ECEN 434: Optimization for ECEN

Lecture 10: Descent Methods for Unconstrained Problems

Tie Liu

Unconstrained minimization problems

- Consider the unconstrained optimization problem:

$$\text{minimize}_{\boldsymbol{x} \in \mathbb{X}} \quad f(\boldsymbol{x})$$

where f is convex and twice continuously differentiable and $\mathbb{X}$ is a nonempty, open set

Unconstrained minimization problems

- Consider the unconstrained optimization problem:

$$\text{minimize}_{\boldsymbol{x} \in \mathbb{X}} \quad f(\boldsymbol{x})$$

  where f is convex and twice continuously differentiable and $\mathbb{X}$ is a nonempty, open set

- We will assume that the the minimum value of f is attainable, i.e., there exists an optimal point $\boldsymbol{x}^*$

Unconstrained minimization problems

- Consider the unconstrained optimization problem:

$$\text{minimize}_{\boldsymbol{x} \in \mathbb{X}} \quad f(\boldsymbol{x})$$

  where f is convex and twice continuously differentiable and $\mathbb{X}$ is a nonempty, open set

- We will assume that the the minimum value of f is attainable, i.e., there exists an optimal point $\boldsymbol{x}^*$

- Since f is convex and differentiable, a necessary and sufficient condition for a point $\boldsymbol{x}^*$ to be optimal is

$$\nabla f(\boldsymbol{x}^*) = \mathbf{0}$$

  Thus, solving the unconstrained minimization problem is the same as finding a solution to the above system of equations

- In a few special cases, we can find a solution to the unconstrained minimization problems by analytically solving the optimality equations, but usually the problem must be solved by an iterative algorithm

- In a few special cases, we can find a solution to the unconstrained minimization problems by analytically solving the optimality equations, but usually the problem must be solved by an iterative algorithm
- By this we mean an algorithm that computes a sequence of points $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots \in \mathbb{X}$ with $f(\boldsymbol{x}^{(k)}) \to p^*$ as $k \to \infty$. Such a sequence of points is called a minimizing sequence for the minimization problem

- In a few special cases, we can find a solution to the unconstrained minimization problems by analytically solving the optimality equations, but usually the problem must be solved by an iterative algorithm
- By this we mean an algorithm that computes a sequence of points $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots \in \mathbb{X}$ with $f(\boldsymbol{x}^{(k)}) \to p^*$ as $k \to \infty$. Such a sequence of points is called a minimizing sequence for the minimization problem
- The algorithm is terminated when $f(\boldsymbol{x}^{(k)}) - p^* \leq \epsilon$, where $\epsilon > 0$ is some specified tolerance

Descent methods

- We focus on the so-called descent methods, which start the algorithm by choosing a suitable starting point $\boldsymbol{x}^{(0)} \in \mathbb{X}$

Descent methods

- We focus on the so-called descent methods, which start the algorithm by choosing a suitable starting point $\boldsymbol{x}^{(0)} \in \mathbb{X}$
- The algorithm then produces a minimizing sequence through the following iterative update:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + t^{(k)} \Delta \boldsymbol{x}^{(k)}$$

Descent methods

- We focus on the so-called descent methods, which start the algorithm by choosing a suitable starting point $\boldsymbol{x}^{(0)} \in \mathbb{X}$
- The algorithm then produces a minimizing sequence through the following iterative update:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + t^{(k)} \Delta \boldsymbol{x}^{(k)}$$

- Here, $k = 0, 1, \ldots$ denotes the iteration number, $\boldsymbol{x}^{(k)}$ is called the step or search direction (even though it need not have unit norm), and the scalar $t^{(k)} \geq 0$ is called the step size or step length at iteration k

Descent methods

- We focus on the so-called descent methods, which start the algorithm by choosing a suitable starting point $\boldsymbol{x}^{(0)} \in \mathbb{X}$
- The algorithm then produces a minimizing sequence through the following iterative update:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + t^{(k)} \Delta \boldsymbol{x}^{(k)}$$

- Here, $k = 0, 1, \ldots$ denotes the iteration number, $\boldsymbol{x}^{(k)}$ is called the step or search direction (even though it need not have unit norm), and the scalar $t^{(k)} \geq 0$ is called the step size or step length at iteration k
- The algorithm terminates when $\|\nabla f(\boldsymbol{x}^{(k)})\| \leq \eta$, where η is small and positive

Descent methods

- We focus on the so-called descent methods, which start the algorithm by choosing a suitable starting point $\boldsymbol{x}^{(0)} \in \mathbb{X}$
- The algorithm then produces a minimizing sequence through the following iterative update:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + t^{(k)} \Delta \boldsymbol{x}^{(k)}$$

- Here, $k = 0, 1, \ldots$ denotes the iteration number, $\boldsymbol{x}^{(k)}$ is called the step or search direction (even though it need not have unit norm), and the scalar $t^{(k)} \geq 0$ is called the step size or step length at iteration k
- The algorithm terminates when $\|\nabla f(\boldsymbol{x}^{(k)})\| \leq \eta$, where η is small and positive
- The above iterative procedure is called a descent method if

$$f(\boldsymbol{x}^{(k+1)}) < f(\boldsymbol{x}^{(k)})$$

except when $\boldsymbol{x}^{(k)}$ is already optimal

Exact line search

- One method for determining the step sizes is exact line search, in which $t^{(k)}$ is chosen to minimize f along the ray $\{\boldsymbol{x}^{(k)} + t\Delta\boldsymbol{x}^{(k)} : t \geq 0\}$:

$$t^{(k)} = \arg\min_{t \geq 0} f(\boldsymbol{x}^{(k)} + t\Delta\boldsymbol{x}^{(k)})$$

Exact line search

- One method for determining the step sizes is exact line search, in which $t^{(k)}$ is chosen to minimize f along the ray $\{\boldsymbol{x}^{(k)} + t\Delta\boldsymbol{x}^{(k)} : t \geq 0\}$:

$$t^{(k)} = \arg\min_{t \geq 0} f(\boldsymbol{x}^{(k)} + t\Delta\boldsymbol{x}^{(k)})$$

