Optimization Theory and Algorithm II **September 9, 2022**

Lecture 2: Review II

Lecturer:Xiangyu Chang *Scribe: Xiangyu Chang*

Edited by: Xiangyu Chang

1 Review

1.1 Modeling in Optimization

Example 1 *Generalized Linear Model (GLM). Let us consider the following three management problems.*

- $b = House\ Price = F(a_1 = number\ of\ rooms, a_2 = school\ distriction, a_3, \dots)$
- $b = Credit\ Rate = F(a_1 = education, a_2 = salary, a_3, \dots)$
- $b = Number\ of\ Visit\ this\ month = F(a_1 = number\ of\ visit\ last\ month, a_2 = RFM, a_3, \dots)$

In this example, we introduced three classic regression models, linear regression(house price), Poisson regression (number of visit this month) and logistic regression (credit rate) derived from GLM. We parameterized the parameters in the statistic models as a linear function of covariant variables $\mathbf{a}$*, and formed the optimization problem from the likelihood.*

Consider the input-output pairs $\{\mathbf{a}_i, b_i\}_{i=1}^m$ *as the data. The procedure can be summarized as following recipe,*

1. *write down a probabilistic model for* b_i
2. *link model parameter* $\mathbf{x}$ *with* $\mathbf{a}_i$
3. *formed the optimization problem using maximum likelihood that aim to discover* $\mathbf{x}$ *with all data* $\{\mathbf{a}_i, b_i\}_{i=1}^m$

Next we instantiate this recipe by three examples.

(i) Linear Regression: Given training data $\{\mathbf{a}_i, b_i\}_{i=1}^m$ *with* $\mathbf{a}_i \in \mathbb{R}^p$ *and* $b_i \in \mathbb{R}$*. Suppose each* $b_i \overset{i.i.d.}{\sim} N(\mu_i, \sigma^2)$*, that is*

$$\begin{aligned} P(b_i|\mu_i, \sigma^2) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\frac{(b_i - \mu_i)^2}{2\sigma^2}\} \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\frac{b_i^2}{2\sigma^2}\} \exp\{-\frac{\frac{1}{2}\mu_i^2 - b_i\mu_i}{\sigma^2}\}. \end{aligned}$$

It is convention to choose the parameters that multiply b_i *as the linear function of the variables* $\mathbf{a}_i$ *with the parametric coefficient* $\mathbf{x}$*. Here we make the assumption that*

$$\theta_i = \mu_i = \langle \mathbf{a}_i, \mathbf{x} \rangle.$$

We wish to examine how we find a good $\mathbf{x}$ *to make this work. Our strategy for this is to maximize the* likelihood *of all observations* $\{b_i\}$ *as a function of* $\mathbf{x}$*, i.e.*

$$\max_{\mathbf{x}} \prod_i \exp\{-\frac{1}{\sigma^2}(\frac{1}{2}\mu_i^2 - b_i\mu_i)\} \Rightarrow \max_{\mathbf{x}} \log \prod_i \exp\{-\frac{1}{2\sigma^2}(\langle \mathbf{a}_i, x \rangle^2 - b_i \langle \mathbf{a}_i, \mathbf{x} \rangle)\}.$$

To maximize this expression, we take the negative log of the expression, i.e. we want to minimize

$$\min_{\mathbf{x}} \ \frac{1}{\sigma^2}\sum_{i=1}^{n}(\frac{1}{2}\langle \mathbf{a}_i, \mathbf{x}\rangle^2 - b_i\langle \mathbf{a}_i, \mathbf{x}\rangle).$$

To write it more compactly, we denote,

$$A = \begin{bmatrix} \mathbf{a}_1^\top \\ \vdots \\ \mathbf{a}_m^\top \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}.$$

And we have,

$$\sum_{i=1}^{m}\langle \mathbf{a}_i, x\rangle^2 = \|A\mathbf{x}\|^2, \qquad \sum_{i=1}^{m} b_i\langle \mathbf{a}_i, \mathbf{x}\rangle = \langle \mathbf{b}, A\mathbf{x}\rangle,$$

we get the minimization problem

$$\arg\min_x \ \frac{1}{2}\|A\mathbf{x}\|^2 - \langle b, A\mathbf{x}\rangle = \arg\min_x \ \frac{1}{2}\|A\mathbf{x} - \mathbf{b}\|^2$$

which is a linear least-squares regression problem.

(ii) Poisson Regression: The fitting problem is to minimize the negative log of the above expression with respect to $\mathbf{x}$*,*

$$\min_{\mathbf{x}} \ \sum_{i=1}^{m}\{\exp(\langle \mathbf{a}_i, \mathbf{x}\rangle) - b_i\langle \mathbf{a}_i, \mathbf{x}\rangle\}.$$

which is a simple Poisson regression.

