Optimization

Yuh-Jye Lee

Data Science and Machine Intelligence Lab
National Chiao Tung University

March 21, 2017

You Have Learned (Unconstrained) Optimization in Your High School

Let $f(x) = ax^2 + bx + c$, $a \neq 0$, $x^* = -\frac{b}{2a}$

$$\text{Case 1}: \; f''(x^*) = 2a > 0 \Rightarrow x^* \in \arg\min_{x \in \mathbb{R}} f(x)$$

$$\text{Case 2}: \; f''(x^*) = 2a < 0 \Rightarrow x^* \in \arg\max_{x \in \mathbb{R}} f(x)$$

For minimization problem (Case I),

- $f'(x^*) = 0$ is called the first order optimality condition.
- $f''(x^*) > 0$ is the second order optimality condition.

Optimization Examples in Machine Learning

1. Maximum likelihood estimation
2. Maximum a posteriori estimation
3. Least squares estimates
4. Gradient descent method
5. Backpropagation

Gradient and Hessian

- Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function. The gradient of function f at a point $x \in \mathbb{R}^n$ is defined as

$$\nabla f(x) = [\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n}] \in \mathbb{R}^n$$

- If $f : \mathbb{R}^n \to \mathbb{R}$ is a twice differentiable function. The Hessian matrix of f at a point $x \in \mathbb{R}^n$ is defined as

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Example of Gradient and Hessian

$$
\begin{aligned}
f(x) &= x_1^2 + x_2^2 - 2x_1 + 4x_2 \\
&= \frac{1}{2}\begin{bmatrix} x_1 & x_2 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -2 & 4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}
\end{aligned}
$$

$$
\nabla f(x) = \begin{bmatrix} 2x_1 - 2 & 2x_2 + 4 \end{bmatrix}, \nabla^2 f(x) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}
$$

By letting $\nabla f(x) = 0$, we have $x^* = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \in \arg\min_{x \in \mathbb{R}^2} f(x)$

Quadratic Functions (Standard Form) $f(x) = \frac{1}{2}x^\top Hx + p^\top x$

Let $f : \mathbb{R}^n \to \mathbb{R}$ and $f(x) = \frac{1}{2}x^\top Hx + p^\top x$
where $H \in \mathbb{R}^{n\times n}$ is a symmetric matrix and $p \in \mathbb{R}^n$
then

$$\nabla f(x) = Hx + p$$

$$\nabla^2 f(x) = H \text{ (Hessian)}$$

Note: If H is positive definite, then $x^* = -H^{-1}p$ is the unique solution of $\min f(x)$.

Least-squares Problem

$\min_{x\in\mathbb{R}^n} \|Ax - b\|_2^2,\ A \in \mathbb{R}^{m\times n},\ b \in \mathbb{R}^m$

$$
\begin{aligned}
f(x) &= (Ax - b)^\top (Ax - b) \\
&= x^\top A^\top Ax - 2b^\top Ax + b^\top b \\
\nabla f(x) &= 2A^\top Ax - 2A^\top b \\
\nabla^2 f(x) &= 2A^\top A \\
x^* &= (A^\top A)^{-1} A^\top b \in \arg\min_{x\in\mathbb{R}^n} \|Ax - b\|_2^2
\end{aligned}
$$

If $A^\top A$ is nonsingular matrix $\Rightarrow$ P.D.
Note : x^* is an analytical solution.

How to Solve an Unconstrained MP

- Get an initial point and iteratively decrease the obj. function value.
- Stop once the stopping criteria satisfied.
- Steep decent might not be a good choice.
- Newtons method is highly recommended.
  - Local and quadratic convergent algorithm.
  - Need to choose a good step size to guarantee global convergence.

The First Order Taylor Expansion

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a differentiable function

$$f(x + d) = f(x) + \nabla f(x)^\top d + \alpha(x, d)\|d\|,$$

where

$$\lim_{d \to 0} \alpha(x, d) = 0$$

If $\nabla f(x)^\top d < 0$ and d is small enough then $f(x + d) < f(x)$.

We call d is a descent direction.