- An exact line search is used when the cost of the minimization problem with one variable is low compared to the cost of computing the search direction itself. In some special cases, the minimizer along the ray can be found analytically and in others it can be computed efficiently

Gradient descent

- By convexity we have

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}^{(k)}) + \nabla f(\boldsymbol{x}^{(k)})^t (\boldsymbol{y} - \boldsymbol{x}^{(k)})$$

Gradient descent

- By convexity we have

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}^{(k)}) + \nabla f(\boldsymbol{x}^{(k)})^t(\boldsymbol{y} - \boldsymbol{x}^{(k)})$$

- Therefore, $\nabla f(\boldsymbol{x}^{(k)})^t(\boldsymbol{y} - \boldsymbol{x}^{(k)}) \geq 0$ implies $f(\boldsymbol{y}) \geq f(\boldsymbol{x}^{(k)})$

Gradient descent

- By convexity we have

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}^{(k)}) + \nabla f(\boldsymbol{x}^{(k)})^t(\boldsymbol{y} - \boldsymbol{x}^{(k)})$$

- Therefore, $\nabla f(\boldsymbol{x}^{(k)})^t(\boldsymbol{y} - \boldsymbol{x}^{(k)}) \geq 0$ implies $f(\boldsymbol{y}) \geq f(\boldsymbol{x}^{(k)})$
- Consequently, the search direction in a descent method must satisfy

$$\nabla f(\boldsymbol{x}^{(k)})^t \Delta \boldsymbol{x}^{(k)} < 0$$

  i.e., it must make an acute angle with the negative gradient

Gradient descent

- By convexity we have

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}^{(k)}) + \nabla f(\boldsymbol{x}^{(k)})^t(\boldsymbol{y} - \boldsymbol{x}^{(k)})$$

- Therefore, $\nabla f(\boldsymbol{x}^{(k)})^t(\boldsymbol{y} - \boldsymbol{x}^{(k)}) \geq 0$ implies $f(\boldsymbol{y}) \geq f(\boldsymbol{x}^{(k)})$
- Consequently, the search direction in a descent method must satisfy

$$\nabla f(\boldsymbol{x}^{(k)})^t \Delta \boldsymbol{x}^{(k)} < 0$$

  i.e., it must make an acute angle with the negative gradient

- A natural choice for the search direction is the negative gradient

$$\Delta \boldsymbol{x}^{(k)} = -\nabla f(\boldsymbol{x}^{(k)})$$

  The resulting algorithm is called the gradient algorithm or gradient descent method

Gradient descent for a quadratic function

- Consider the problem of minimizing

$$f(x_1, x_2) = \frac{1}{2}\left(\sigma_1 x_1^2 + \sigma_2 x_2^2\right)$$

over $\mathbb{R}^2$, where $\sigma_1 \geq \sigma_2 > 0$ are two positive constants

Gradient descent for a quadratic function

- Consider the problem of minimizing

$$f(x_1, x_2) = \frac{1}{2}\left(\sigma_1 x_1^2 + \sigma_2 x_2^2\right)$$

  over $\mathbb{R}^2$, where $\sigma_1 \geq \sigma_2 > 0$ are two positive constants

- Apparently, $(x_1^*, x_2^*) = (0, 0)$ is the unique minimum point and the optimal value $p^* = 0$

Gradient descent for a quadratic function

- Consider the problem of minimizing

$$f(x_1, x_2) = \frac{1}{2}\left(\sigma_1 x_1^2 + \sigma_2 x_2^2\right)$$

  over $\mathbb{R}^2$, where $\sigma_1 \geq \sigma_2 > 0$ are two positive constants

- Apparently, $(x_1^*, x_2^*) = (0, 0)$ is the unique minimum point and the optimal value $p^* = 0$

- Consider gradient descent with exact line search:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - t\nabla f(\boldsymbol{x}^{(k)}) = \begin{bmatrix} (1-\sigma_1 t)x_1^{(k)} \\ (1-\sigma_2 t)x_2^{(k)} \end{bmatrix}$$

- Note that

$$f(\boldsymbol{x}^{(k+1)}) = \frac{1}{2}\left[\sigma_1(1-\sigma_1 t)^2(x_1^{(k)})^2 + \sigma_2(1-\sigma_2 t)^2(x_2^{(k)})^2\right]$$

  is a quadratic function of t

- Note that

$$f(\boldsymbol{x}^{(k+1)}) = \frac{1}{2}\left[\sigma_1(1-\sigma_1 t)^2 (x_1^{(k)})^2 + \sigma_2(1-\sigma_2 t)^2 (x_2^{(k)})^2\right]$$

  is a quadratic function of t

- Thus, an exact line search yields:

$$t^{(k)} = \frac{\sigma_1^2 (x_1^{(k)})^2 + \sigma_2^2 (x_2^{(k)})^2}{\sigma_1^3 (x_1^{(k)})^2 + \sigma_2^3 (x_2^{(k)})^2}$$

  and hence

$$\boldsymbol{x}^{(k+1)} = \frac{x_1^{(k)} x_2^{(k)} (\sigma_1 - \sigma_2)}{\sigma_1^3 (x_1^{(k)})^2 + \sigma_2^3 (x_2^{(k)})^2} \begin{bmatrix} -\sigma_2^2 x_2^{(k)} \\ \sigma_1^2 x_1^{(k)} \end{bmatrix}$$

- For $k = 0$, we have

$$\boldsymbol{x}^{(1)} = \frac{x_1^{(0)} x_2^{(0)} (\sigma_1 - \sigma_2)}{\sigma_1^3 (x_1^{(0)})^2 + \sigma_2^3 (x_2^{(0)})^2} \begin{bmatrix} -\sigma_2^2 x_2^{(0)} \\ \sigma_1^2 x_1^{(0)} \end{bmatrix}$$

- For $k = 0$, we have

$$\boldsymbol{x}^{(1)} = \frac{x_1^{(0)} x_2^{(0)} (\sigma_1 - \sigma_2)}{\sigma_1^3 (x_1^{(0)})^2 + \sigma_2^3 (x_2^{(0)})^2} \begin{bmatrix} -\sigma_2^2 x_2^{(0)} \\ \sigma_1^2 x_1^{(0)} \end{bmatrix}$$