(iii) Logistic Regression: Let $b_i \in \{0, 1\}$ *and* p_i *be the probability of success, i.e.*

$$\begin{aligned} p(b_i|p_i) &= p_i^{b_i}(1-p_i)^{1-b_i} \\ &= \exp\{b_i \ln p_i + (1-b_i)\ln(1-p_i)\} \\ &= \exp\{b_i(\ln p_i - \ln(1-p_i)) + \ln(1-p_i)\} \\ &= \exp\{b_i \ln \tfrac{p_i}{1-p_i} + \ln(1-p_i)\}. \end{aligned}$$

The choice $\theta_i = \ln \frac{p_i}{1-p_i}$ *is called the canonical parameter, i.e.* $p_i = \frac{\exp(\theta_i)}{1+\exp(\theta_i)} = (1+\exp(-\theta_i))^{-1}$*. Letting* $\theta_i = \langle a_i, x\rangle$ *and noting that* $\ln(1-p_i) = -\ln(1+\exp(\theta_i))$ *the probability becomes*

$$p(b_i|\theta_i) = \exp\{b_i\theta_i - \ln(1+\exp(\theta_i))\}.$$

The fitting problem is found by minimizing the negative log of the above expression,

$$\min_x \sum_{i=1}^{m}[\ln(1+\exp(\theta_i)) - b_i\theta_i] = \min_x \sum_{i=1}^{m}\ln(1+\exp(\langle \mathbf{a}_i, \mathbf{x}\rangle)) - \langle \mathbf{b}, A\mathbf{x}\rangle.$$

(iv) GLM: We find that given a family of distributions for b_i*, given* μ_i, σ^2 *we have*

$$f(b_i|\mu_i, \sigma^2) = g_1(b_i, \sigma^2)\exp\{\frac{b_i\mu_i - g_2(\mu_i)}{g_3(\sigma^2)}\}$$

for some functions g_1, g_2, g_3*. And* g_2 *is given by*

1. $g_2(\mu_i) = \frac{1}{2}\mu_i^2$ *for linear regression,*
2. $g_2(\mu_i) = \exp(\mu_i)$ *for Poisson regression,*

3. $g_2(\mu_i) = \ln(1 + \exp(\mu_i))$ *for logistic regression.*

This gives us the problem

$$\min_{\mathbf{x}} \sum_{i=1}^{n} g_2(\langle \mathbf{a}_i, \mathbf{x} \rangle) - \langle \mathbf{b}, A\mathbf{x} \rangle.$$

The difficulty of this problem depends on properties of g_2. In these three cases g_2 is convex and smooth, but this won't always be the case. This motivates us to look at properties of continuous functions.

We will discuss basic function properties that will determine how good will an optimization algorithm perform on them.

Example 2 *(Management Decision Tree Analysis)*

A management decision tree is a branched flowchart showing multiple pathways for potential decisions and outcomes.

