2 f(w_k) - \nabla^2 f(t w_k + (1-t) w^*)] (w_k - w^*) \right\| dt \quad (\text{Jensen's inequality}) \\ & \leq \frac{1}{\mu} \int_{t=0}^1 \left\| \nabla^2 f(w_k) - \nabla^2 f(t w_k + (1-t) w^*) \right\| \|w_k - w^*\| dt \quad (\text{Definition of matrix norm}) \end{aligned}$$

## Newton Method - Phase 2

From the previous slide,

$$\|w_{k+1} - w^*\| \leq \frac{1}{\mu} \int_{t=0}^1 \|\nabla^2 f(w_k) - \nabla^2 f(t w_k + (1-t) w^*)\| \|w_k - w^*\| dt$$

Since the Hessian is  $M$ -Lipschitz,

$$\begin{aligned} &\leq \frac{1}{\mu} \int_{t=0}^1 M \|w_k - t w_k - (1-t) w^*\| \|w_k - w^*\| dt \\ &= \frac{M}{\mu} \|w_k - w^*\| \int_{t=0}^1 \|(1-t)(w_k - w^*)\| dt \\ &= \frac{M}{\mu} \|w_k - w^*\|^2 \int_{t=0}^1 (1-t) dt \\ \implies \|w_{k+1} - w^*\| &\leq \frac{M}{2\mu} \|w_k - w^*\|^2 \end{aligned}$$

## Newton Method - Phase 2

Recall that for Phase 2 of the Newton method,  $\|w_{k+1} - w^*\| \leq c \|w_k - w^*\|^2$  where  $c := \frac{M}{2\mu}$ .

**Claim:** If in Phase 2,  $\|w_0 - w^*\| \leq \frac{1}{2c} = \frac{\mu}{M}$ , then after  $T$  iterations of the Pure Newton update,  $\|w_T - w^*\| \leq \left(\frac{1}{2}\right)^{2^T} \frac{1}{c} = \left(\frac{1}{2}\right)^{2^T} \frac{2\mu}{M}$ .

**Proof:** Let us prove it by induction.

**Base-case:** For  $T = 0$ ,  $\|w_T - w^*\| \leq \frac{\mu}{M}$  which is true by our assumption.

**Inductive hypothesis:** If the statement is true for iteration  $k$ , then  $\|w_k - w^*\| \leq \left(\frac{1}{2}\right)^{2^k} \frac{1}{c}$ .

$$\|w_{k+1} - w^*\| \leq c \|w_k - w^*\|^2 \leq c \left( \left(\frac{1}{2}\right)^{2^k} \frac{1}{c} \right)^2 = \frac{1}{c} \left(\frac{1}{2}\right)^{2^{k+1}},$$

which completes the induction. Hence,  $\|w_T - w^*\| \leq \left(\frac{1}{2}\right)^{2^T} \frac{2\mu}{M}$ . For  $\|w_T - w^*\| \leq \epsilon$ , we need  $T$  such that,

$$\left(\frac{1}{2}\right)^{2^T} \frac{2\mu}{M} \leq \epsilon \implies T \geq \frac{1}{\log(2)} \log \left( \frac{\log(2\mu/M\epsilon)}{\log(2)} \right)$$



## Newton Method - Phase 2

- From the previous slide, we can conclude that Phase 2 of the Newton method requires  $O(\log(\log(1/\epsilon)))$  iterations to achieve an  $\epsilon$  sub-optimality.
- This rate of convergence is often referred to as **quadratic** or **super-linear** convergence. Note that there is no dependence on  $\kappa$  and the dependence on  $\frac{\mu}{M}$  is in the log log.
- But the bound is true only if  $\|w_0 - w^*\| \leq \frac{\mu}{M}$  i.e. we enter Phase 2 only when we are “close enough” to the solution. This is referred to as **local convergence**. Hence, the Newton method has super-linear local convergence.
- Algorithmically, since we do not know  $w^*$ , we do not know when to start Phase 2 of the algorithm. By strong-convexity,

$$\|\nabla f(x) - \nabla f(y)\| \geq \mu \|x - y\| \implies \|w_0 - w^*\| \leq \frac{1}{\mu} \|\nabla f(w_0)\|$$

Hence, in order to ensure that  $\|w_0 - w^*\| \leq \frac{\mu}{M}$ , it suffices to guarantee that  $\|\nabla f(w_0)\|^2 \leq \alpha := \frac{\mu^4}{M^2}$ . This can be checked algorithmically.

Questions?

# Newton Method

**Theorem:** If  $\|\nabla f(w)\|^2 \leq \alpha = \frac{\mu^4}{M^2}$ , the algorithm switches to Phase 2 for  $T$  iterations of the pure Newton step and ensures that  $\|w_T - w^*\| \leq \left(\frac{1}{2}\right)^{2^T} \frac{2\mu}{M}$ .

- In order to prove global convergence for the Newton method i.e. starting from any initialization, we need to prove that Phase 1 of the Newton step can result in an iterate  $w$  such that  $\|\nabla f(w)\|^2 \leq \alpha$  and we can switch to Phase 2.
- Recall that for Phase 1, we will use the Backtracking Armijo line-search. For a prospective step-size  $\tilde{\eta}_k$ , check the (more general) Armijo condition,

$$f(w_k - \tilde{\eta}_k d_k) \leq f(w_k) - c \tilde{\eta}_k \underbrace{\langle \nabla f(w_k), d_k \rangle}_{\text{Newton decrement}}$$

where  $c \in (0, 1)$  is a hyper-parameter and  $d_k = [\nabla^2 f(w_k)]^{-1} \nabla f(w_k)$  is the Newton direction. If  $\tilde{\eta}_k$  satisfies the above condition, use the Newton update with  $\eta_k = \tilde{\eta}_k$ .