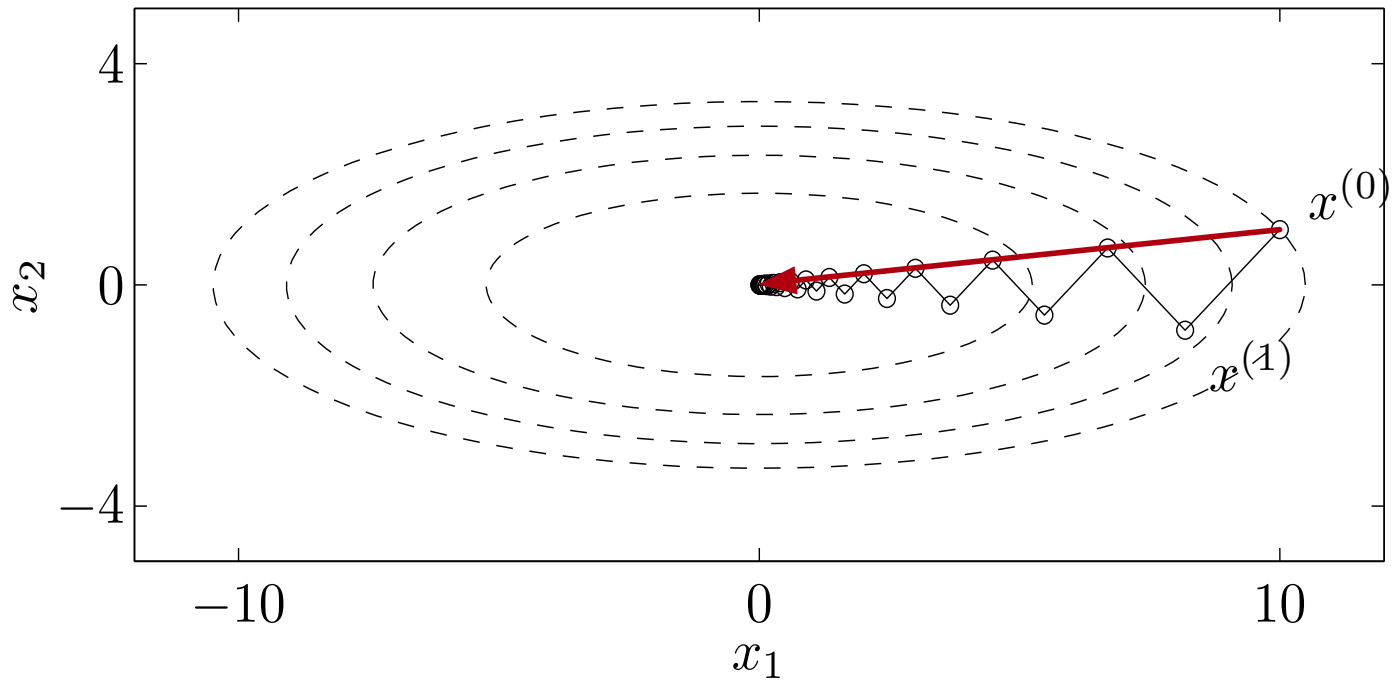
- For $k = 1$, we have

$$\boldsymbol{x}^{(2)} = \frac{x_1^{(1)} x_2^{(1)} (\sigma_1 - \sigma_2)}{\sigma_1^3 (x_1^{(1)})^2 + \sigma_2^3 (x_2^{(1)})^2} \begin{bmatrix} -\sigma_2^2 x_2^{(1)} \\ \sigma_1^2 x_1^{(1)} \end{bmatrix} = \rho \boldsymbol{x}^{(0)}$$

where

$$\rho = \frac{\sigma_1 \sigma_2 (\sigma_1 - \sigma_2)^2 (x_1^{(0)})^2 (x_2^{(0)})^2}{\left[\sigma_1 (x_1^{(0)})^2 + \sigma_2 (x_2^{(0)})^2\right] \left[\sigma_1^3 (x_1^{(0)})^2 + \sigma_2^3 (x_2^{(0)})^2\right]}$$

i.e., for every two iterations we move directly in the line toward the minimum point $\boldsymbol{x}^* = (0, 0)$


4
0
−4
−10
0
10
x_1
x_2
$x^{(0)}$
$x^{(1)}$

- Since f is a quadratic function

$$f(\boldsymbol{x}^{(2)}) - p^* = \rho^2 \left[f(\boldsymbol{x}^{(0)}) - p^* \right]$$

  i.e., the optimality gap forms a geometric series with ratio $\approx \rho$ between two consecutive iterations

- Since f is a quadratic function

$$f(\boldsymbol{x}^{(2)}) - p^* = \rho^2 \left[f(\boldsymbol{x}^{(0)}) - p^* \right]$$

  i.e., the optimality gap forms a geometric series with ratio $\approx \rho$ between two consecutive iterations

- In the context of iterative numerical methods, this is called linear convergence since the optimality gap of an iteration is a linear function of the optimality gap of the previous iteration

- Since f is a quadratic function

$$f(\boldsymbol{x}^{(2)}) - p^* = \rho^2 \left[f(\boldsymbol{x}^{(0)}) - p^* \right]$$

  i.e., the optimality gap forms a geometric series with ratio $\approx \rho$ between two consecutive iterations

- In the context of iterative numerical methods, this is called linear convergence since the optimality gap of an iteration is a linear function of the optimality gap of the previous iteration

- Such a convergence behavior is considered to be slow, especially when the ratio ρ (known as convergence rate) is close to 1

- Since f is a quadratic function

$$f(\boldsymbol{x}^{(2)}) - p^* = \rho^2 \left[f(\boldsymbol{x}^{(0)}) - p^* \right]$$

  i.e., the optimality gap forms a geometric series with ratio $\approx \rho$ between two consecutive iterations

- In the context of iterative numerical methods, this is called linear convergence since the optimality gap of an iteration is a linear function of the optimality gap of the previous iteration

