MAT 3007 – Optimization

Newton's Method and the Projected Gradient Method

Lecture 18 *July 20th*

Andre Milzarek SDS / CUHK-SZ

Repetition

Recap: Optimization Methods

One-Dimensional Problems:

- Bisection method and golden section method.

High-Dimensional Problems:

- Gradient descent method: Choose $d^k = -\nabla f(x^k)$ as descent direction.
- Stepsize: Exact line search and backtracking line search.

Convergence and Properties:

- Global convergence: every accumulation point is a stationary point (independent of the initial point).
- Directions are perpendicular when using exact line search.
- If ∇f is Lipschitz continuous and f is (strongly) convex, then we can expect a linear convergence rate.

Newton's Method – in $\mathbb{R}^n$

Newton's Method in $\mathbb{R}$

In the one-dimensional case, Newton's method is given by:

$$x^{k+1} = x^k - \frac{g(x^k)}{g'(x^k)}$$

where $g(x) = f'(x)$.

Two ways to view Newton's method:

- Approximate the derivative function $g = f'$ locally by a linear function.
- Approximate the original function locally by a quadratic function.

$\rightsquigarrow$ Use the same perspectives to derive Newton's method in $\mathbb{R}^n$.

Newton's Method in High Dimensional Case

We want to solve $\min_{x\in\mathbb{R}^n} f(x)$ with $f : \mathbb{R}^n \to \mathbb{R}$.

At x^k, we approximate the objective function by its second order Taylor expansion:

$$f(x) \approx f(x^k) + \nabla f(x^k)^\top (x - x^k) + \frac{1}{2}(x - x^k)^\top \nabla^2 f(x^k)(x - x^k)$$

We minimize this quadratic approximation and get:

$$x = x^k - (\nabla^2 f(x^k))^{-1} \nabla f(x^k).$$

This motivates to define the search direction (Newton direction):

$$d^k = -(\nabla^2 f(x^k))^{-1} \nabla f(x^k).$$

In the gradient descent method, the direction is $-\nabla f(x^k)$.

- Newton's method refines the search direction by using the second-order information: $\nabla^2 f(x^k)$.

Newton's Method in High Dimensional Case

We can also consider the nonlinear equation $\nabla f(x) = 0$.

Using a Taylor expansion at x^k, we have

$$\nabla f(x) \approx \nabla f(x^k) + \nabla^2 f(x^k)(x - x^k) =: q_k(x).$$

The solution to $q_k(x) = 0$ is

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

which is also Newton's step.

- In these derivations, we assume that $\nabla^2 f(x)$ is invertible in the search region.

Connection to Descent Directions

Recap: A vector d is a descent direction if $\nabla f(x)^\top d < 0$.

- If we go a very small step in that direction, the objective value must be decreasing (due to Taylor's expansion).
- In the gradient descent method, we have $d = -\nabla f(x)$ and

$$\nabla f(x)^\top d = -\|\nabla f(x)\|^2 < 0.$$

Newton's Step as a Descent Step

In Newton's method, we have

$$d = -(\nabla^2 f(x))^{-1} \nabla f(x).$$

Then, it holds that:

$$\nabla f(x)^\top d = -\nabla f(x)^\top (\nabla^2 f(x))^{-1} \nabla f(x).$$

- If f is convex, then $\nabla^2 f(x)$ is positive semidefinite and we obtain $\nabla f(x)^\top d \leq 0$.
- If $\nabla^2 f(x)$ is positive definite, then $\nabla f(x)^\top d < 0$.

$\rightsquigarrow$ In this case, Newton's direction is a descent direction.

Step Length

As we said earlier, Newton's method may not converge unless the starting point is close.

One way to ensure convergence is to again use a step size parameter α_k in

$$x^{k+1} = x^k + \alpha_k d^k$$

where $d^k = -(\nabla^2 f(x^k))^{-1} \nabla f(x^k)$ is Newton's direction.

- We can use backtracking line search to determine α_k.