Steep Descent with Exact Line Search

Start with any $x^0 \in \mathbb{R}^n$. Having x^i, stop if $\nabla f(x^i) = 0$.
Else compute x^{i+1} as follows:

1. Steep descent direction: $d^i = -\nabla f(x^i)$
2. Exact line search: Choose a stepsize such that

$$\frac{df(x^i + \lambda d^i)}{d\lambda} = f'(x^i + \lambda d^i) = 0$$

3. Updating: $x^{i+1} = x^i + \lambda d^i$

MATLAB Code for Steep Descent with Exact Line Search (Quadratic Function Only)

function $[x, f\_value, iter] = grdlines(Q, p, x0, esp)$
%
% min $0.5 * x^\top Qx + p^\top x$
% *Solving unconstrained minimization via*
% *steep descent with exact line search*
%

```
flag = 1;
iter = 0;
while flag > esp
    grad = Qx0+p;
    temp1 = grad'*grad;
    if temp1 < 10^-12
        flag = esp;
    else
        stepsize = temp1/(grad'*Q*grad);
        x1 = x0 - stepsize*grad;
        flag = norm(x1-x0);
        x0 = x1;
    end;
        iter = iter + 1;
end;
x = x0;
f_value = 0.5*x'*Q*x+p'*x;
```

The Key Idea of Newton's Method

Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be a twice differentiable function

$$f(x+d) = f(x) + \nabla f(x)^\top d + \frac{1}{2} d^\top \nabla^2 f(x) d + \beta(x,d) \parallel d \parallel$$

where $\lim_{d \to 0} \beta(x,d) = 0$

At i^{th} iteration, use a quadratic function to approximate

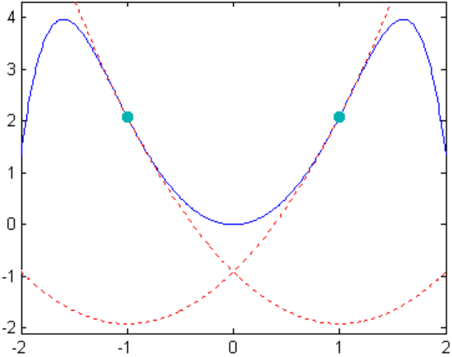
$$f(x) \approx f(x^i) + \nabla f(x^i)(x - x^i) + \frac{1}{2}(x - x^i)^\top \nabla^2 f(x^i)(x - x^i)$$

$x^{i+1} = \arg\min \tilde{f}(x)$

Newton's Method

Start with $x^0 \in \mathbb{R}^n$. Having x^i ,stop if $\nabla f(x^i) = 0$
Else compute x^{i+1} as follows:

1. Newton direction: $\nabla^2 f(x^i) d^i = -\nabla f(x^i)$
   Have to solve a system of linear equations here!
2. Updating: $x^{i+1} = x^i + d^i$
   - Converge only when x^0 is close to x^* enough.



$f(x) = \frac{1}{6}x^6 + \frac{1}{4}x^4 + 2x^2$
$g(x) = f(x^i) + f'(x^i)(x - x^i) + \frac{1}{2}f''(x^i)(x - x^i)^2$
It can not converge to the optimal solution.

Constrained Optimization Problem

Problem setting: Given function f, g_i, $i = 1, ..., k$ and h_j, $j = 1, ..., m$, defined on a domain $\Omega \subseteq \mathbb{R}^n$,

$$\begin{aligned} \min_{x \in \Omega} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, \quad \forall i \\ & h_j(x) = 0, \quad \forall j \end{aligned}$$

where $f(x)$ is called the objective function and $g(x) \leq 0$, $h(x) = 0$ are called constrains.

Example

$$\begin{aligned} \min \quad & f(x) = 2x_1^2 + x_2^2 + 3x_3^2 \\ \text{s.t.} \quad & 2x_1 - 3x_2 + 4x_3 = 49 \end{aligned}$$

<sol>

$$L(x,\beta) = f(x) + \beta(2x_1 - 3x_2 + 4x_3 - 49),\ \beta \in \mathbb{R}$$

$$\begin{aligned} \frac{\partial}{\partial x_1} L(x,\beta) = 0 \quad & \Rightarrow \quad 4x_1 + 2\beta = 0 \\ \frac{\partial}{\partial x_2} L(x,\beta) = 0 \quad & \Rightarrow \quad 2x_2 - 3\beta = 0 \\ \frac{\partial}{\partial x_3} L(x,\beta) = 0 \quad & \Rightarrow \quad 6x_3 + 4\beta = 0 \end{aligned}$$