- Such a convergence behavior is considered to be slow, especially when the ratio ρ (known as convergence rate) is close to 1

- Let $c := \sigma_1/\sigma_2 \geq 1$. The convergence rate

$$\rho = \frac{c(c-1)^2 (x_1^{(0)})^2 (x_2^{(0)})^2}{\left[c(x_1^{(0)})^2 + (x_2^{(0)})^2 \right] \left[c^3 (x_1^{(0)})^2 + (x_2^{(0)})^2 \right]}$$

  is apparently sensitive to the initial point $(x_1^{(0)}, x_2^{(0)})$

- The worse case occurs when $|x_2^{(0)}/x_1^{(0)}| = c$ and in this case, the convergence rate

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

which is very close to 1 when $c = \frac{\sigma_1}{\sigma_2} \gg 1$

- The worse case occurs when $|x_2^{(0)}/x_1^{(0)}| = c$ and in this case, the convergence rate

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

which is very close to 1 when $c = \frac{\sigma_1}{\sigma_2} \gg 1$

- Finally, note here that the Hessian for the quadratic function $f(x_1, x_2) = \frac{1}{2}\left(\sigma_1 x_1^2 + \sigma_2 x_2^2\right)$ is

$$\nabla^2 f = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$$

so σ_1 and σ_2 are the largest and smallest eigenvalue of the Hessian respectively, and ρ is simply the condition number of the Hessian

Gradient descent for strongly convex functions

- For a general objective function f, gradient descent with exact line search is guaranteed to converge if f is strongly convex on its domain, i.e., there exists an $m > 0$ such that

$$\nabla^2 f(\boldsymbol{x}) \succeq m\boldsymbol{I}, \quad \forall \boldsymbol{x} \in \mathrm{dom}(f)$$

Gradient descent for strongly convex functions

- For a general objective function f, gradient descent with exact line search is guaranteed to converge if f is strongly convex on its domain, i.e., there exists an $m > 0$ such that

$$\nabla^2 f(\boldsymbol{x}) \succeq m\boldsymbol{I}, \quad \forall \boldsymbol{x} \in \mathrm{dom}(f)$$

- Under the strong convexity assumption, it can be shown that

$$f(\boldsymbol{x}^{(k+1)}) - p^* \leq \rho \left[f(\boldsymbol{x}^{(k)}) - p^* \right]$$

where

$$\rho = 1 - \frac{1}{\sup_{\boldsymbol{x} \in \mathrm{dom}(f)} \frac{\lambda_1(\nabla^2 f(\boldsymbol{x}))}{\lambda_n(\nabla^2 f(\boldsymbol{x}))}}$$

Gradient descent for strongly convex functions

- For a general objective function f, gradient descent with exact line search is guaranteed to converge if f is strongly convex on its domain, i.e., there exists an $m > 0$ such that

$$\nabla^2 f(\boldsymbol{x}) \succeq m\boldsymbol{I}, \quad \forall \boldsymbol{x} \in \mathrm{dom}(f)$$

- Under the strong convexity assumption, it can be shown that

$$f(\boldsymbol{x}^{(k+1)}) - p^* \leq \rho \left[f(\boldsymbol{x}^{(k)}) - p^* \right]$$

where

$$\rho = 1 - \frac{1}{\sup_{\boldsymbol{x} \in \mathrm{dom}(f)} \frac{\lambda_1(\nabla^2 f(\boldsymbol{x}))}{\lambda_n(\nabla^2 f(\boldsymbol{x}))}}$$

- That is, the gradient method in general exhibits approximately linear convergence, i.e., the optimality gap $f(\boldsymbol{x}^{(k)}) - p^*$ converges to zero approximately as a geometric series

- The convergence rate depends greatly on the condition number of the Hessians. Convergence can be very slow, even for problems that are moderately well conditioned (say, with condition number in the 100s). When the condition number is larger (say, 1000 or more) the gradient method is so slow that it is useless in practice

Steepest descent

- The first-order Taylor approximation of $f(\boldsymbol{x}+\boldsymbol{v})$ around $\boldsymbol{x}$ is

$$f(\boldsymbol{x}+\boldsymbol{v}) \approx f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^t \boldsymbol{v}$$

- The second term on the righthand side, $\nabla f(\boldsymbol{x})^t \boldsymbol{v}$, is the directional derivative of f at $\boldsymbol{x}$ in the direction $\boldsymbol{v}$. It gives the approximate change in f for a small step $\boldsymbol{v}$. The step $\boldsymbol{v}$ is a descent direction if the directional derivative is negative

Steepest descent

- The first-order Taylor approximation of $f(\boldsymbol{x}+\boldsymbol{v})$ around $\boldsymbol{x}$ is

$$f(\boldsymbol{x}+\boldsymbol{v}) \approx f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^t \boldsymbol{v}$$

- The second term on the righthand side, $\nabla f(\boldsymbol{x})^t \boldsymbol{v}$, is the directional derivative of f at $\boldsymbol{x}$ in the direction $\boldsymbol{v}$. It gives the approximate change in f for a small step $\boldsymbol{v}$. The step $\boldsymbol{v}$ is a descent direction if the directional derivative is negative

- Consider the problem of how to choose $\boldsymbol{v}$ to make the directional derivative as negative as possible. Since the directional derivative $\nabla f(\boldsymbol{x})^t \boldsymbol{v}$ is linear in $\boldsymbol{v}$, it can be made as negative as we like by taking $\boldsymbol{v}$ large (provided $\boldsymbol{v}$ is a descent direction). To make the question sensible we have to limit the length of $\boldsymbol{v}$ or normalize by it

Steepest descent

- The first-order Taylor approximation of $f(\boldsymbol{x}+\boldsymbol{v})$ around $\boldsymbol{x}$ is

$$f(\boldsymbol{x}+\boldsymbol{v}) \approx f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^t \boldsymbol{v}$$

- The second term on the righthand side, $\nabla f(\boldsymbol{x})^t \boldsymbol{v}$, is the directional derivative of f at $\boldsymbol{x}$ in the direction $\boldsymbol{v}$. It gives the approximate change in f for a small step $\boldsymbol{v}$. The step $\boldsymbol{v}$ is a descent direction if the directional derivative is negative