Complete Procedure of Newton's Method

The Newton Method

1. Initialization: Select an initial point $x^0 \in \mathbb{R}^n$.

For $k = 0, 1, \ldots$:

2. Compute the Newton direction d^k which is the solution of the linear system
$$\nabla^2 f(x^k) d^k = -\nabla f(x^k).$$
3. Choose a step size α_k by backtracking line search and calculate $x^{k+1} = x^k + \alpha_k d^k$.
4. If $\|\nabla f(x^{k+1})\| \leq \varepsilon$, then STOP and x^{k+1} is the output.

- We can also check whether d^k is a good descent direction:

$$-\nabla f(x^k)^\top d^k \geq \gamma_1 \min\{1, \|d^k\|^{\gamma_2}\}\|d^k\|^2, \quad \gamma_1, \gamma_2 \in (0, 1).$$

$\rightsquigarrow$ Otherwise we use the gradient direction: $d^k = -\nabla f(x^k)$.

Theorem: Convergence of Newton's Method

Let f be twice cont. diff. and let x^* be a local minimizer of f. For some given $\varepsilon > 0$ assume that:

- there exists $\mu > 0$ with $\nabla^2 f(x) \succeq \mu I$ for any $x \in B_\varepsilon(x^*)$.
- there exists $L > \mu$ with $\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x - y\|$ for all $x, y \in B_\varepsilon(x^*)$.

Let $(x^k)_k$ be generated by Newton's method. Then for $k = 0, 1, \ldots$

$$\|x^{k+1} - x^*\| \leq \frac{L}{2\mu}\|x^k - x^*\|^2$$

and in addition, if $\|x^0 - x^*\| \leq \frac{\mu \min\{1,\varepsilon\}}{L}$, then

$$\|x^k - x^*\| \leq \frac{2\mu}{L}\left(\frac{1}{2}\right)^{2^k}, \quad k = 0, 1, 2, \ldots$$

Convergence: Summary

Convergence Newton's Method:

- "Hessian locally uniformly positive definite + Lipschitz continuous" and "x^0 close to x^*"

$$\Longrightarrow \quad (x^k)_k \text{ converges q-quadratically to } x^*$$

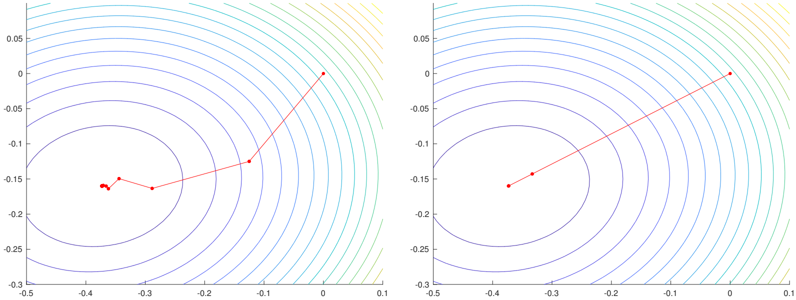
- Proof: See Theorem 5.2 in Beck: "Introduction to Nonlinear Optimization".

Example

Minimize

$$f(x) = \exp(x_1 + x_2) + x_1^2 + 3x_2^2 - x_1x_2$$

using the gradient method and Newton's method.



- Left: Gradient method (Armijo). Right: Newton's method.

Newton's Method is Fast, But ...

It requires computing the second-order derivative in each iteration.

- This might be computationally expensive, especially if the second-order derivative does not have a closed-form (when f is only available as a black box).
- Memory: we also require space to store the Hessian matrix.
  - In $\mathbb{R}^n$, we need additional allocation space of size n^2, compared to the "n space" required for gradient descent method.
- More matrix computations are required in each iteration.

A Partial Solution: Quasi-Newton Methods

- We do not calculate $\nabla^2 f(x)$ in each step. Instead, we only approximate it using the past gradients.
- More Ideas: solve the Newton system only approximately ($\rightsquigarrow$ inexact Newton methods).

Gradient Descent Method vs Newton's Method

The two methods have the same framework:

Start from a point x^0, set tolerance $\epsilon > 0$ and $k = 0$.