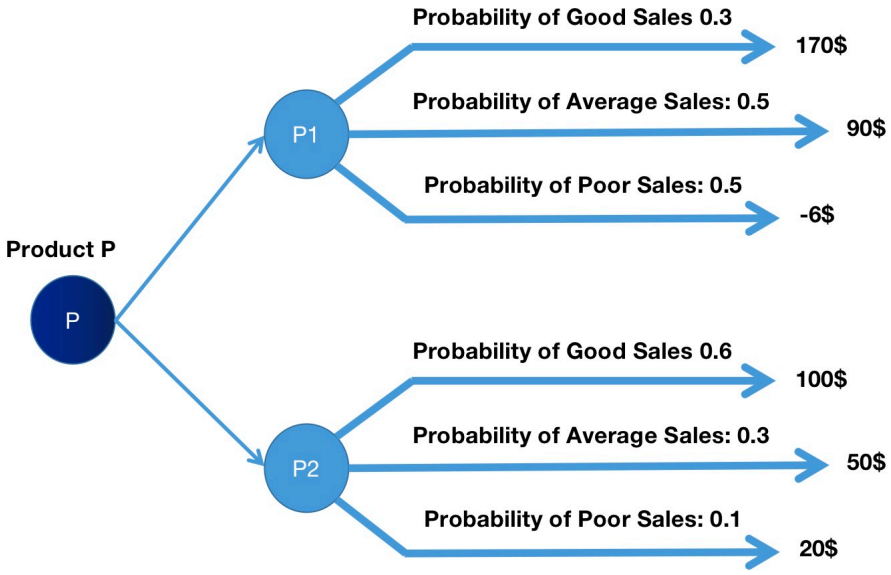

Figure 1: An example of Management Decision Tree

- *Suppose that a company is considering to develop a new product P. The product P includes two different types. The company employ a marketing research institute to study that which type is better?*
- *Based on the study results, the marketing research institute report: (1) If they produce the first type P1, then P1 has 0.3 chance for good sales with profit 170$ per unit; 0.5 chance for average sales with profit 90$ per unit; 0.2 chance for poor sales with -6$ per unit.*
- *Based on the study results, the marketing research institute report: (1) If they produce the first type P2, then P2 has 0.6 chance for good sales with profit 100$ per unit; 0.3 chance for average sales with profit 50$ per unit; 0.1 chance for poor sales with 30$ per unit.*
- **Q:** *which one is better? P1 or P2?*
- *For determining P1 or P2, management decision tree analysis is a commonly used method (see Figure 1).*

- *The main idea is to calculate the so-called* **expected reward** *as follows:*

$$I_1 = 170 \times 0.3 + 90 \times 0.5 - 6 \times 0.2 = 94.8,$$

and

$$I_2 = 100 \times 0.6 + 50 \times 0.3 + 20 \times 0.1 = 77.$$

- *So,* $I_1 > I_2$*, we need to choice P1.*

Example 2 is a signal step decision making problem. What about multiple step decision making problem?

Example 3 *(Markov Decision Processing and Reinforcement Learning)*

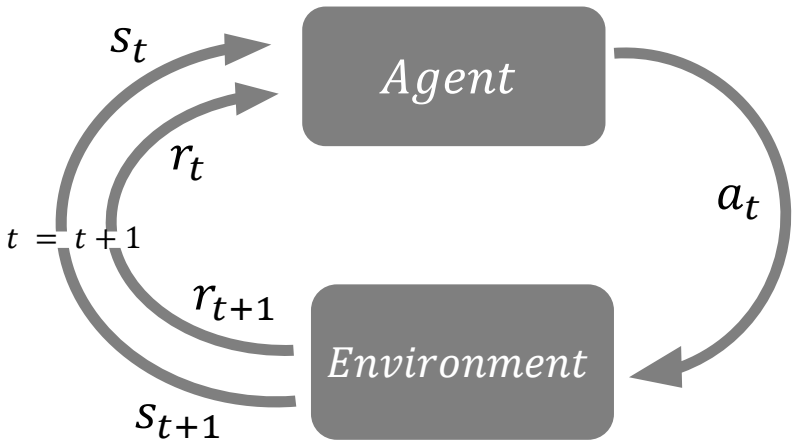

Figure 2: Markov Decision Processing

The above multiple decision making problem (See Figure 2) can be formalized as a Markov Decision Processing (MDP).

- *State Space is considered as a finite state space with cardinality* $||$.
- *Action Space is considered as a finite action space with cardinality* $||$.
- *Transition Probability:*

$$\mathbb{P}(s_{t+1}|a_t, s_t, a_{t-1}, s_{t-1}, \ldots, s_0) = \mathbb{P}(s_{t+1}|a_t, s_t). \tag{1}$$

- *Expected Reward:*

$$\mathbb{E}(r_t|a_t, s_t, a_{t-1}, s_{t-1}, \ldots, s_0) = \mathbb{E}(r_{t+1}|a_t, s_t) = r(a_t, s_t). \tag{2}$$

- *Accumulated Reward:*

$$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r(a_t, s_t) \tag{3}$$

where $\tau = (s_0, a_0, s_1, a_1, \ldots,)$ *is a trajectory (see Figure 3) and* $0 < \gamma < 1$ *is a discount factor.*

- *Policy* $\pi : s \in \rightarrow \Delta()$ *and* $a \sim \pi(a|s)$.
- *Aim: Finding an optimal policy for maximizing the expected accumulated reward.*

Optimization Formulation:

$$\max_{\pi} \mathbf{E}_{\tau \sim \pi}[R(\tau)]. \tag{4}$$

Reinforcement Learning is commonly used method to solve the above optimization.