- Consider the problem of how to choose $\boldsymbol{v}$ to make the directional derivative as negative as possible. Since the directional derivative $\nabla f(\boldsymbol{x})^t \boldsymbol{v}$ is linear in $\boldsymbol{v}$, it can be made as negative as we like by taking $\boldsymbol{v}$ large (provided $\boldsymbol{v}$ is a descent direction). To make the question sensible we have to limit the length of $\boldsymbol{v}$ or normalize by it

- Let $\|\cdot\|$ be any norm on $\mathbb{R}^n$. We define a normalized steepest descent direction (with respect to the norm $\|\cdot\|$) as

$$\Delta \boldsymbol{x}_{nsd} = \arg\min\{\nabla f(\boldsymbol{x})^t \boldsymbol{v} : \|\boldsymbol{v}\| = 1\}$$

- It is also convenient to consider a steepest descent step $\Delta \boldsymbol{x}_{sd}$ that is unnormalized, by scaling the normalized steepest descent direction in a particular way:

$$\Delta \boldsymbol{x}_{sd} = \|\nabla f(\boldsymbol{x})\|_* \Delta \boldsymbol{x}_{nsd}$$

where $\|\cdot\|_*$ denotes the dual norm

- It is also convenient to consider a steepest descent step $\Delta \boldsymbol{x}_{sd}$ that is unnormalized, by scaling the normalized steepest descent direction in a particular way:

$$\Delta \boldsymbol{x}_{sd} = \|\nabla f(\boldsymbol{x})\|_* \Delta \boldsymbol{x}_{nsd}$$

  where $\|\cdot\|_*$ denotes the dual norm

- It can be shown that for the steepest descent step, we have

$$\nabla f(\boldsymbol{x})^t \Delta \boldsymbol{x}_{sd} = \|\nabla f(\boldsymbol{x})\|_* \nabla f(\boldsymbol{x})^t \Delta \boldsymbol{x}_{nsd} = -\|\nabla f(\boldsymbol{x})\|_*^2$$

- It is also convenient to consider a steepest descent step $\Delta \boldsymbol{x}_{sd}$ that is unnormalized, by scaling the normalized steepest descent direction in a particular way:

$$\Delta \boldsymbol{x}_{sd} = \|\nabla f(\boldsymbol{x})\|_* \Delta \boldsymbol{x}_{nsd}$$

  where $\|\cdot\|_*$ denotes the dual norm

- It can be shown that for the steepest descent step, we have

$$\nabla f(\boldsymbol{x})^t \Delta \boldsymbol{x}_{sd} = \|\nabla f(\boldsymbol{x})\|_* \nabla f(\boldsymbol{x})^t \Delta \boldsymbol{x}_{nsd} = -\|\nabla f(\boldsymbol{x})\|_*^2$$

- The steepest descent method uses the steepest descent direction as search direction (when exact line search is used, scale factors in the descent direction have no effect, so either normalized or unnormalized direction can be used)

Steepest descent for Euclidean norm

- If we take the norm $\|\cdot\|$ to be the Euclidean norm we find that the steepest descent direction is simply the negative gradient, i.e.,

$$\Delta \boldsymbol{x}_{sd} = -\nabla f(\boldsymbol{x})$$

  Thus, the steepest descent method for the Euclidean norm coincides with the gradient descent method

Steepest descent for quadratic norms

- Consider the quadratic norm

$$\|\boldsymbol{z}\|_{\boldsymbol{P}} = \sqrt{\boldsymbol{z}^t \boldsymbol{P} \boldsymbol{z}} = \|\boldsymbol{P}^{1/2} \boldsymbol{z}\|_2$$

  where $\boldsymbol{P}$ is a positive definite matrix

Steepest descent for quadratic norms

- Consider the quadratic norm

$$\|\boldsymbol{z}\|_{\boldsymbol{P}} = \sqrt{\boldsymbol{z}^t \boldsymbol{P} \boldsymbol{z}} = \|\boldsymbol{P}^{1/2} \boldsymbol{z}\|_2$$

  where $\boldsymbol{P}$ is a positive definite matrix

- The normalized steepest descent direction is given by

$$\Delta \boldsymbol{x}_{nsd} = -\left(\nabla f(\boldsymbol{x})^t \boldsymbol{P}^{-1} \nabla f(\boldsymbol{x})\right)^{-1/2} \boldsymbol{P}^{-1} \nabla f(\boldsymbol{x})$$

Steepest descent for quadratic norms

- Consider the quadratic norm

$$\|\boldsymbol{z}\|_{\boldsymbol{P}} = \sqrt{\boldsymbol{z}^t \boldsymbol{P} \boldsymbol{z}} = \|\boldsymbol{P}^{1/2} \boldsymbol{z}\|_2$$

  where $\boldsymbol{P}$ is a positive definite matrix

- The normalized steepest descent direction is given by

$$\Delta \boldsymbol{x}_{nsd} = -\left(\nabla f(\boldsymbol{x})^t \boldsymbol{P}^{-1} \nabla f(\boldsymbol{x})\right)^{-1/2} \boldsymbol{P}^{-1} \nabla f(\boldsymbol{x})$$

- The dual norm is given by $\|\boldsymbol{z}\|_* = \|\boldsymbol{P}^{-1/2} \boldsymbol{z}\|_2$, so the steepest descent step with respect to $\|\cdot\|_{\boldsymbol{P}}$ is given by

$$\Delta \boldsymbol{x}_{sd} = -\boldsymbol{P}^{-1} \nabla f(\boldsymbol{x})$$

Steepest descent for quadratic norms

- Consider the quadratic norm

$$\|\boldsymbol{z}\|_{\boldsymbol{P}} = \sqrt{\boldsymbol{z}^t \boldsymbol{P} \boldsymbol{z}} = \|\boldsymbol{P}^{1/2} \boldsymbol{z}\|_2$$