1. If $||\nabla f(x^k)|| < \epsilon$, then output x^k, otherwise compute the search direction d^k and continue;
2. Choose step size α_k by backtracking line search;
3. Set $x^{k+1} = x^k + \alpha_k d^k$, $k = k + 1$, go back to step 1.

In the gradient descent method:

$$d^k = -\nabla f(x^k)$$

In Newton's method:

$$d^k = -(\nabla^2 f(x^k))^{-1} \nabla f(x^k)$$

Tradeoff between Newton and Gradient Method

- Gradient method is simple and fast in each iteration and needs less storage. However, it takes much more iterations to converge.
- Newton's method takes much less iterations to converge. However, each iteration is more expensive in both computation and storage space.

There are many numerical improvements that one can make but the main tradeoff remains:

- Whether the gradient method or Newton's method should be used depends largely on the problem at hand.

Algorithms for Constrained Problems

Agenda

In the following, we will extend our discussions to constrained optimization problems.

- We will introduce one algorithm for such problems – the projected gradient method.

Constrained Optimization

We consider the following constrained optimization problem:

$$\begin{array}{ll} \text{minimize}_x & f(x) \\ \text{subject to} & x \in \Omega. \end{array}$$

In the methods for unconstrained problems, the main idea is:

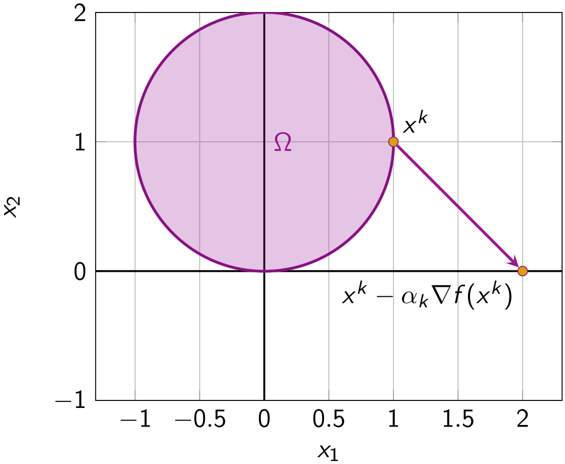
- At each x^k, compute a descent direction d^k.
- Then find an appropriate stepsize α_k and update $x^{k+1} = x^k + \alpha_k d^k$.
- Both the gradient and Newton's method are based on this basic idea.

Under Additional Constraints:

- x^{k+1} can become infeasible!

In the following, we assume we have a feasible initial point.

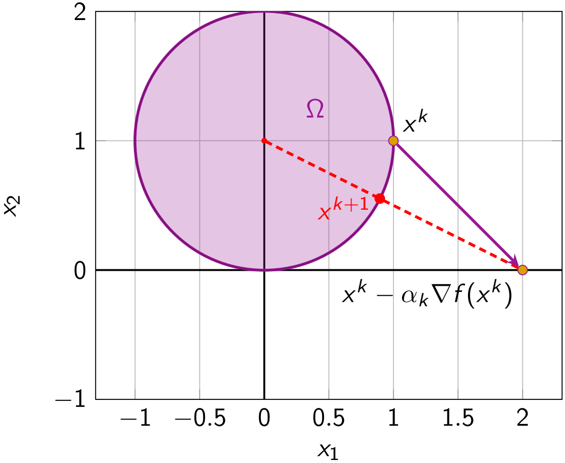
Illustration



- One solution to this problem is to use the projected gradient method.

Main Idea of the Projected Gradient Method

Project the gradient step back onto the feasible set:



- We can use the general framework by maintaining feasibility.

Projections

Euclidean Projections

We now define the term "projection" mathematically:

Definition: Euclidean Projections

Let $\Omega \subset \mathbb{R}^n$ be a nonempty, closed, convex set. The (Euclidean) projection of x onto Ω is defined as the unique minimizer y^* of the constrained optimization problem:

$$\min_y \ \frac{1}{2}\|x - y\|^2 \quad \text{s.t.} \quad y \in \Omega$$

and we write $y^* = \mathcal{P}_\Omega(x)$.