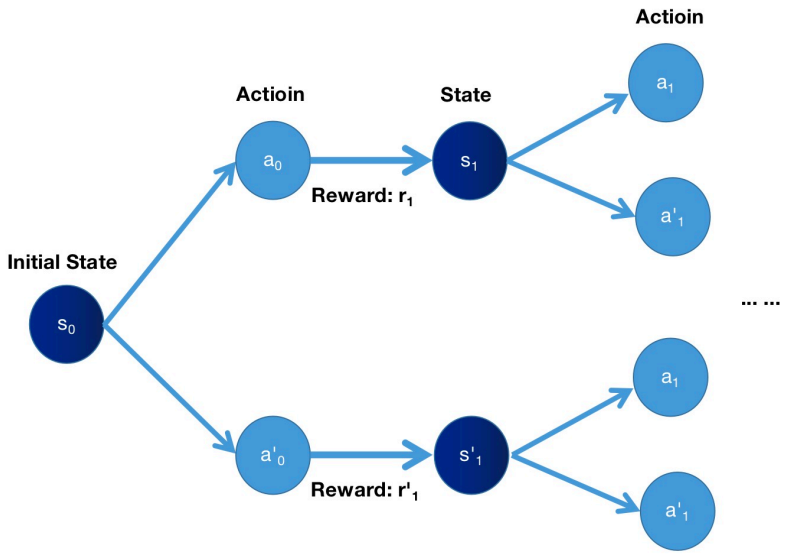

Figure 3: Trajectory of the Markov Decision Processing

1.2 Algorithms in Optimization

Let us consider an optimization problem

$$\min_{x} \ f(\mathbf{x}), \tag{5}$$

$$\text{s.t. } \mathbf{x} \in \mathcal{X}, \tag{6}$$

where $f : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$, and further assume that $\mathbf{x}^*$ is the optimal global point or solution for it. **??**.

An optimization algorithm is to design for pursuing the $\mathbf{x}^*$. However, usually it is not easy.

We consider the least squares problem,

$$\min_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2. \tag{7}$$

Q: How to find x^* for the least squares problem?

Generally, I believe that you should known that to compute the derivative to obtain $f'(\mathbf{x})$ and set $f'(\mathbf{x}) = 0$. Then the solution of $f'(\mathbf{x}) = 0$ maybe the optimal solution of Eq.(7). What does it mean $f'(\mathbf{x})$ for a function defined on $\mathbb{R}^n$?

Definition 1 *A function $f : \mathbb{R}^n \to \mathbb{R}$ is Fŕechet-differential at $\mathbf{x}$, if there exists a vector $g \in \mathbb{R}^n$ such that*

$$\lim_{\Delta\mathbf{x}\to 0} \frac{f(\mathbf{x}+\Delta\mathbf{x}) - f(\mathbf{x}) - g^\top \Delta\mathbf{x}}{\|\Delta\mathbf{x}\|} = 0. \tag{8}$$

Then g is called the gradient of f at $\mathbf{x}$, denoted as $g := \nabla f(\mathbf{x})$. If we further choose that $\Delta\mathbf{x} = \epsilon e_i$, and $e_i = (0, \ldots, \underbrace{1}_{i^{th}\ position}, 0, \ldots, 0)^\top$, then

$$\nabla f(\mathbf{x}) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right)^\top \in \mathbb{R}^n.$$

Definition 2 *We define the Hessian matrix of function $f : \mathbb{R}^n \to \mathbb{R}$ at point $\mathbf{x}$ is*

$$\begin{aligned}\nabla^2 f(\mathbf{x}) &= \left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right) \in \mathbb{R}^{n^2} \\ &= \begin{pmatrix} \frac{\partial^2 f}{\partial^2 x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial^2 x_n} \end{pmatrix}.\end{aligned}$$

Commonly, we assume that the Hessian matrix $\nabla^2 f(\mathbf{x})$ is a symmetric matrix (actually need some regularity conditions).

Definition 3 *Let $f : \mathbb{R}^n \to \mathbb{R}^m$, namely for any $\mathbf{x} \in \mathbb{R}^n$, $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\top \in \mathbb{R}^m$, the Jacobi matrix is denoted as*

$$J(\mathbf{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} \in \mathbb{R}^{m \times n}.$$

Example 4 *$f(\mathbf{x}) = \mathbf{a}^\top \mathbf{x}$, then $\nabla f(\mathbf{x}) = \mathbf{a}, \nabla^2 f(\mathbf{x}) = \mathbf{0} \in \mathbb{R}^{n^2}$, why???*

Example 5 *$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$.*

Let us consider a general case, suppose that $G : \mathbb{R}^m \to \mathbb{R}$ and $G(\mathbf{z}) = g(z_1) + g(z_2) + \cdots + g(z_m)$ and $z_i = \mathbf{a}_i^\top \mathbf{x}, A \in^{m \times n}$, where $\mathbf{z} = (z_1, \ldots, z_m)^\top$. Let us derive that

$$\frac{\partial G(A\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial (g(\mathbf{a}_1^\top \mathbf{x}) + g(\mathbf{a}_2^\top \mathbf{x}) + \cdots + g(\mathbf{a}_m^\top \mathbf{x}))}{\partial \mathbf{x}} \tag{9}$$