  where $\boldsymbol{P}$ is a positive definite matrix

- The normalized steepest descent direction is given by

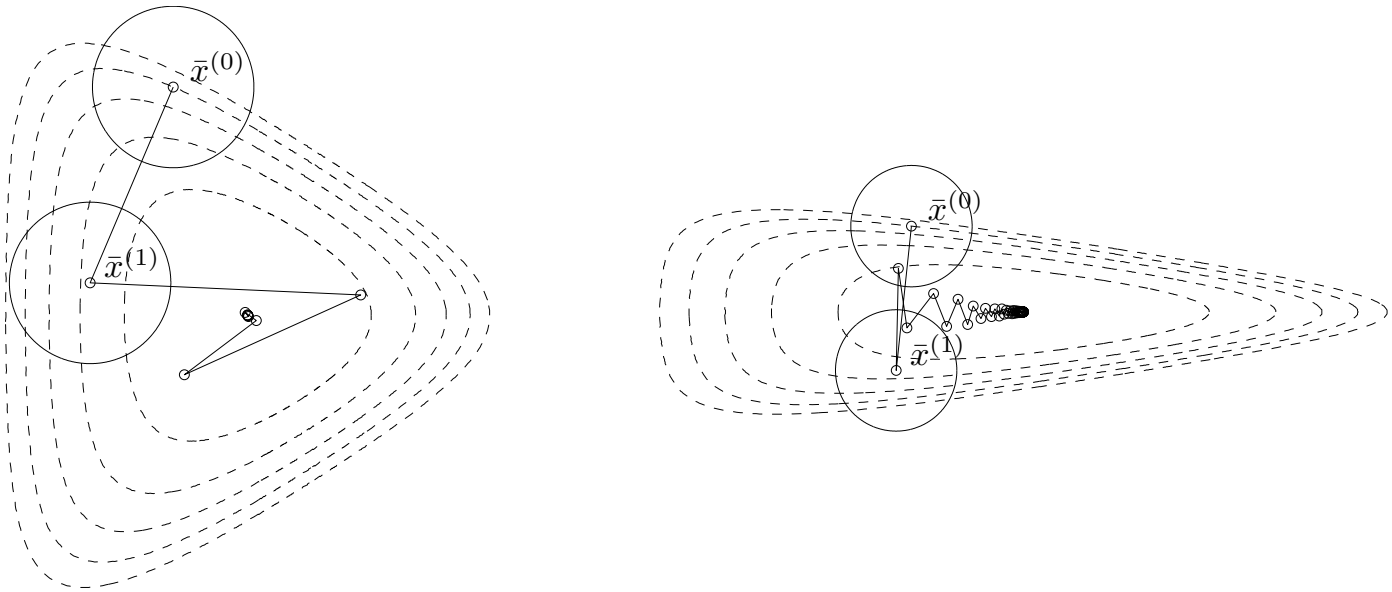
$$\Delta \boldsymbol{x}_{nsd} = -\left(\nabla f(\boldsymbol{x})^t \boldsymbol{P}^{-1} \nabla f(\boldsymbol{x})\right)^{-1/2} \boldsymbol{P}^{-1} \nabla f(\boldsymbol{x})$$

- The dual norm is given by $\|\boldsymbol{z}\|_* = \|\boldsymbol{P}^{-1/2} \boldsymbol{z}\|_2$, so the steepest descent step with respect to $\|\cdot\|_{\boldsymbol{P}}$ is given by

$$\Delta \boldsymbol{x}_{sd} = -\boldsymbol{P}^{-1} \nabla f(\boldsymbol{x})$$

- Alternatively, the steepest descent method in the quadratic norm $\|\cdot\|_{\boldsymbol{P}}$ can be thought of as the gradient method applied to the problem after the change of coordinates

$$\tilde{\boldsymbol{x}} = \boldsymbol{P}^{1/2} \boldsymbol{x}$$



- This observation provides a prescription for choosing $\boldsymbol{P}$: It should be chosen so that the Hessians with respect to $\tilde{\boldsymbol{x}}$ are well conditioned

Steepest descent for ℓ_1-norm

- Consider the steepest descent method for the ℓ_1-norm. It can be shown that a normalized steepest descent direction is given by

$$\Delta \boldsymbol{x}_{nsd} = -\text{sgn}\left(\frac{\partial f(\boldsymbol{x})}{\partial x_{i^*}}\right)\boldsymbol{e}_{i^*}$$

where $i^* = \arg\max_i \left|\frac{\partial f(\boldsymbol{x})}{\partial x_i}\right|$, and $\boldsymbol{e}_i$ is the ith standard basis vector

Steepest descent for ℓ_1-norm

- Consider the steepest descent method for the ℓ_1-norm. It can be shown that a normalized steepest descent direction is given by

$$\Delta \boldsymbol{x}_{nsd} = -\text{sgn}\left(\frac{\partial f(\boldsymbol{x})}{\partial x_{i^*}}\right)\boldsymbol{e}_{i^*}$$

  where $i^* = \arg\max_i \left|\frac{\partial f(\boldsymbol{x})}{\partial x_i}\right|$, and $\boldsymbol{e}_i$ is the ith standard basis vector

- An unnormalized steepest descent step is then

$$\Delta \boldsymbol{x}_{sd} = \|\nabla f(\boldsymbol{x})\|_\infty \Delta \boldsymbol{x}_{nsd} = -\frac{\partial f(\boldsymbol{x})}{\partial x_{i^*}}\boldsymbol{e}_{i^*}$$

Steepest descent for ℓ_1-norm

- Consider the steepest descent method for the ℓ_1-norm. It can be shown that a normalized steepest descent direction is given by

$$\Delta \boldsymbol{x}_{nsd} = -\text{sgn}\left(\frac{\partial f(\boldsymbol{x})}{\partial x_{i^*}}\right)\boldsymbol{e}_{i^*}$$

  where $i^* = \arg\max_i \left|\frac{\partial f(\boldsymbol{x})}{\partial x_i}\right|$, and $\boldsymbol{e}_i$ is the ith standard basis vector

- An unnormalized steepest descent step is then

$$\Delta \boldsymbol{x}_{sd} = \|\nabla f(\boldsymbol{x})\|_\infty \Delta \boldsymbol{x}_{nsd} = -\frac{\partial f(\boldsymbol{x})}{\partial x_{i^*}}\boldsymbol{e}_{i^*}$$

