Optimization 1 — Tutorial 4

November 12, 2020

Iterative Methods

Algorithm 1 Gradient Descent Method

Input: continuously differentiable function, $\varepsilon > 0$ and $\mathbf{x}^0 \in \mathbb{R}^n$.
Steps: repeat the following process until a stopping criterion is satisfied:

- Choose a step size $t_k > 0$.
- Update $\mathbf{x}^{k+1} := \mathbf{x}^k - t_k \nabla f\left(\mathbf{x}^k\right)$.

Output: $\mathbf{x}^k$.

Algorithm 2 Newton's Method

Input: twice continuously differentiable function, $\varepsilon > 0$ and $\mathbf{x}^0 \in \mathbb{R}^n$.
Steps: repeat the following process until a stopping criterion is satisfied:

- Solve $\nabla^2 f\left(\mathbf{x}^k\right)\mathbf{d}^k = -\nabla f\left(\mathbf{x}^k\right)$.
- Choose a step size $t_k > 0$.
- Update $\mathbf{x}^{k+1} := \mathbf{x}^k + t_k \mathbf{d}^k$.

Output: $\mathbf{x}^k$.

MATLAB implementation of the two methods:

```
function x=gradient_constant(grad_f,x0,t,epsilon)
x=x0; g=grad_f(x);
while norm(g)>epsilon
        x=x-t*g;
        g=grad_f(x);
end

function x=pure_newton(grad_f,hess_f,x0,epsilon)
x=x0; g=grad_f(x); h=hess_f(x); iter=0;
while norm(g)>epsilon && iter<10000
        iter=iter+1;
        x=x-h\g;
        g=grad_f(x);
        h=hess_f(x);
end
```

Problem 1

Let $f\colon \mathbb{R}^n \to \mathbb{R}$ be the function defined by

$$f(x,y) = x^2 + xy + y^2 = \begin{pmatrix} x \\ y \end{pmatrix}^T \begin{pmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

(a) Find the strict global minimum point of f.

(b) Compute one iteration of the Gradient Descent method using fixed step size $t_k \equiv \frac{1}{L_{\nabla f}}$, exact line search and backtracking with $(s, \alpha, \beta) = \left(1, \frac{1}{10}, \frac{1}{4}\right)$. Assume that $(1, 1)$ is the starting point for all three methods. Write your calculations in detail. Did the methods converge to the global minimum point after one iteration?

Problem 2

In the Fermat-Weber problem we aim at solving the following minimization problem

$$\min_{\mathbf{x} \in \mathbb{R}^n} \left\{ f(\mathbf{x}) = \sum_{i=1}^{m} \omega_i \|\mathbf{x}_i - \mathbf{a}_i\| \right\}.$$

Weiszfeld's method is the following fixed-point method for finding a stationary point of f (assuming all anchors are not stationary points):

$$\mathbf{x}^{k+1} = \frac{1}{\sum_{i=1}^{m} \frac{\omega_i}{\|\mathbf{x}_i^k - \mathbf{a}_i\|}} \sum_{i=1}^{m} \frac{\omega_i \mathbf{a}_i}{\|\mathbf{x}_i^k - \mathbf{a}_i\|}.$$

Show that the method is actually Gradient Descent method and find the step size t_k for all $k \geq 0$.

Cholesky Factorization

In order for the Newton's method to generate a well-defined descent sequence, we need $\nabla^2 f\left(\mathbf{x}^k\right)$ to be PD. The Cholesky factorization is a relatively numerically stable method that checks whether a matrix is PD.

Let $\mathbf{A} \in \mathbb{R}^{n \times n}$. Then $\mathbf{A} \succ 0$ if and only if there exists a lower triangular matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$ such that $\mathbf{A} = \mathbf{L}\mathbf{L}^T$ (and it is called the Cholesky factorization of $\mathbf{A}$).

Problem 3

1. Given a Cholesky factorization of $\mathbf{A}$, show how to solve the system $\mathbf{A}\mathbf{x} = \mathbf{b}$.
2. Show how to attain a Cholesky factorization of a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$.

Hybrid Method

If $f \colon \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function that is bounded from below, then by a proper selection of the sequence $\{t_k\}_{k \geq 0}$ we can show that for Gradient Descent, the sequence $\left\{\mathbf{x}^k\right\}_{k \geq 0}$ converges to a stationary point of f. The rate of convergence is relatively slow (sub-linear). However, Newton's method has a better convergence rate (quadratic) in the vicinity of a stationary point, but the Hessian is not necessarily PD in each iteration. The Hybrid Method incorporates the advantages of each method.

Algorithm 3 Hybrid Method

Input: a continuously differentiable function, $\varepsilon > 0$ and $\mathbf{x}^0 \in \mathbb{R}^n$.
Steps: repeat the following process until a stopping criterion is satisfied:

- If $\nabla^2 f\left(\mathbf{x}^k\right) \succ 0$, perform Netown's method update.
- Otherwise, perform Gradient Descent update.

Output: $\mathbf{x}^k$.

MATLAB implementation of the Hybrid Method using the Cholseky factorization and constant step size:

```
function x=hybrid(grad_f,hess_f,x0,t,epsilon)
x=x0; g=grad_f(x); h=hess_f(x);
while norm(g)>epsilon
        [L,p]=chol(h,'lower');
        if p==0
                d=L'\(L\g);
        else
                d=g;
        end
        x=x-t*d;
        g=grad_f(x);
        h=hess_f(x);
end
```