**Q:** Why does the Newton direction make an acute angle with the gradient direction?

# Newton Method - Phase 1

- Using a similar proof as the standard Back-tracking Armijo line-search, we can show that the step-size returned by the back-tracking procedure at iteration  $k$  is lower-bounded as:

$$\eta_k \geq \min \left\{ \frac{2\mu(1-c)}{L}, \eta_{\max} \right\} \text{ (Need to prove this in Assignment 2).}$$

- At iteration  $k$ ,  $\eta_k$  is the step-size returned by the Back-tracking Armijo line-search and satisfies the general Armijo condition. Hence,

$$\begin{aligned} f(w_k - \eta_k d_k) - f^* &\leq [f(w_k) - f^*] - c \eta_k \langle \nabla f(w_k), d_k \rangle \\ \implies f(w_{k+1}) - f^* &\leq [f(w_k) - f^*] - c \eta_k \langle \nabla f(w_k), [\nabla^2 f(w_k)]^{-1} \nabla f(w_k) \rangle \end{aligned}$$

Since  $\nabla^2 f(w_k)$  is P.S.D,  $\langle \nabla f(w_k), [\nabla^2 f(w_k)]^{-1} \nabla f(w_k) \rangle \geq 0$  and we need to lower-bound it,

$$\begin{aligned} \langle \nabla f(w_k), [\nabla^2 f(w_k)]^{-1} \nabla f(w_k) \rangle &\geq \lambda_{\min}[\nabla^2 f(w_k)]^{-1} \|\nabla f(w_k)\|^2 \\ \implies f(w_{k+1}) - f^* &\leq [f(w_k) - f^*] - c \eta_k \lambda_{\min}[\nabla^2 f(w_k)]^{-1} \|\nabla f(w_k)\|^2 \\ f(w_{k+1}) - f^* &\leq [f(w_k) - f^*] - \frac{c \eta_k}{L} \|\nabla f(w_k)\|^2 \\ &\quad \text{(Since } \lambda_{\min}[\nabla^2 f(w_k)]^{-1} = \frac{1}{\lambda_{\max}[\nabla^2 f(w_k)]} = \frac{1}{L} \text{)} \end{aligned}$$

## Newton Method - Phase 1

Recall that  $f(w_{k+1}) - f^* \leq [f(w_k) - f^*] - c \eta_k / L \|\nabla f(w_k)\|^2$ .

$$f(w_{k+1}) - f^* \leq [f(w_k) - f^*] - \frac{c \min \left\{ \frac{2\mu(1-c)}{L}, \eta_{\max} \right\}}{L} \|\nabla f(w_k)\|^2 \quad (\text{Lower-bound on } \eta_k)$$

$$\leq [f(w_k) - f^*] - \frac{\min \left\{ \frac{\mu}{2L}, \frac{\eta_{\max}}{2} \right\}}{L} \|\nabla f(w_k)\|^2 \quad (\text{Setting } c = 1/2)$$

$$\leq \left( 1 - \frac{\mu \min \left\{ \frac{\mu}{L}, \eta_{\max} \right\}}{L} \right) [f(w_k) - f^*] \quad (\|\nabla f(w_k)\|^2 \geq 2\mu[f(w_k) - f^*])$$

$$\implies f(w_{k+1}) - f^* \leq \left( 1 - \frac{\mu^2 \min \{1, \kappa \eta_{\max}\}}{L^2} \right) [f(w_k) - f^*]$$

Recurring from  $k = 0$  to  $\tau - 1$  and setting  $\eta_{\max} = 1$

$$f(w_\tau) - f^* \leq \left( 1 - \frac{1}{\kappa^2} \right)^\tau [f(w_0) - f^*] \leq \exp \left( \frac{-\tau}{\kappa^2} \right) [f(w_0) - f^*]$$

# Newton Method

Recall that  $f(w_\tau) - f^* \leq \exp\left(\frac{-\tau}{\kappa^2}\right) [f(w_0) - f^*]$ . Phase 1 terminates when  $\|\nabla f(w_\tau)\|^2 = \alpha$ . Using  $L$ -smoothness,  $\|\nabla f(w_\tau)\|^2 \leq 2L [f(w_\tau) - f^*]$ . To terminate Phase 1, we want

$$\begin{aligned} 2L [f(w_\tau) - f^*] &= 2L \exp\left(\frac{-\tau}{\kappa^2}\right) [f(w_0) - f^*] = \alpha \\ \implies \tau &= \kappa^2 \log\left(\frac{2L M^2 [f(w_0) - f^*]}{\mu^4}\right) \end{aligned} \quad (\text{Since } \alpha = \frac{\mu^4}{M^2})$$

- Hence, iterations required for global convergence to an  $\epsilon$  sub-optimality is,

$$\underbrace{\kappa^2 \log\left(\frac{2L M^2 [f(w_0) - f^*]}{\mu^4}\right)}_{\text{Phase 1}} + \underbrace{\frac{1}{\log(2)} \log\left(\frac{\log(2^{\mu/M\epsilon})}{\log(2)}\right)}_{\text{Phase 2}} = O(\kappa^2 + \log(\log(1/\epsilon)))$$

- Recall that GD requires  $O(\kappa \log(1/\epsilon))$  iterations. If we do a matrix inversion in every iteration, cost of each iteration is  $O(d^3)$ . Since computing gradients is linear in  $d$ , the cost of each GD iteration is  $O(d)$ . Comparing computational complexity:

Gradient Descent:  $O(d\kappa \log(1/\epsilon))$     Newton Method:  $O((d^3\kappa^2 + d^3 \log(\log(1/\epsilon))))$

- Newton method is more efficient than GD for small  $d$  (low-dimension) and small  $\epsilon$  (high precision).

Questions?