Observation:

- The projection $y^* = \mathcal{P}_\Omega(x)$ is the point in Ω that has the minimum distance to x.

Example I: Linear Constraints

We first consider the simple case where Ω consists of linear equality constraints:

$$\Omega = \{x : Ax = b\},$$

where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$ are given.

Euclidean Projection:

- Suppose that A has full row rank ($m \leq n$), then it holds that:

$$\mathcal{P}_\Omega(x) = x - A^\top (AA^\top)^{-1}[Ax - b].$$

Example II: Box Constraints

Suppose that Ω is given by box constraints:

$$\Omega = \{x \in \mathbb{R}^n : x_i \in [a_i, b_i],\ \forall\ i\} = [a, b],$$

where $a, b \in \mathbb{R}^n$, $a \leq b$, are given.

Euclidean Projection:

- The projection onto Ω can be computed as follows:

$$[\mathcal{P}_\Omega(x)]_i = \mathcal{P}_{[a_i, b_i]}(x_i) = \max\{\min\{x_i, b_i\}, a_i\} \quad \forall\ i.$$

Example III: Ball Constraints

Suppose that Ω is a Euclidean ball with radius $r > 0$ and center $m \in \mathbb{R}^n$, i.e.:

$$\Omega = \{x \in \mathbb{R}^n : \|x - m\| \leq r\}.$$

Euclidean Projection:

- The projection onto Ω can be computed as follows:

$$\mathcal{P}_\Omega(x) = \begin{cases} x & \text{if } \|x - m\| \leq r, \\ m + \frac{r}{\|x-m\|}(x - m) & \text{if } \|x - m\| \geq r. \end{cases}$$

Remark: Projections

Observation:

- For many sets, explicit formulae for the projections can be derived.
- Main Tool: KKT-conditions and convexity.
- Many more interesting projections can be expressed efficiently:

$$\mathcal{P}_{\Delta_n}(x), \quad \mathcal{P}_{\mathbb{S}^n_+}(X), \quad \dots$$

  where $\Delta_n := \{x : \mathbb{1}^\top x = 1,\, x \geq 0\}$ is the n-simplex and $\mathbb{S}^n_+$ is the set of positive semidefinite matrices.

- In general: an optimization problem needs to be solved to obtain $\mathcal{P}_\Omega(x)$!

Properties of Projections and Descent Directions

Optimization Problems with Convex Constraints

We consider constrained minimization problems of the form:

$$\min_{y} \; f(y) \quad \text{s.t.} \quad y \in \Omega, \tag{1}$$

where $\Omega \subset \mathbb{R}^n$ is a convex and closed set.

- We can derive the following optimality condition:

Theorem: FONC for Problems with Convex Constraints

Let f be C^1 on an open set that contains the convex, closed set $\Omega \subset \mathbb{R}^n$. Let $y^* \in \Omega$ be a local minimizer of (1), then:

$$\nabla f(y^*)^\top (y - y^*) \geq 0, \quad \forall \; y \in \Omega. \tag{2}$$

- If f is convex, then $y^* \in \Omega$ is global minimizer of (1) iff the FONC is satisfied.
- A point satisfying (2) is again called a stationary point of (1).

Properties of the Projection Mapping

⇝ Projections are a special case with $f(y) = \frac{1}{2}\|y - x\|^2$.

Projection Theorem

Let Ω be a nonempty, closed, and convex set. Then:

- A point y^* is the projection of x onto Ω, i.e., $y^* = \mathcal{P}_\Omega(x)$, if and only if

$$(y^* - x)^\top (y - y^*) \geq 0, \quad \forall\; y \in \Omega.$$

- The mapping $\mathcal{P}_\Omega : \mathbb{R}^n \to \mathbb{R}^n$ is Lipschitz continuous with constant $L = 1$.
- The vector x^* is a stationary point of (1) if and only if

$$x^* - \mathcal{P}_\Omega(x^* - \lambda \nabla f(x^*)) = 0 \quad \text{for any } \lambda > 0.$$

Projected Gradient Method: Basic Idea

In the gradient method, we perform a gradient step of the form:

$$x^k - \lambda_k \nabla f(x^k),$$

where $\lambda_k > 0$ is a step size.