- Thus, the normalized steepest descent step in ℓ_1-norm can always be chosen to be a standard basis vector (or a negative standard basis vector). It is the coordinate axis direction along which the approximate decrease in f is greatest

Coordinate descent

- The steepest descent algorithm in the ℓ_1 naturally motivates the following coordinate descent algorithm: At each iteration we select a component of $\nabla f(\boldsymbol{x})$ with maximum absolute value, and then decrease or increase the corresponding component of $\boldsymbol{x}$, according to the sign of $(\nabla f(\boldsymbol{x}))_{i^*}$

Coordinate descent

- The steepest descent algorithm in the ℓ_1 naturally motivates the following coordinate descent algorithm: At each iteration we select a component of $\nabla f(\boldsymbol{x})$ with maximum absolute value, and then decrease or increase the corresponding component of $\boldsymbol{x}$, according to the sign of $(\nabla f(\boldsymbol{x}))_{i^*}$
- Since only one component of the variable $\boldsymbol{x}$ is updated at each iteration, this can greatly simplify, or even trivialize, the line search

Coordinate descent

- The steepest descent algorithm in the ℓ_1 naturally motivates the following coordinate descent algorithm: At each iteration we select a component of $\nabla f(\boldsymbol{x})$ with maximum absolute value, and then decrease or increase the corresponding component of $\boldsymbol{x}$, according to the sign of $(\nabla f(\boldsymbol{x}))_{i^*}$
- Since only one component of the variable $\boldsymbol{x}$ is updated at each iteration, this can greatly simplify, or even trivialize, the line search
- Consider, for example, the problem:

$$\text{minimize} \quad f(\boldsymbol{x}) = \ln \left(\sum_{i=1}^{n} \sum_{j=1}^{n} M_{i,j}^2 e^{x_i - x_j} \right)$$

- It is easy to minimize f one component at a time. Keeping all components except the kth fixed, we can write

$$f(\boldsymbol{x}) = \ln(\alpha_k + \beta_k e^{-x_k} + \gamma_k e^{x_k})$$

- It is easy to minimize f one component at a time. Keeping all components except the kth fixed, we can write

$$f(\boldsymbol{x}) = \ln(\alpha_k + \beta_k e^{-x_k} + \gamma_k e^{x_k})$$

- The minimum of $f(\boldsymbol{x})$, as a function of x_k, is obtained for

$$x_k = \frac{1}{2} \ln \left(\frac{\beta_k}{\gamma_k} \right)$$

So for this problem an exact line search can be carried out using a simple analytical formula

Convergence analysis

- Similar to gradient descent, it can be shown that for strongly convex functions, steepest descent with exact line search also converges linearly to the minimum point, and the linear convergence rate is again mainly controlled by the condition number of the Hessians

Newton's methods

- The second-order Taylor approximation of $f(\boldsymbol{x}+\boldsymbol{v})$ around $\boldsymbol{x}$ is

$$f(\boldsymbol{x}+\boldsymbol{v}) \approx f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^t \boldsymbol{v} + \frac{1}{2}\boldsymbol{v}^t \nabla^2 f(\boldsymbol{x}) \boldsymbol{v}$$

  which is a convex quadratic function of $\boldsymbol{v}$ and minimized at

$$\boldsymbol{v} = -\nabla^2 f(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x})$$

Newton's methods

- The second-order Taylor approximation of $f(\boldsymbol{x}+\boldsymbol{v})$ around $\boldsymbol{x}$ is

$$f(\boldsymbol{x}+\boldsymbol{v}) \approx f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^t \boldsymbol{v} + \frac{1}{2}\boldsymbol{v}^t \nabla^2 f(\boldsymbol{x}) \boldsymbol{v}$$

  which is a convex quadratic function of $\boldsymbol{v}$ and minimized at

$$\boldsymbol{v} = -\nabla^2 f(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x})$$

- This suggests the following pure Newton method:

$$\Delta \boldsymbol{x}_{nt} = -\nabla^2 f(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x}) \quad \text{and} \quad t = 1$$

Newton's methods

- The second-order Taylor approximation of $f(\boldsymbol{x}+\boldsymbol{v})$ around $\boldsymbol{x}$ is

$$f(\boldsymbol{x}+\boldsymbol{v}) \approx f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^t \boldsymbol{v} + \frac{1}{2}\boldsymbol{v}^t \nabla^2 f(\boldsymbol{x}) \boldsymbol{v}$$

  which is a convex quadratic function of $\boldsymbol{v}$ and minimized at

$$\boldsymbol{v} = -\nabla^2 f(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x})$$

- This suggests the following pure Newton method:

$$\Delta \boldsymbol{x}_{nt} = -\nabla^2 f(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x}) \quad \text{and} \quad t = 1$$

- It is straightforward to verify that

$$\nabla f(\boldsymbol{x})^t \Delta \boldsymbol{x}_{nt} = -\nabla f(\boldsymbol{x})^t \nabla^2 f(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x}) < 0$$

  unless $\nabla f(\boldsymbol{x}) = \boldsymbol{0}$

- Unfortunately, pure Newton method may not be a descent method. That is, it is possible that $f(\boldsymbol{x}^{(k+1)}) > f(\boldsymbol{x}^{(k)})$, and this may happen if our starting point is far away from the minimum point

- Unfortunately, pure Newton method may not be a descent method. That is, it is possible that $f(\boldsymbol{x}^{(k+1)}) > f(\boldsymbol{x}^{(k)})$, and this may happen if our starting point is far away from the minimum point
- On the other hand, if the starting point is sufficiently close to the minimum point, it is observed that pure Newton method exhibits superior convergence properties: The optimality gap decreases as a quadratic function of the optimality gap from the previous iteration

- Unfortunately, pure Newton method may not be a descent method. That is, it is possible that $f(\boldsymbol{x}^{(k+1)}) > f(\boldsymbol{x}^{(k)})$, and this may happen if our starting point is far away from the minimum point

- On the other hand, if the starting point is sufficiently close to the minimum point, it is observed that pure Newton method exhibits superior convergence properties: The optimality gap decreases as a quadratic function of the optimality gap from the previous iteration