$$= \sum_{i=1}^m \frac{\partial g(\mathbf{a}_i^\top \mathbf{x})}{\partial \mathbf{x}} = \sum_{i=1}^m \frac{\partial g(\mathbf{a}_i^\top \mathbf{x})}{\partial \mathbf{a}_i^\top \mathbf{x}} \times \frac{\partial \mathbf{a}_i^\top \mathbf{x}}{\partial \mathbf{x}} \tag{10}$$

$$= \sum_{i=1}^m \frac{\partial g(\mathbf{a}_i^\top \mathbf{x})}{\partial \mathbf{a}_i^\top \mathbf{x}} \mathbf{a}_i \tag{11}$$

$$= A^\top \nabla G(\mathbf{z}). \tag{12}$$

Theorem 1 *(First-order Optimality Condition) Consider a non-constrained optimization problem $\min_{\mathbf{x}} f(\mathbf{x})$, where $f : \mathbb{R}^n \to \mathbb{R}$ and $f \in C^1$. If $\mathbf{x}^*$ is a local minimum, then*

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

The points which satisfy the equation $\nabla f(\mathbf{x}) = 0$ are called stationary points.

Theorem 2 *(Second-order Optimality Condition) Consider a non-constrained optimization problem $\min_{\mathbf{x}} f(\mathbf{x})$, where $f : \mathbb{R}^n \to \mathbb{R}$ and $f \in C^2$. If $\mathbf{x}^*$ is a local minimum, then*

$$\nabla f(\mathbf{x}^*) = 0 \text{ and } \nabla^2 f(\mathbf{x}^*) \geq 0,$$

where $\nabla^2 f(\mathbf{x}^*) \geq 0$ means the Hessian matrix is a positive semi-definite matrix.

Theorem 3 *(Sufficient Condition) Consider a non-constrained optimization problem $\min_{\mathbf{x}} f(\mathbf{x})$, where $f : \mathbb{R}^n \to \mathbb{R}$ and $f \in C^2$. If*

$$\nabla f(\mathbf{x}^*) = 0 \text{ and } \nabla^2 f(\mathbf{x}^*) > 0,$$

where $\nabla^2 f(\mathbf{x}^*) > 0$ means the Hessian matrix is a positive definite matrix. Then $\mathbf{x}^*$ is a local minimum of f.

These proofs can be found at Page 161-163 of the text book.

We go back to this example and further assume that $G(\mathbf{z}) = \frac{1}{2}\|\mathbf{z}-\mathbf{b}\|^2 = \frac{1}{2}\sum_{i=1}^{n}(z_i - b_i)^2, z_i = \mathbf{a}_i^\top \mathbf{x}$. *Thus,* $\nabla G(\mathbf{z}) = (z_1 - b_1, \ldots, z_n - b_n)^\top$. *Finally, based on Eq.(12),*

$$\begin{aligned}\nabla f(\mathbf{x}) = \frac{\partial G(\mathbf{z})}{\partial \mathbf{x}} &= A^\top(\mathbf{z}-\mathbf{b}) \\ &= A^\top(A\mathbf{b}-\mathbf{b}) = A^\top A\mathbf{x} - A\mathbf{b}\end{aligned}$$

and

$$\nabla^2 f(\mathbf{x}) = A^\top A.$$

Recall the least squares problem (7), and set $\nabla f(\mathbf{x}) = \nabla \frac{1}{2}\|A\mathbf{x}-\mathbf{b}\|_2^2 = 0$, then we obtain the so-called *normal equation*:

$$A^\top A\mathbf{x} - A^\top \mathbf{b} = 0. \tag{13}$$

If $A^\top A$ is invertible, then $\mathbf{x}^* = (A^\top A)^{-1}A^\top \mathbf{b}$, which is called *a closed form solution.*

According to the definition of stationary point, we know that $\mathbf{x}^* = (A^\top A)^{-1}A^\top \mathbf{b}$ is a stationary point of the least squares problem. Furthermore, if $\nabla^2 f(\mathbf{x}) = A^\top A$ is a positive definite matrix (invertible), then $\mathbf{x}^* = (A^\top A)^{-1}A^\top \mathbf{b}$ is a local minimum according to Theorem 3.

The procedure of obtaining the closed form solution can be seen as an algorithm for solving the linear least squares problem.

Optimization needs **Iterative Algorithms.** Why???? Let us recall the normal equation (13), and the solution $\mathbf{x}^* = (A^\top A)^{-1}A^\top \mathbf{b}$. Generally, the computational complexity of $(A^\top A)^{-1} \in \mathbb{R}^{n^2}$ is $O(n^3)$. why??????

The iterative algorithm usually has the following general form in Algorithm 1.