Motivation & Strategy:

- Setting $x^{k+1} = x^k - \lambda_k \nabla f(x^k)$ might likely generate infeasible iterates: $x^{k+1} \notin \Omega$.
- Idea: We project the step $x^k - \lambda_k \nabla f(x^k)$ back onto Ω:

$$x^{k+1} = \mathcal{P}_\Omega(x^k - \lambda_k \nabla f(x^k)).$$

Drawback:

- How to choose $\lambda_k > 0$? If λ_k is determined by line search, every adjustment of λ_k requires to reevaluate the projection $\mathcal{P}_\Omega(x^k - \lambda_k \nabla f(x^k))$. This can be expensive!
- Can we guarantee descent and convergence?

Important Observation

Observation:

$$x^{k+1} = \mathcal{P}_\Omega(x^k - \lambda_k \nabla f(x^k)) = x^k + [\mathcal{P}_\Omega(x^k - \lambda_k \nabla f(x^k)) - x^k].$$

- This is close to our usual update form: $x^{k+1} = x^k + \alpha_k d^k$.
- Setting $d^k = x^k - \mathcal{P}_\Omega(x^k - \lambda_k \nabla f(x^k))$, we can consider:

$$x^{k+1} = x^k + \alpha_k d^k = (1 - \alpha_k)x^k + \alpha_k \mathcal{P}_\Omega(x^k - \lambda_k \nabla f(x^k)).$$

$\rightsquigarrow$ If $\alpha_k \in [0, 1]$, then the convexity of Ω implies that x^{k+1} will be feasible if $x^k \in \Omega$!

Why does / Can this work?

Descent Directions

Descent Directions

Let $x \in C$ and $\lambda > 0$ be given. If x is not a stationary point of (1), then the direction $d := \mathcal{P}_\Omega(x - \lambda \nabla f(x)) - x$ is a descent direction and it holds that

$$\nabla f(x)^\top d \leq -\frac{1}{\lambda}\|d\|^2 < 0.$$

- ▶ We can now reuse similar techniques like backtracking to generate step sizes!
- ▶ We can use $\|\mathcal{P}_\Omega(x^k - \lambda_k \nabla f(x^k)) - x^k\| \leq \varepsilon$ as stopping criterion.

⇝ The only difference between the gradient and the projected method is the search direction and stopping condition.

The Projected Gradient Method

Projected Gradient Method

1. Initialization: Choose an initial point $x^0 \in \Omega$ and $\sigma, \gamma \in (0,1)$.

For $k = 0, 1, \dots$:

2. Select $\lambda_k > 0$ and compute $\nabla f(x^k)$ and the new direction $d^k = \mathcal{P}_\Omega(x^k - \lambda_k \nabla f(x^k)) - x^k$.
3. If $\|d^k\| \leq \lambda_k \varepsilon$, then STOP and x^k is the output.
4. Choose a maximal step size $\alpha_k \in \{1, \sigma, \sigma^2, \dots\} \subset (0,1]$ that satisfies the Armijo condition
$$f(x^k + \alpha_k d^k) - f(x^k) \leq \gamma \alpha_k \cdot \nabla f(x^k)^\top d^k.$$
5. Set $x^{k+1} = x^k + \alpha_k d^k$.

Theorem: Convergence Projected Gradient Method

Assumptions: f is cont. diff., Ω is nonempty, convex, and closed and the step sizes $(\lambda_k)_k$ are bounded:

$$0 < \underline{\lambda} \leq \lambda_k \leq \overline{\lambda} \quad \forall \ k.$$

$\rightsquigarrow$ Every accumulation point of (x^k) is a stationary point.

If ∇f is additionally Lipschitz continuous, we obtain:

- If f is convex, we converge to global solutions of the problem.
- If $\lambda_k \leq \frac{2(1-\gamma)}{L}$, then $\alpha_k = 1$ will be accepted as step size!
- If f is (strongly) convex, then $(x^k)_k$ converges linearly to a global solution $x^* \in \Omega$.