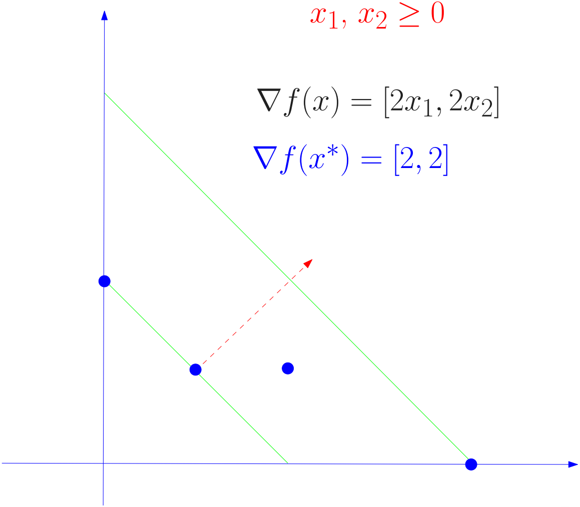
$$2x_1 - 3x_2 + 4x_3 - 49 = 0 \Rightarrow \beta = -6$$

$$\Rightarrow x_1 = 3,\ x_2 = -9,\ x_3 = 4$$

$$\min_{x \in R^2} x_1^2 + x_2^2 \qquad \begin{aligned} x_1 + x_2 &\leq 4 \\ -x_1 - x_2 &\leq -2 \\ x_1,\, x_2 &\geq 0 \end{aligned}$$

$$\nabla f(x) = [2x_1, 2x_2]$$

$$\nabla f(x^*) = [2, 2]$$



Definitions and Notation

- Feasible region:

$$\mathcal{F} = \{x \in \Omega \mid g(x) \leq 0, h(x) = 0\}$$

where $g(x) = \begin{bmatrix} g_1(x) \\ \vdots \\ g_k(x) \end{bmatrix}$ and $h(x) = \begin{bmatrix} h_1(x) \\ \vdots \\ h_m(x) \end{bmatrix}$

- A solution of the optimization problem is a point $x^* \in \mathcal{F}$ such that $\nexists x \in \mathcal{F}$ for which $f(x) < f(x^*)$ and x^* is called a global minimum.

Definitions and Notation

- A point $\bar{x} \in \mathcal{F}$ is called a local minimum of the optimization problem if $\exists \varepsilon > 0$ such that

$$f(x) \geq f(\bar{x}), \qquad \forall x \in \mathcal{F} \text{ and } \|x - \bar{x}\| < \varepsilon$$

- At the solution x^*, an inequality constraint $g_i(x)$ is said to be active if $g_i(x^*) = 0$, otherwise it is called an inactive constraint.
- $g_i(x) \leq 0 \Leftrightarrow g_i(x) + \xi_i = 0$, $\xi_i \geq 0$ where ξ_i is called the slack variable

Definitions and Notation

- Remove an inactive constraint in an optimization problem will NOT affect the optimal solution
  - Very useful feature in SVM
- If $\mathcal{F} = \mathbb{R}^n$ then the problem is called unconstrained minimization problem
  - Least square problem is in this category
  - SSVM formulation is in this category
  - Difficult to find the global minimum without convexity assumption

The Most Important Concepts in Optimization(minimization)

- A point is said to be an *optimal solution* of a unconstrained minimization if there exists no decent direction
  $\Longrightarrow \nabla f(x^*) = 0$
- A point is said to be an optimal solution of a constrained minimization if there exists no feasible decent direction
  $\Longrightarrow$ KKT conditions
  - There might exist decent direction but move along this direction will leave out the feasible region

Minimum Principle

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex and differentiable function $\mathcal{F} \subseteq \mathbb{R}^n$ be the feasible region.

$$x^* \in \arg\min_{x \in \mathcal{F}} f(x) \iff \nabla f(x^*)(x - x^*) \geq 0 \quad \forall x \in \mathcal{F}$$