Algorithm 1 General Form of Iterative Algorithm

1: **Input:** Something you need
2: **Initialization:** a starting point $\mathbf{x}_0$, and step index $t = 0$
3: **while** a stop condition false **do**
4:

$$\mathbf{x}_{t+1} := \textbf{Iterative Algorithm}(\mathbf{x}_t),$$

and

$$t := t + 1.$$

5: **end while**
6: **Output:** The sequence $\{\mathbf{x}_t\}_{t=0}^{T}$.

Then we hope that $\lim_{t\to\infty} \mathbf{x}_t = \mathbf{x}^*$.

Example 6 *(Solving the Normal Equation)*

Denote that $\tilde{A} = A^\top A$ *and* $\tilde{\mathbf{b}} = A^\top \mathbf{b}$, *then normal equation becomes that* $\tilde{A}\mathbf{x} = \tilde{\mathbf{b}}$. *How to compute it efficiently?*

- *Jacobi Iterative Algorithm: Let* $\tilde{A} = B + D$, *where* $D = diag(\tilde{A})$ *and* $B = \tilde{A} - D$. *Then the normal equation is* $(D+B)\mathbf{x} = \tilde{\mathbf{b}}$. *Thus,* $D\mathbf{x} = -B\mathbf{x} + \tilde{\mathbf{b}}$. *Finally,*

$$\mathbf{x} = -D^{-1}B\mathbf{x} + D^{-1}\tilde{\mathbf{b}}. \tag{14}$$

Based on Eq.(14), Jacobi iterative algorithm is designed via

$$\mathbf{x}_{t+1} = -D^{-1}B\mathbf{x}_t + D^{-1}\tilde{\mathbf{b}}, \tag{15}$$

and the scalar form is

$$x_{t+1,i} = \frac{\tilde{b_i} - \sum_{j=1, j\neq i}^{n} x_{t,j}}{\tilde{a}_{ii}},$$

where we suppose that $\tilde{a}_{ii} \neq 0$ *for all* $i = 1, \ldots, n$.

Insights: *If* $\lim_{t\to\infty} \mathbf{x}_t = \mathbf{x}^*$, *then* $\lim_{t\to\infty} \mathbf{x}_{t+1} = -D^{-1}B \lim_{t\to\infty} \mathbf{x}_t + D^{-1}\tilde{\mathbf{b}}$. *Thus,* $\mathbf{x}^* = -D^{-1}B\mathbf{x}^* + D^{-1}\tilde{\mathbf{b}}$. *This indicates* $\mathbf{x}^*$ *satisfies the normal equation* (13).

- *Gauss-Seidel Algorithm: Let* $\tilde{A} = L + U + D$, *where* $D = diag(\tilde{A})$, L *is the Lower triangular matrix of* $\tilde{A}$ *and* U *is the upper triangular matrix of* $\tilde{A}$. *Then the normal equation is* $(D + L + U)\mathbf{x} = \tilde{\mathbf{b}}$. *Thus,* $D\mathbf{x} = -L\mathbf{x} - U\mathbf{x} + \tilde{\mathbf{b}}$. *Finally,*

$$\mathbf{x} = -D^{-1}L\mathbf{x} - D^{-1}U\mathbf{x} + D^{-1}\tilde{\mathbf{b}}. \tag{16}$$

Based on Eq.(16), *Gauss-seidel iterative algorithm is designed via*

$$\mathbf{x}_{t+1} = -D^{-1}L\mathbf{x}_{t+1} - D^{-1}U\mathbf{x}_t + D^{-1}\tilde{\mathbf{b}}, \tag{17}$$

and the scalar form is

$$x_{t+1,i} = \frac{\tilde{b_i} - \sum_{j=1}^{i-1} \tilde{a}_{ij}x_{t+1,j} - \sum_{j=i+1}^{n} \tilde{a}_{ij}x_{t,j}}{\tilde{a}_{ii}},$$

where we suppose that $\tilde{a}_{ii} \neq 0$ *for all* $i = 1, \ldots, n$.

Insights: *If* $\lim_{t\to\infty} \mathbf{x}_t = \mathbf{x}^*$, *then* $\lim_{t\to\infty} \mathbf{x}_{t+1} = -D^{-1}L \lim_{t\to\infty} \mathbf{x}_{t+1} - D^{-1}U \lim_{t\to\infty} \mathbf{x}_t + D^{-1}\tilde{\mathbf{b}}$. *Thus,* $\mathbf{x}^* = -D^{-1}L\mathbf{x}^* - D^{-1}U\mathbf{x}^* + D^{-1}\tilde{\mathbf{b}}$. *This indicates* $\mathbf{x}^*$ *satisfies the normal equation* (13).

The procedure of obtaining the iterative solution can be seen as an algorithm for solving the linear least squares problem.

