STA 141C - Big Data & High Performance Statistical Computing **Spring 2022**

Lecture 12: Newton's method

Lecturer: Bo Y.-C. Ning *February 15, 2022*

Disclaimer: *My notes may contain errors, distribution outside this class is allowed only with the permission of the Instructor.*

Announcement

- HW1 graded: Sec 1. mean 10.84/15, sd 3.6; Sec 2. mean 10.71/15, sd 3.87
- Four students received extra credit (recommended by the reader)
- HW2 due this week
- HW3 due in one week
- Grade will be curved, no need to worry too much.

Last time

- Singular value decomposition
- Power method

Today

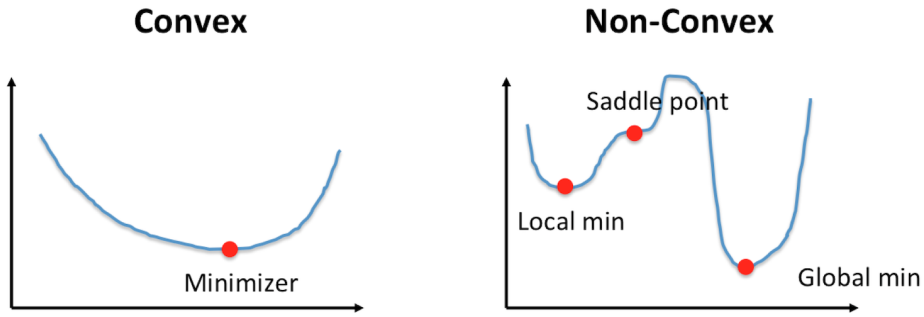
- Newton's algorithm

1 Introduction

Our goal is to minimize a function $\min_\theta f(\theta)$, and let's assume the function f is twice differentiable. If f is a convex function, then $f'(\theta^\star) = 0$ implies that $\theta^\star$ is the global minimum. If f if s non-convex function, then $f'(\theta^\star) = 0$ implies that $\theta^\star$ can be the global minimum, local minimum, or a saddle point.

Newton's method was originally developed for finding roots of nonlinear equations $f(\theta) = 0$, which is also known as the Newton-Raphson method. It is considered as the gold standard for its fast (quadratic) convergence. The idea is to apply iterative quadratic approximation until

$$\frac{\|\theta^{(t+1)} - \theta^\star\|}{\|\theta^{(t)} - \theta^\star\|^2} \to \text{a constant}$$



To understand Newton's method, we first recall the Taylor expansion to approximate a function $f(x)$ around a real number a:

$$f(\theta) = f(a) + \frac{f'(a)}{1}(x-a) + \frac{f''(a)}{2}(x-a)^2 + ...$$

- $f(x) = \dfrac{1}{1-x}$

$$f(x) = 1 + x + x^2 + x^3 + ...$$

- $f(x) = e^x$

$$f(x) = 1 + x + x^2 + ...$$

- $f(x) = \log(1-x)$ for $|x| < 1$

$$f(x) = -x + \frac{x^2}{2} + \frac{x^3}{3} + ...$$

- $\theta \in \mathbb{R}^p$:

$$f(\theta) = f(a) + (\theta - a)' f'(a) + \frac{1}{2}(\theta - a)' f''(a)(\theta - a) + ...$$

The idea of Newton's method is to apply Taylor expansion around the current iterate $\theta^{(t)}$ at iteration t,

$$L(\theta) = L(\theta^{(t)}) + dL(\theta^{(t)})(\theta - \theta^{(t)}) + \frac{(\theta - \theta^{(t)})' d^2 L(\theta^{(t)})(\theta - \theta^{(t)})}{2}.$$

One wants to maximize the right hand size by equating its gradient

$$-dL(\theta^{(t)}) - d^2 L(\theta^{(t)})(\theta - \theta^{(t)}) = 0$$

This implies a solution for θ:

$$\theta = \theta^{(t)} - [d^2 L(\theta^{(t)})]^{-1} dL(\theta^{(t)})$$

In statistics models, $-d^2 L(\theta^{(t)})$ is often the fisher information matrix, thus we instead write

$$\theta = \theta^{(t)} + [-d^2 L(\theta^{(t)})]^{-1} dL(\theta^{(t)}).$$

Set $\theta^{(t+1)} = \theta$ and run the algorithm T many times until

$$\|\theta^{(t+1)} - \theta^{(t)}\| \leq \epsilon$$

2 Applications

1. MLE for Poisson distribution

Suppose $x_1, \ldots, x_n \sim \text{Poisson}(\lambda)$.