- To overcome the starting-point issue, one may consider using exact line search as opposed to a constant step size $t = 1$. This leads to the so-called damped Newton method

Convergence analysis

- The convergence behavior of damped Newton method can be divided into two stages

Convergence analysis

- The convergence behavior of damped Newton method can be divided into two stages
- In the first, the damped Newton stage, we are far from the minimum point (as measured by $\|\nabla f(\boldsymbol{x}^{(k)})\|_2$), but we make constant progress, i.e., $f(\boldsymbol{x}^{(k+1)}) \leq f(\boldsymbol{x}^{(k)}) - \gamma$ for some positive constant γ, (in the worst case) towards the optimal solution

Convergence analysis

- The convergence behavior of damped Newton method can be divided into two stages
- In the first, the damped Newton stage, we are far from the minimum point (as measured by $\|\nabla f(\boldsymbol{x}^{(k)})\|_2$), but we make constant progress, i.e., $f(\boldsymbol{x}^{(k+1)}) \leq f(\boldsymbol{x}^{(k)}) - \gamma$ for some positive constant γ, (in the worst case) towards the optimal solution
- When $\|\nabla f(\boldsymbol{x}^{(k)})\|_2$ is sufficiently small, we enter the quadratically convergent phase, where things are settled very quickly: At most six or so iterations are required to produce a solution of very high accuracy

Convergence analysis

- The convergence behavior of damped Newton method can be divided into two stages
- In the first, the damped Newton stage, we are far from the minimum point (as measured by $\|\nabla f(\boldsymbol{x}^{(k)})\|_2$), but we make constant progress, i.e., $f(\boldsymbol{x}^{(k+1)}) \leq f(\boldsymbol{x}^{(k)}) - \gamma$ for some positive constant γ, (in the worst case) towards the optimal solution
- When $\|\nabla f(\boldsymbol{x}^{(k)})\|_2$ is sufficiently small, we enter the quadratically convergent phase, where things are settled very quickly: At most six or so iterations are required to produce a solution of very high accuracy
- Finally, we mention that a very important feature of Newton's methods is that it is independent of linear (or affine) changes of coordinates. This is in stark contrast to the gradient (or steepest descent) method, which is strongly affected by changes of coordinates

Backtracking line search

- Most line searches used in practice are inexact: The step length is chosen to approximately minimize f along the ray $\{\boldsymbol{x} + t\Delta\boldsymbol{x} : t \geq 0\}$, or even to just reduce f "enough"

Backtracking line search

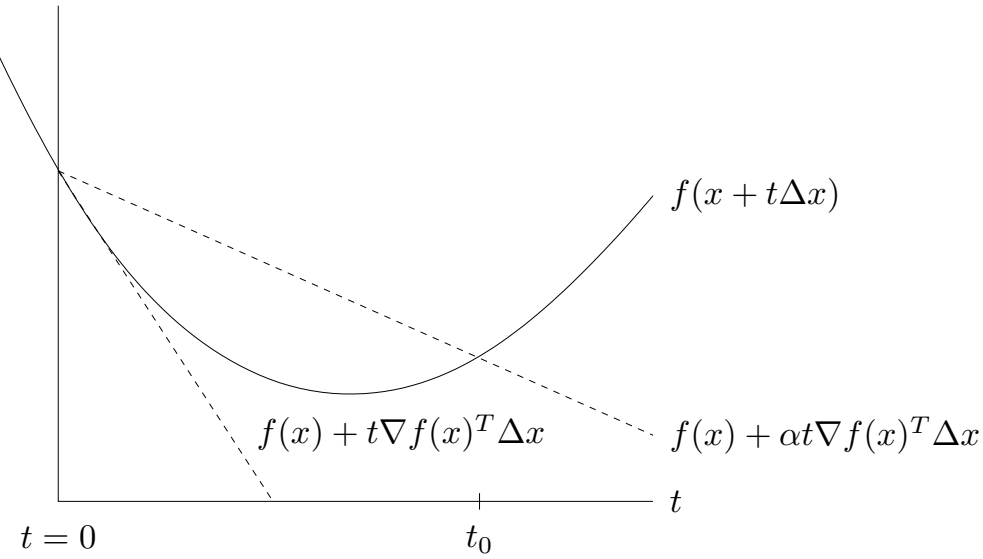
- Most line searches used in practice are inexact: The step length is chosen to approximately minimize f along the ray $\{\boldsymbol{x} + t\Delta\boldsymbol{x} : t \geq 0\}$, or even to just reduce f "enough"
- One inexact line search method that is very simple and quite effective is called backtracking line search. It depends on two constants α and β with $0 < \alpha < 0.5$ and $0 < \beta < 1$

Backtracking line search

- Most line searches used in practice are inexact: The step length is chosen to approximately minimize f along the ray $\{\boldsymbol{x} + t\Delta\boldsymbol{x} : t \geq 0\}$, or even to just reduce f "enough"
- One inexact line search method that is very simple and quite effective is called backtracking line search. It depends on two constants α and β with $0 < \alpha < 0.5$ and $0 < \beta < 1$
- The line search is called backtracking because it starts with a unit step size and then reduces it by the factor β until the stopping condition

$$f(\boldsymbol{x} + t\Delta\boldsymbol{x}) \leq f(\boldsymbol{x}) + \alpha t \nabla f(\boldsymbol{x})^t \Delta\boldsymbol{x}$$

holds



- Since $\Delta \boldsymbol{x}$ is a descent direction, we have $\nabla f(\boldsymbol{x})^t \Delta \boldsymbol{x} < 0$, so for small enough t we have

$$f(\boldsymbol{x} + t\Delta \boldsymbol{x}) \approx f(\boldsymbol{x}) + t\Delta f(\boldsymbol{x})^t \Delta \boldsymbol{x} < f(\boldsymbol{x}) + \alpha t \Delta f(\boldsymbol{x})^t \Delta \boldsymbol{x}$$

  which shows that the backtracking line search eventually terminates. The constant α can be interpreted as the fraction of the decrease in f predicted by linear extrapolation that we will accept