Remark 1 *Algorithms in optimization can be commonly summarized as three types, but it's not limited to these.*

- *Closed Form Solution, see* (13).
- *Iterative Algorithm, see Algorithm 1.*
- *Heuristic Algorithms, which will not be covered by the course.*

1.3 Related Theory in Optimization

"Nothing is more practical than a good theory."– by V. Vapnik [Vapnik, 1998].

What kind of theory we have to learn in Optimization?

- Theory can support you to construct models. You have see them in many examples (e.g., MLE).
- Theory can help you develop algorithms. For example, convex analysis, KTT conditions, duality theory, optimally conditions, and among others.
- Theory can implicitly show the convergence property of the optimization algorithms. Convergence theory is to show that under what conditions the sequences $\{\mathbf{x}_t\}_{t=1}^{\infty}$ and $\{f(\mathbf{x}_t)\}_{t=1}^{\infty}$ satisfy

$$\lim_{t\to\infty} \mathbf{x}_t = \mathbf{x}^* \text{ and } \lim_{t\to\infty} f(\mathbf{x}_t) = f^* = f(\mathbf{x}^*).$$

Convergence Rate:

- linear convergence:

$$\frac{\|\mathbf{x}_{t+1} - \mathbf{x}^*\|}{\|\mathbf{x}_t - \mathbf{x}^*\|} \leq a,$$

  where $a \in (0, 1)$.

- Super-linear convergence:

$$\lim_{t\to\infty} \frac{\|\mathbf{x}_{t+1} - \mathbf{x}^*\|}{\|\mathbf{x}_t - \mathbf{x}^*\|} = 0.$$

- sub-linear convergence:

$$\lim_{t\to\infty} \frac{\|\mathbf{x}_{t+1} - \mathbf{x}^*\|}{\|\mathbf{x}_t - \mathbf{x}^*\|} = 1.$$

- Others theoretical bounds:

$$\|\mathbf{x}_t - \mathbf{x}^*\| \leq O(t, Q),$$

  and

$$\|f(\mathbf{x}_t) - f^*\| \leq O(t, Q),$$

  where Q includes some constants related to the original optimization problem.

We justify the convergence theory of Jacobi and Gauss-Seidel algorithms for demonstrating an concrete example.

Theorem 4 *Suppose that we have the linear equation with form $\mathbf{x} = B\mathbf{x} + C$, then we can develop an iterative algorithm*

$$\mathbf{x}_{t+1} = B\mathbf{x}_t + C. \tag{18}$$

For any initial point $\mathbf{x}_0$, the generated sequence $\{\mathbf{x}_t\}_{t=0}^{\infty}$ converges at $\mathbf{x}^*$ if and only if $\rho(B) := \lambda_{\max}(B) < 1$, where $\lambda_{\max}(B)$ is the biggest eigenvalue of B and $\rho(B)$ is so-called spectral radius of B. (In fact, $\rho(B) := \sqrt{\lambda_{\max}(B^\top B)}$), and when B is a normal matrix, $\rho(B)$ is also equal to $\lambda_{\max}(B)$)

1.4 Gradient Descent

Let us consider a unconstrained optimization problem

$$\min_{\mathbf{x}} f(\mathbf{x}) \tag{19}$$

where $\mathbf{x} \in dom(f) \subseteq \mathbb{R}^n$, f is a continuous and F-differential function, i.e. $f \in C^1$.

Basic Idea: The algorithm we need is

$$\mathbf{x}^{t+1} = \mathbf{x}^t + s\mathbf{d}, \text{ such that } f(\mathbf{x}^{t+1}) \leq f(\mathbf{x}^t),$$

where $\mathbf{d} \in \mathbb{R}^n$ is a *descent direction* and $s \in \mathbb{R}$ is referred as to the *step size* of the descent algorithm. **Note that s is also called learning rate in the machine learning or deep learning community.**

Given the descent algorithm, we need to determinate that

- How to choose the descent direction?
- How to choose the step size?

Insights: According to the Taylor expansion, we have that

$$f(\mathbf{x}^{t+1}) = f(\mathbf{x}^t) + \langle \nabla f(\mathbf{x}^t), \mathbf{x}^{t+1} - \mathbf{x}^t \rangle + o(\|\mathbf{x}^{t+1} - \mathbf{x}^t\|), \tag{20}$$

where $\lim_{\mathbf{x}^{t+1} \to \mathbf{x}^t} \frac{o(\|\mathbf{x}^{t+1} - \mathbf{x}^t\|)}{\|\mathbf{x}^{t+1} - \mathbf{x}^t\|} = 0$. You can review the little "o" notation by yourself. Furthermore,

$$f(\mathbf{x}^{t+1}) = f(\mathbf{x}^t) + s\langle \nabla f(\mathbf{x}^t), \mathbf{d} \rangle + o(s\|\mathbf{d}\|). \tag{21}$$

Q: Could you please guess a descent direction?