Theorem: Convergence Projected Gradient Method

Assumptions: f is cont. diff., Ω is nonempty, convex, and closed and the step sizes $(\lambda_k)_k$ are bounded:

$$0 < \underline{\lambda} \leq \lambda_k \leq \overline{\lambda} \quad \forall\ k.$$

$\rightsquigarrow$ Every accumulation point of (x^k) is a stationary point.

If ∇f is additionally Lipschitz continuous, we obtain:

- If f is convex, we converge to global solutions of the problem.
- If $\lambda_k \leq \frac{2(1-\gamma)}{L}$, then $\alpha_k = 1$ will be accepted as step size!
- If f is (strongly) convex, then $(x^k)_k$ converges linearly to a global solution $x^* \in \Omega$.

Example I

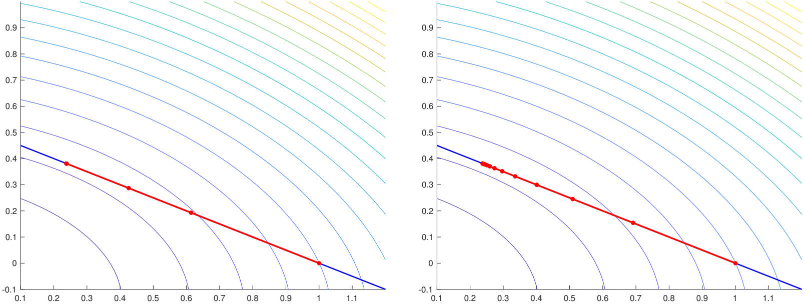
$$\begin{array}{rc} \text{minimize} & e^{x_1+x_2} + x_1^2 + 3x_2^2 - x_1x_2 \\ \text{subject to} & x_1 + 2x_2 = 1 \end{array}$$

Setting $A = [1, 2]$, $b = 1$, and $\Omega = \{x : Ax = b\}$, we have

$$\mathcal{P}_\Omega(x) = \left[\begin{array}{cc} 4/5 & -2/5 \\ -2/5 & 1/5 \end{array} \right] x + \begin{bmatrix} 1/5 \\ 2/5 \end{bmatrix}$$

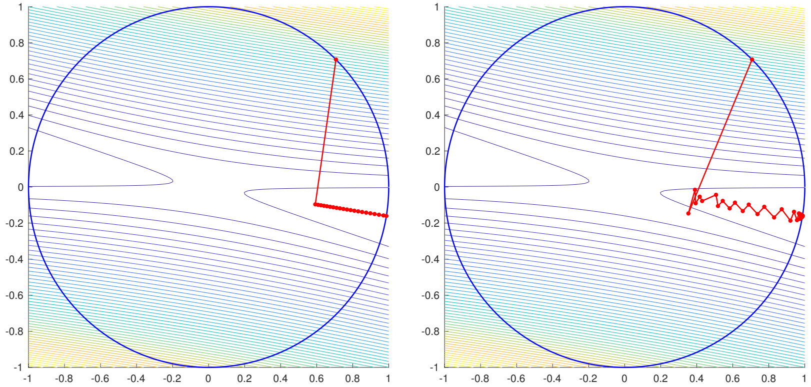
We use the initial feasible point $x^0 = (1, 0)^\top$ and $\lambda_k = 1$ and $\lambda_k = 0.1$.

Example I: Results



- Left: $\lambda_k = 1$. Right: $\lambda_k = 0.1$.

Example II



- We apply the projected gradient method to the problem:

$$\min_{x} \ \frac{1}{2}x^\top Ax \quad \text{s.t.} \quad \|x\| \leq 1, \quad A = \begin{pmatrix} 0 & 1 \\ 1 & 6 \end{pmatrix}.$$

- We set $\lambda_k = \lambda = L^{-1}$ and $L = \|A\| = \lambda_{\max}(A)$ and $\lambda_k = 1$.

Questions?