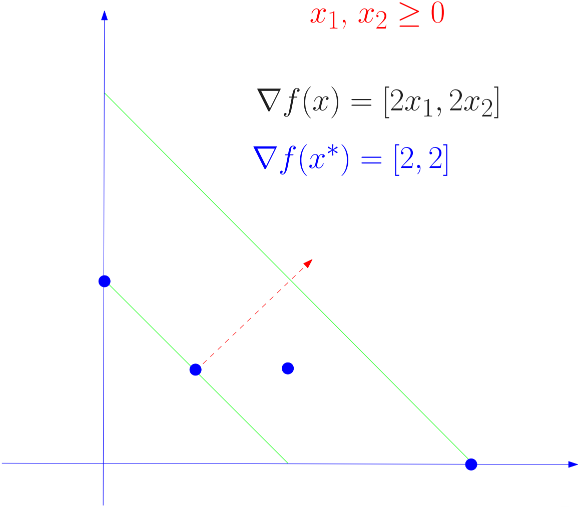
Example:

$$\min(x-1)^2 \quad \text{s.t.} \quad a \leq x \leq b$$

$$\min_{x \in R^2} x_1^2 + x_2^2 \qquad \begin{aligned} x_1 + x_2 &\leq 4 \\ -x_1 - x_2 &\leq -2 \\ x_1,\, x_2 &\geq 0 \end{aligned}$$

$$\nabla f(x) = [2x_1, 2x_2]$$

$$\nabla f(x^*) = [2, 2]$$



Linear Programming Problem

- An optimization problem in which the objective function and all constraints are linear functions is called a linear programming problem

$$
\begin{array}{lll}
\text{(LP)} & \min & p^\top x \\
 & \text{s.t.} & Ax \leq b \\
 & & Cx = d \\
 & & L \leq x \leq U
\end{array}
$$

Linear Programming Solver in MATLAB

X=LINPROG(f,A,b) attempts to solve the linear programming problem:

$$\min_{x} \quad f'*x \quad \text{subject to: } A*x <= b$$

X=LINPROG(f,A,b,Aeq,beq) solves the problem above while additionally satisfying the equality constraints Aeq*x = beq.

X=LINPROG(f,A,b,Aeq,beq,LB,UB) defines a set of lower and upper bounds on the design variables, X, so that the solution is in the range LB <= X <= UB.
Use empty matrices for LB and UB if no bounds exist. Set LB(i) = -Inf if X(i) is unbounded below; set UB(i) = Inf if X(i) is unbounded above.

X=LINPROG(f,A,b,Aeq,beq,LB,UB,X0) sets the starting point to X0. This option is only available with the active-set algorithm. The default interior point algorithm will ignore any non-empty starting point.

You can type “help linprog” in MATLAB to get more information!

L_1-Approximation: $\min_{x \in \mathbb{R}^n} \|Ax - b\|_1$

$$\|z\|_1 = \sum_{i=1}^{m} |z_i|$$

$$\min_{x,s} \mathbf{1}^\top s$$

$$\text{s.t. } -s \le Ax - b \le s$$

Or

$$\min_{x,s} \sum_{i=1}^{m} s_i$$

$$\text{s.t. } -s_i \le A_i x - b_i \le s_i \quad \forall i$$

$$\min_{x,s} \begin{bmatrix} 0 & \cdots & 0 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} x \\ s \end{bmatrix}$$

$$\text{s.t. } \begin{bmatrix} A & -I \\ -A & -I \end{bmatrix}_{2m \times (n+m)} \begin{bmatrix} x \\ s \end{bmatrix} \le \begin{bmatrix} b \\ -b \end{bmatrix}$$

Chebyshev Approximation: $\min_{x \in \mathbb{R}^n} \|Ax - b\|_\infty$

$$\|z\|_\infty = \max_{1 \leq i \leq m} |z_i|$$

$$\begin{aligned} \min_{x,\gamma} \quad & \gamma \\ \text{s.t.} \quad & -\mathbf{1}\gamma \leq Ax - b \leq \mathbf{1}\gamma \end{aligned}$$

$$\begin{aligned} \min_{x,s} \; & \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \gamma \end{bmatrix} \\ \text{s.t.} \; & \begin{bmatrix} A & -\mathbf{1} \\ -A & -\mathbf{1} \end{bmatrix}_{2m \times (n+1)} \begin{bmatrix} x \\ \gamma \end{bmatrix} \leq \begin{bmatrix} b \\ -b \end{bmatrix} \end{aligned}$$