- Likelihood function: $f(x^n|\lambda) = \prod_{i=1}^n f(x_i|\lambda) = \prod_{i=1}^n \left[\lambda^{x_i} e^{-\lambda}/x_i!\right]$
- Log-likelihood function:

$$L(\lambda|x^n) = \log f(x^n|\lambda) = \sum_{i=1}^n x_i \log\lambda - n\lambda - \sum_{i=1}^n \log(x_i!)$$

- $dL(\lambda^{(t)}) = \frac{\sum_{i=1}^n x_i}{\lambda^{(t)}} - n$
- $d^2L(\lambda^{(t)}) = -\frac{\sum_{i=1}^n x_i}{{\lambda^{(t)}}^2}$
- Update $\lambda = \lambda^{(t)} + \left(\frac{\sum_{i=1}^n x_i}{{\lambda^{(t)}}^2}\right)^{-1}\left(\frac{\sum_{i=1}^n x_i}{\lambda^{(t)}} - n\right) = 2\lambda^{(t)} - \frac{n{\lambda^{(t)}}^2}{\sum_{i=1}^n x_i}$.

2. Multivariate normal distribution

For a Gaussian model with a known variance $\Sigma = I_p$, $X_1, \ldots, X_n \sim N(\theta, I_p)$,

- Likelihood function: $f(X^n|\theta) = \prod_{i=1}^n f(X_i|\theta) = \prod_{i=1}^n \left[\left(\frac{1}{\sqrt{2\pi}}\right)^p \exp\left(-\frac{1}{2}(X_i-\theta)'(X_i-\theta)\right)\right]$
- $f(X^n|\theta) = \left(\frac{1}{\sqrt{2\pi}}\right)^{np} \exp\left(-\frac{1}{2}\sum_{i=1}^n (X_i-\theta)'(X_i-\theta)\right)$
- Log-likelihood function:

$$L(\theta \mid X^n) = \log f(X^n \mid \theta) = -\frac{np}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^n (X_i-\theta)'(X_i-\theta).$$

- What is $d^2L(\theta^{(t)})$ and $dL(\theta^{(t)})$.
- $\theta = \theta^{(t)} + \sum_{i=1}^n (X_i - \theta^{(t)})/n$

Issues with Newton's method:

- 1. Need to derive, evaluate, and invert the information matrix
  - Exploit the structure in Hessian whenever possible
- 2. Bad starting points
- 3. Stability: Newton's iteration is not guaranteed to be an ascent algorithm.
  - Approximate $-d^2L(\theta^t)$ by a positive definition matrix A (if not)
  - Line search (backtracking)

```r
rm(list = ls())

# simulate a dataset
p <- 10
n <- 100
theta_0 <- rep(1, p)
set.seed(2022)
library(mvtnorm)
x <- rmvnorm(n, theta_0, diag(p))

# MLE for theta
theta.mle <- colMeans(x)

# Newton's method for theta
newton_mvnorm <- function(x, theta.int = NULL, total = 100, epsilon = 1e-6) {

  n <- dim(x)[1]
  p <- dim(x)[2]
  theta.val <- matrix(0, p, total)
  diff.val <- rep(0, total)

  if (n < p) {
    warnings("sample size n should be larger than p!")
  }

  if (is.null(theta.int) == T) {
    theta.int <- rnorm(p)
  }

  theta.old <- theta.int

  for (i in 1:total) {
    theta.new <- theta.old + (colSums(x) - n*theta.old)/n
    diff <- sqrt(sum((theta.new - theta.old)^2))
    theta.val[, i] <- theta.new
    diff.val[i] <- diff
    if (diff <= epsilon) {
      break
    } else {
      theta.old <- theta.new
    }
  }
  return(list(theta.est = theta.new, iter = i, theta.value = theta.val[, 1:i],
         diff.value = diff.val[1:i]))
}
output <- newton_mvnorm(x, theta.int = rep(-10, p))
```

3 Line search

Replacing the original equation with the following one:

$$\theta^{(t+1)} = \theta^{(t)} + s[A^{(t)}]^{-1} dL(\theta^{(t)}) = \theta^{(t)} + s\Delta\theta^{(t)}$$

where $A^{(t)}$ is a pd approximation of $-d^2L(\theta^{(t)})$ and s is a step length

Why? By first-order Taylor expansion

$$\begin{aligned} & L(\theta^{(t)} + s\Delta\theta^{(t)}) - L(\theta^{(t)}) \\ &= dL(\theta^{(t)})s\Delta\theta^{(t)} + o(s) \\ &= sdL(\theta^{(t)})[A^{(t)}]^{-1}dL(\theta^{(t)}) + o(s) \end{aligned}$$

One can make s as small as possible to make sure it is strictly positive.

- How to choose s? Step-halving: $s = 1, 1/2, ...$
- How to approximate $-d^2L(\theta)$? More of an art than science. Often requires problem specific analysis
- Taking $A = I$, the method is known as the gradient ascent
- Fisher's scoring method: replace $-d^2L(\theta)$ with the expected Fisher information matrix

  $$I(\theta) = E(-d^2L(\theta)) = E(dL(\theta)dL(\theta))$$

  is always p.s.d.