Let $\mathbf{d} = -\nabla f(\mathbf{x}^t)$, then

$$f(\mathbf{x}^{t+1}) = f(\mathbf{x}^t) - s\|\nabla f(\mathbf{x}^t)\|^2 + o(s\|\nabla f(\mathbf{x}^t)\|) \approx f(\mathbf{x}^t) - s\|\nabla f(\mathbf{x}^t)\|^2 \leq f(\mathbf{x}^t), \tag{22}$$

when s is "small enough".

The iterative algorithm choosing the descent direction $\mathbf{d} = -\nabla f(\mathbf{x}^t)$ is referred to as Gradient Descent Method. The remaining question is to find the proper step size s.

The first method is the *Exact Line Search*:

$$s_t = \arg\min_{s \in \mathbb{R}} f(\mathbf{x}^t - s \cdot \nabla f(\mathbf{x}^t)). \tag{23}$$

The second method is the *Backtracking Line Search.*

Up to now, we have learned the gradient descent algorithm with linear search. The advantage of the algorithm is the simple interpretation. However, the linear search step involved GD makes more computational effort to find a proper step size. This also leads to difficulties in theoretical analysis (See Page 222).

Q: Whether exists a method to provide a proper step size s which can guarantee the convergence of the gradient descent algorithm without line search.

The answer is **Yes!** for the specific objective function.

Definition 4 *$f : \mathbb{R}^n \to \mathbb{R}$ is a β-smooth function if*

- *∇f exists which is continuous.*
- *For any $\mathbf{x}_1, \mathbf{x}_2 \in dom(f)$,*
$$\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\|_2 \leq \beta \|\mathbf{x}_1 - \mathbf{x}_2\|_2. \tag{24}$$
This means ∇f is a β-Lipshitz continuous function.

Let us show some examples:

- $f(\mathbf{x}) = \langle \mathbf{b}, A\mathbf{x} \rangle$ is a 0-smooth function.
- $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{b} - A\mathbf{x}\|^2$ is a $\lambda_{\max}(A^\top A)$-smooth function.

Theorem 5 *Suppose that $\{\mathbf{x}^t\}_{t=0}^{\infty}$ is generated by GD and the given tolerance $\epsilon > 0$, if $T \geq \frac{2\beta(f(\mathbf{x}_0) - f^*)}{\epsilon^2}$, then*

$$\min_{t=0,1,\ldots,T-1} \|\nabla f(\mathbf{x}^t)\| \leq \epsilon. \tag{25}$$

Theorem 6 *Let f be a convex and β-smooth function, and $\{\mathbf{x}^t\}_{t=0}^{\infty}$ is generated by GD. Then for any $\epsilon > 0$, take $T \geq \frac{\beta}{\epsilon}\|\mathbf{x}^0 - \mathbf{x}^*\|^2$,*

$$f(\mathbf{x}^T) - f^* \leq \epsilon. \tag{26}$$

Theorem 7 *Assume that f is a β-smooth and α-strongly convex function, and $f^* = \inf f(\mathbf{x})$ exists, $\{\mathbf{x}^t\}_{t=0}^{\infty}$ generated by GD, then for any $\epsilon > 0$, choose $T \geq \frac{2\beta}{\alpha} \log \frac{\|\mathbf{x}^0 - \mathbf{x}^*\|}{\epsilon}$, it has $\|\mathbf{x}^T - \mathbf{x}^*\| \leq \epsilon$.*

1.5 Summary

Optimality Conditions:

- Necessary: $\nabla f(\mathbf{x}^*) = 0$.
- Necessary: $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*) \succeq 0$
- Sufficient: $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*) > 0$

Table 1: Convergence Theory

	β-smooth	+ Convex	$+\alpha$-strong Convex
$\min_{1\le t\le T} \|\nabla f(\mathbf{x}^t)\|$	$O(1/\sqrt{T})$	$O(1/T)$	NA
$f(\mathbf{x}^T) - f(\mathbf{x}^*)$	NA	$O(1/T)$	$\frac{\beta}{2}\exp(-\frac{\alpha}{\beta}T)\|\mathbf{x}^0 - \mathbf{x}^*\|^2$
$\|\mathbf{x}^T - \mathbf{x}^*\|^2$	NA	NA	$\exp(-\frac{\alpha}{\beta}T)\|\mathbf{x}^0 - \mathbf{x}^*\|^2$

References

[Vapnik, 1998] Vapnik, V. (1998). *Statistical learning theory.* John Wiley, New York.